

The `hyperref-generic` module

A generic driver for `hyperref`

The L^AT_EX Project*

Version 0.95a, released 2021-02-22

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

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

The driver *requires* the new PDF management code, so documents wanting to use it should start like this

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

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

1 Avoiding transition problems

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

- Package options are processed at the end of the driver, Class options are ignored. But not every option already works as package options, in some cases `hyperref` interferes. So it is recommended for most options —with the exception of a few mentioned below in section 8—to set them in `\hypersetup`, not as package option.

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

- If the package `color` is used it is currently recommended to define colors after `hyperref`. This driver uses `l3color` for the colors, some special color models of `xcolor` won't work.
- Load a color package or `graphicx` to get the right page sizes.
- Report problems! Only known problem can be resolved.

2 Bookmarks / outlines

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

3 “Metadata”

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

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

The current handling of the metadata is problematic:

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

To resolve some of this problem the driver will

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

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

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

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

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

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

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

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

4 Dates

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

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

5 PDF page size (mediabox)

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

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

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

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

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

Some main changes are

- The default colors have been changed.
- `citecolor` is no longer set with `allcolors` and it is not part of the color scheme, it is only supported for compability.
- a number of color schemes have been predefined.

6.1 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.

6.2 New Keys

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

- Beside `pdfborder` there are also `linkborder`, `urlborder` etc
- Beside `pdfhighlight` there are also `linkhighlight`, `urlhighlight` etc
- Beside `pdfborderstyle` there are also `linkborderstyle`, `urlborderstyle` etc
- Beside `colorlinks` there are also `colorlink`, `colorurl` etc
- Beside `ocgcolorlinks` there are also `ocgcolorlink`, `ocgcolorurl`, etc TODO

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

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

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

6.3 Public interfaces

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

```
\l_hyp_annot_colorlink_bool,
\l_hyp_annot_colorurl_bool,
\l_hyp_annot_colorfile_bool,
\l_hyp_annot_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,
```

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

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

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

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

6.4 Changed behaviour

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

Color keys accept the following input syntax:

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

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

frenchlinks The option `frenchlinks` does nothing at all.

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

link margin The driver sets a default link margin—this is identical to `pdftex` and `luatex` driver, but a change for the `xetex` and `dvips` driver. The (undocumented) command `\setpdflinkmargin` does nothing. Use either the key `pdflinkmargin` or `\pdfannot_link_margin:n` to change the margin. See also the description in section 13 and in the `hyperref` manual.

7 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.

8 Package options from `hyperref`

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

Options that still must be set as package options are

- `backref`
- `CJKbookmarks` this key should not be used anymore. At some time it will be removed.
- `destlabels` (destination names are taken from `\label` if possible)

- `encap`
- `hyperfigures` (according to the `hyperref` manual it makes figures hyper links, but actually is a no-op for most drivers, and it does nothing with this driver either.)
- `hyperfootnotes`
- `hyperindex`
- `implicit` (redefine L^AT_EX internals)
- `nesting` unneeded key, see comment below in 13. 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`

9 Disabling links

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

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

10 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.

11 Dropped options

A number of options are ignored by this driver

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

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

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

unicode This is always true.

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

```
\DeclareDocumentMetadata{standard=A-2b}
```

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

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

- Resetbuttons, which use the action `/S/ResetForm` are suppressed.
- In widget annotations, the `/AA` dictionary is suppressed.

12 Destinations

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

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

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

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

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

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

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

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

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

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

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

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

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

13 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 **cmYk** (four numbers). While the PDF reference allows four numbers, PDF readers don’t necessarily handle this correctly, so the value can be wrong.

destlabel (*setup key*) This is a boolean key. Currently it must be set as package option. If set to true, the

name of a destination is taken from a following `\label`, if there is one before the next destination command. This requires two compilations to get the correct coordinates in the destination. In the first compilation the alias name is recorded in the aux-file:

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

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

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

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

will create

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

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

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

pdfborderstyle (*setup key*) The value of this key is the content of the BS dictionary. As an example **linkborderstyle** (*setup key*) `/Type/Border /W 1 /S/U /D[3 2]`

	Key	Values	comment / example
urlborderstyle (<i>setup key</i>)	<code>/Type</code>	<code>/Border</code>	optional
fileborderstyle (<i>setup key</i>)	<code>/W</code>	<code><number></code>	Width of border line
runborderstyle (<i>setup key</i>)	<code>/S</code>	<code>/S</code>	solid (default)
menuborderstyle (<i>setup key</i>)		<code>/D</code>	dash pattern can be set with <code>/D</code>
		<code>/B</code>	beveled
		<code>/I</code>	inset
		<code>/U</code>	underline
	<code>/D</code>	<code><array of numbers></code>	dash pattern (lines/gaps) (default [3])

pdfcreationdate (*setup key*) Setting these keys is normally not needed. If they are used the values are stored directly in the Info dictionary for `/Creationdate` and `/ModDate`. The values are converted to strings but not processed further, so they should have the correct PDF format without the enclosing parentheses, e.g. `D:20200202111111+01'00'`.

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

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

File I

hyperref-generic driver implementation

```
1 <*package>
2 <@@=hyp>
3 \ProvidesFile{hgeneric-testphase.def}
4 [2021/02/22 v0.95a %
5 generic Hyperref driver for the LaTeX PDF management testphase bundle]
6
7 \RequirePackage{etoolbox} %why?
8 \ExplSyntaxOn
9 \cs_gset_eq:NN \@expl@color@set@@nnn \color_set:nnn
10 \file_input:n {hyperref-colorschemes.def}
11 \ExplSyntaxOff
```

1 messages

Redirect the message name:

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

At first a message for the testing of the resource management

```
14 \msg_new:nnnn
15 { hyp }
16 { missing-resource-management }
17 { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
18 {
19   Activate~it~with \\
20   \tl_to_str:n{\RequirePackage{pdfmanagement-testphase}}\\
21   \tl_to_str:n{\DeclareDocumentMetadata{<options>}}\\
22   before~\tl_to_str:n{\documentclass}
23 }
```

The pdfversion should be set in \DeclareDocumentMetadata

```
24 \msg_new:nnnn
25 { hyp }
26 { pdfversion-disabled }
27 {
28   This~hyperref~driver~ignores~the~pdfversion~key!\\
29   Set~the~pdfversion~in~\token_to_str:N \DeclareDocumentMetadata
30 }
31 {
32   For example:\\
33   \tl_to_str:n
34   {
35     \DeclareDocumentMetadata { pdfversion=1.7 }
36   }
37 }
```

A generic message for ignored keys.

```
38 \msg_new:nnn
39 { hyp }
40 { key-dropped }
41 {
42   This~hyperref~driver~ignores~the~key~#1!\\
43   Please~check~the~documentation.
44 }
```

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

```
45 \msg_new:nnn
46 { hyp }
47 { pdfa-no-push-button }
48 { PDF/A:~Push~button~with~JavaScript~is~prohibited }
49
50 \msg_new:nnn
51 { hyp }
52 { pdfa-no-reset-button }
53 { PDF/A:~Reset~action~is~prohibited }
```

pdf/A message for not allowed Named actions

```
54 \msg_new:nnn
55 { hyp }
56 { pdfa-no-named-action }
57 { PDF/A:~Named~action~#1~is~prohibited }
```

A message if the destination name is empty.

```
58 \msg_new:nnn
59 { hyp }
60 { empty-destination-name }
61 {
62   Empty~destination~name,\\
63   using~'#1'
64 }
```

A message if the destination check fails

```
65 \msg_new:nnn
66 { hyp }
67 { invalid-destination-value }
68 {
69   Invalid~value~'#1'~of~'#2'  \\
70   is~replaced~by~'Fit'~\msg_line_context:.
71 }
```

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

```
72 \msg_new:nnn
73 { hyp }
74 { ignore-deprecated-or-unknown-option-in-pdf-version }
75 {
76   Option~'#1'~is~unknown~or~deprecated~in\\
77   pdf~version~#2.~Ignored.
78 }
79 \msg_new:nnn
80 { hyp }
81 { ignore-deprecated-or-unknown-value-in-pdf-version }
82 {
```

```

83     Value~'#1'~is~unknown~or~deprecated~in\\
84     pdf~version~#2.~Ignored.
85 }
86 \msg_new:nnn
87 { hyp }
88 { replace-deprecated-or-unknown-value-in-pdf-version }
89 {
90     Value~'#1'~is~unknown~or~deprecated~in\\
91     pdf~version~#2. Value~'#3'~is used instead.
92 }

```

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:

```

93 \msg_new:nnn
94 { hyp }
95 { unknown-key }
96 {
97     unknown-key~#2~of~module~'#1'~set~to~'#3'.
98 }
99 \msg_new:nnn
100 { hyp }
101 { unknown-key-to-Hyp }
102 {
103     ignored-in-family-Hyp-unknown-key~#1.
104 }

```

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

```

105 \cs_new:Npn \__hyp_clist_display:n #1 {*~#1\\}
106 \msg_new:nnn
107 { hyp }
108 { unknown-choice }
109 {
110     Value~'#3'~is~invalid~for~key~'#1'.\\
111     The~key~accepts~only~the~choices\\
112     \clist_map_function:nN { #2 }\__hyp_clist_display:n
113 }
114
115 \msg_new:nnn
116 { hyp }
117 { unknown-choice+empty }
118 {
119     Value~'#3'~is~invalid~for~key~'#1'.\\
120     The~key~accepts~only~the~choices\\
121     \clist_map_function:nN { #2 }\__hyp_clist_display:n
122     An~empty~value~removes~the~setting.
123 }
124
125 \msg_new:nnn
126 { hyp }
127 { no-bool }
128 {
129     Value~'#2'~is~invalid~for~key~'#1'.\\
130     The~key~accepts~only~the~choices\\
131     *~true\\

```

```

132     *~false \\
133     *~and~an~empty~value~which~removes~the~setting.\\
134     No~value~is~equivalent~to~using~‘true‘.
135 }

```

2 Variants

```

136 \cs_generate_variant:Nn\pdf_destination:nn {nf}

```

3 Overwriting/providing commands from hyperref

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

```

137 \chardef\Hy@VersionChecked=1 %don't check the version!
138 %\cs_set_protected:Npn \PDF@SetupDoc{}
139 %\PDF@FinishDoc{}% dummy needed for hyperref ...

```

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

```

140 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n

```

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

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

```

141 \providecommand\@pdfborder{}
142 \providecommand\@pdfborderstyle{}
143 \newcommand\OBJ@OCG@view {} % needed in hyperref
144 \def\Hy@numberline#1{#1\c_space_tl} %needed by bookmark

```

We force unicode as option. As bookmarks are still using \pdfstringdef we also need to load puenc.def.

```

145 \HyPsd@LoadUnicode
146 \Hy@unicodetrue
147 \let\HyPsd@pdfencoding\HyPsd@pdfencoding@unicode
148 \Hy@DisableOption{unicode}

```

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

```

149 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
150 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:
151 \legacy_if:nT { Hy@setpdfversion }
152 {
153     \msg_warning:nn { hyp }{ pdfversion-disabled }
154 }
155 \Hy@DisableOption{pdfversion}

```

\Acrobatmenu should use the new internal link command

```

156 \RenewDocumentCommand \Acrobatmenu { m m }
157 {
158     \hyper@linknamed {#1} {#2}
159 }

```

`\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!

```

160 % TODO should go at some time ...
161 % \kv@set@family@handler{Hyp}
162 % { \msg_warning:nnx {hyp}{unknown-key-to-Hyp}{#1} }
163 \cs_set_protected:Npn \hypersetup #1
164 {
165     %\kvsetkeys{Hyp} {#1}
166     \keys_set:nn { hyp / setup }{ #1 }
167 }
168 % TODO for now unknown keys should only give warnings.
169 \keys_define:nn { hyp / setup }
170 {
171     unknown .code:n =
172     {
173         \msg_warning:nnxxx { hyp } { unknown-key }
174         { hyp~/~setup }{ \l_keys_key_str } { #1 }
175     }
176 }

```

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.

```

177 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
178 {
179     %\tl_set:cn {@#1}{#2}
180     \AddToDocumentProperties[hyperref]{#1}{#2}
181 }
182 \cs_generate_variant:Nn \__hyp_store_metadata:nn {xn,nx,xx}

```

4.2 citecolor

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

```

183 \hook_new:n{hyp/link/cite}
184 \color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
185 \color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
186 \keys_define:nn { hyp / setup }
187 {
188     ,citecolor .code:n = {\__hyp_color_set:nn {hyp/color/cite}{#1}}
189     ,citebordercolor
190     .code:n = {\__hyp_color_set:nn {hyp/color/citeborder}{#1}}
191 }
192 \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
193 {
194     \keys_set:nn { hyp / setup }
195     {

```



```

196         ,linkbordercolor= hyp/color/citeborder
197         ,linkcolor        = hyp/color/cite
198     }
199 }

```

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

```

200 \bool_lazy_and:nnF
201 { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
202 { \msg_error:nn { hyp}{ missing-resource-management } }

```

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

```

203 \legacy_if:nT { Hy@bookmarks }
204 {
205     \AddToHook{begindocument/before}[hyperref/bookmark]{\RequirePackage{bookmark}}
206 }
207 \legacy_if:nT { Hy@draft }
208 {
209     \PassOptionsToPackage{draft}{bookmark}
210 }

```

6 Reference and label commands

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

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

```

    \__hyp_ref_label:nn
__hyp_ref_if_exist:nn 211 %
    \__hyp_ref_check:nn 212 \cs_new_protected:Npn \__hyp_ref_label:nn #1 #2 %label/attributes
    \__hyp_ref_value:nn 213 {
214     \@bsphack
215     \ref_label:nn{#1}{#2}
216     \@esphack
217 }
218 \cs_generate_variant:Nn \__hyp_ref_label:nn {en}

```

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

```

219 \prg_new_eq_conditional:Nnn \__hyp_ref_if_exist:nn \ref_if_exist:nn { p , T , F , TF }
220 \prg_generate_conditional_variant:Nnn \__hyp_ref_if_exist:nn {en} { p , T , F , TF }

```

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

```

221 \cs_new_protected:Npn \__hyp_ref_check:nn #1 #2 %label/attribute
222 {
223     \__hyp_ref_if_exist:nnF {#1}{#2}
224 }

```

```

225 \protect\G@refundefinedtrue
226 \@latex@warning
227 {
228   Reference~'\tl_to_str:n {#1}'~with~attribute~'\tl_to_str:n {#2}'~
229   on~page~\thepage~\space undefined
230 }
231 }
232 }
233 \cs_generate_variant:Nn \__hyp_ref_check:nn {en}

```

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

```

234 \cs_new:Npn \__hyp_ref_value:nn #1 #2
235 {
236   \ref_value:nn{#1}{#2}
237 }
238 \cs_generate_variant:Nn \__hyp_ref_value:nn {en}

```

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

7 Variables

7.1 Private temporary variables

At first a few generic tmp variables

```

\l__hyp_tmpa_tl
\l__hyp_tmpa_seq
\l__hyp_tmpa_int
\l__hyp_tmpa_box
\l__hyp_tmpa_str
239 \box_new:N \l__hyp_tmpa_box
240 \tl_new:N \l__hyp_tmpa_tl
241 \seq_new:N \l__hyp_tmpa_seq
242 \int_new:N \l__hyp_tmpa_int
243 \str_new:N \l__hyp_tmpa_str

```

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

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

```

\l__hyp_dest_name_tmpa_tl
\l__hyp_uri_tmpa_tl
\l__hyp_filename_tmpa_tl
\l__hyp_text_tmpa_str\l__hyp_text_tmpa_str
244 \tl_new:N \l__hyp_dest_name_tmpa_tl
245 \tl_new:N \l__hyp_uri_tmpa_tl
246 \tl_new:N \l__hyp_filename_tmpa_tl
247 \tl_new:N \l__hyp_para_tmpa_tl
248 \str_new:N \l__hyp_text_tmpa_str
249 \str_new:N \g__hyp_text_tmpa_str

```

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

7.2 Constants

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

```

250 \tl_const:Nn \c__hyp_dest_undefined_tl {UNDEFINED}

```

(End definition for `\c__hyp_dest_undefined_tl`.)

\c__hyp_annot_types_seq
\c__hyp_map_annot_hyp_prop
\c__hyp_map_hyp_annot_prop

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

```

251 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
252   {url,link,file,menu,run}
253 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
254   {
255     URI    = url,
256     GoTo   = link,
257     GoToR  = file,
258     Named  = menu,
259     Launch = run
260   }
261 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
262   {
263     url    = URI,
264     link   = GoTo,
265     file   = GoToR,
266     menu   = Named,
267     run    = Launch
268   }
269
```

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

7.3 Variables

\g__hyp_dest_pdfstartpage_tl
\g__hyp_dest_pdfstartview_tl
\l__hyp_dest_pdfremotestartview_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!

```

270 \tl_new:N \g__hyp_dest_pdfstartpage_tl
271 \tl_new:N \g__hyp_dest_pdfstartview_tl
272 \tl_new:N \l__hyp_dest_pdfremotestartview_tl

```

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

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

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

```

273 \tl_new:N \l__hyp_text_enc_uri_print_tl
274 \tl_new:N \l__hyp_text_enc_info_print_tl
275 \tl_new:N \l__hyp_text_enc_dest_tl
276 \tl_new:N \l__hyp_text_enc_dest_print_tl
277 \tl_new:N \l__hyp_text_enc_file_print_tl
278 \tl_new:N \l__hyp_text_enc_para_print_tl
279
280 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
281 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
282 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
283 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
284 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
285 \tl_set:Nn \l__hyp_text_enc_para_print_tl {utf8/string}

```

(End definition for \l__hyp_text_enc_uri_print_tl and others.)

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

286 \tl_new:N \l__hyp_dest_pdfview_tl

(End definition for \l__hyp_dest_pdfview_tl.)

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

hyp/annot/url (color name)

hyp/annot/file (color name)

\g__hyp_bordercolormodel_str

hyp/annot/run (color name)

hyp/annot/menu (color name)

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

287 \str_new:N \g__hyp_bordercolormodel_str

(End definition for \g__hyp_bordercolormodel_str.)

7.4 Booleans

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

\l_hyp_annot_colorurl_bool

\l_hyp_annot_colorfile_bool

\l_hyp_annot_colorrurl_bool

\l_hyp_annot_colormenu_bool

288 \seq_map_inline:Nn \c__hyp_annot_types_seq

289 {

290 \bool_new:c {l_hyp_annot_color#1_bool}

291 }

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

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

\l_hyp_annot_ocgcolorurl_bool

\l_hyp_annot_ocgcolorfile_bool

\l_hyp_annot_ocgcolorrurl_bool

\l_hyp_annot_ocgcolormenu_bool

292 \seq_map_inline:Nn \c__hyp_annot_types_seq

293 {

294 \bool_new:c {l_hyp_annot_ocgcolor#1_bool}

295 }

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

\not_Named_bool, \l_hyp_annot_Launch_bool

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

296 \seq_map_inline:Nn \c_pdfannot_link_types_seq

297 {

298 \bool_new:c {l__hyp_annot_#1_bool}

299 \bool_set_true:c {l__hyp_annot_#1_bool}

300 }

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

7.5 Boxes

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

301 \box_new:N \l__hyp_dest_box

(End definition for \l__hyp_dest_box.)

7.6 Regex

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

```

302 \regex_const:Nn \c__hyp_dest_startview_regex
303 {
304   \A\ *
305   (?:
306     (?:XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
307     |
308     (?:Fit\b|FitB\b)
309     |
310     (?:\b(?:FitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1})
311     |
312     (?:FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
313   )
314 }
```

(End definition for `\c__hyp_dest_startview_regex`.)

7.7 PDF dictionaries

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

```

315 \pdfdict_new:n {l__hyp_page/Trans}
316 \pdfdict_put:nnn {l__hyp_page/Trans}{Type}{/Trans}
```

(End definition for `l__hyp_page/Trans`.)

8 PDF string conversion

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

`hyp/text/pdfstring`

```

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

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

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

`__hyp_text_purify:nN`

```

318 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
319 {
320   \str_set:Nx #2 {\text_purify:n { #1 } }
321 }
```

(End definition for `__hyp_text_purify:nN`.)

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

`__hyp_text_cleanup:N`

```
322 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
323 {
324
325 }
```

(End definition for __hyp_text_cleanup:N.)

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

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

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

`__hyp_text_string_from_unicode:nN`

```
326 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
327 {
328   \pdf_string_from_unicode:nVN { #1 } #2 #2
329 }
```

(End definition for __hyp_text_string_from_unicode:nN.)

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

`__hyp_text_pdfstring:nnN`

```
330 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
331 {
332   \group_begin:
333   \Hy@pdfstringtrue
334   \hook_use:n {hyp/text/pdfstring}
335   \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
336   \__hyp_text_cleanup:N \l__hyp_text_tmpa_str
337   \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
338   \str_gset_eq:NN \g__hyp_text_tmpa_str\l__hyp_text_tmpa_str
339   \group_end:
340   \str_set_eq:NN #3 \g__hyp_text_tmpa_str
341 }
342 \cs_generate_variant:Nn \__hyp_text_pdfstring:nnN {xnN,onN,xoN,ooN,noN}
```

(End definition for __hyp_text_pdfstring:nnN.)

!!! temporary until all instances are gone

```
343 \cs_new_protected:Npn \Hy@pstringdef #1 #2
344 { \__hyp_text_pdfstring:xnN {#2} {utf8/string-raw}#1 }
```

This is a special version for info keys:

```

\__hyp_text_pdfstring_info:nN
345 \cs_new_protected:Npn \__hyp_text_pdfstring_info:nN #1 #2
346 {
347   \__hyp_text_pdfstring:nn { #1 } { \l__hyp_text_enc_info_print_tl } #2
348 }

```

(End definition for __hyp_text_pdfstring_info:nN.)

9 Pagelabels

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

The current implementation in hyperref/hyperref drivers:

xetex: hxdetex.def, line 80-110
`\HyPL@StorePageLabel` writes to the aux-file at begin document (after reading the aux) `\HyPL@SetPageLabels` is called (defined in hyperref.sty after the driver loading) which calls `\Hy@PutCatalog{/PageLabels<</Nums[\HyPL@Labels]>>}`

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

dvipdfm: identical to xetex

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

luatex identical to pdftex

The code in hyperref inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-23 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
349 \cs_new_protected:Npn \__hyp_PageLabels_gpush:
350 {
351   \pdfmanagement_add:nnx {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
352 }
353
354 \def\Hy@PutCatalog #1 {}
355
356
357 \legacy_if:nT { Hy@pdfpagelabels }
358 {

```

```

359 \cs_set_protected:Npn \HyPL@StorePageLabel #1
360 {
361   \tl_gput_right:Nx \HyPL@Labels { \the\Hy@abspage<<#1>> }
362   \__hyp_PageLabels_gpush:
363 }
364 }

```

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

10 Core Hyperref Commands

Every hyperref has to define eight core command:

```

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

```

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

10.1 Anchors / destinations

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

```

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

```

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

```

\__hyp_destination:nn

```

```

365 \Hy@WrapperDef \__hyp_destination:nn #1 #2
366 {
367   \mode_if_horizontal:T { \@savsf\spacefactor }
368   \Hy@SaveLastskip      %defined in hyperref
369   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
370   \__hyp_text_pdfstring:xoN
371   { \HyperDestNameFilter{#1} }
372   { \l__hyp_text_enc_dest_tl }
373   \l__hyp_tmpa_tl
374   \str_if_eq:nnTF {#2} {fitrbox}
375   {
376     \exp_args:NV
377     \pdf_destination:nnnn \l__hyp_tmpa_tl
378     { \box_wd:N \l__hyp_dest_box }

```



```

379         { \box_ht:N \l__hyp_dest_box }
380         { \box_dp:N \l__hyp_dest_box }
381     }
382     {
383         \exp_args:NV
384         \pdf_destination:nf
385         { \l__hyp_tmpa_tl }
386         { #2 }
387     }
388     \Hy@RestoreLastskip    %defined in hyperref
389     \mode_if_horizontal:T { \spacefactor\@savsf }
390 }

```

(End definition for `__hyp_destination:nn`.)

This are the three destinations commands. They are modelled along the xetex version. It is not quite clear if really all three are needed for the backends supported by this driver, but changing the hyperref code would be difficult.

```

\hyper@anchor
\hyper@anchorstart
\hyper@anchorend
391 \cs_new_protected:Npn \hyper@anchor #1
392 {
393     \exp_args:NnV
394     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
395 }
396
397 \cs_new_protected:Npn \hyper@anchorstart #1
398 {
399     \Hy@activeanchortrue
400     \exp_args:NnV
401     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
402 }
403
404 \cs_new_protected:Npn \hyper@anchorend
405 {
406     \Hy@activeanchorfalse
407 }

```

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

10.2 GoTo Links

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

At first the internal link commands:

```

408 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
409 {
410     \mode_leave_vertical:
411     \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
412     \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
413     {
414         \msg_warning:nnx

```

```

415         { hyp }
416         { empty-destination-name }
417         { \c__hyp_dest_undefined_tl }
418         \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
419     }
420     {
421         \__hyp_text_pdfstring:xoN
422         { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
423         { \l__hyp_text_enc_dest_tl }
424         \l__hyp_dest_name_tmpa_tl
425     }
426     \exp_args:No
427     \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
428 }
429
430 \cs_new_protected:Npn \__hyp_link_goto_end:
431 {
432     \pdfannot_link_goto_end:
433 }

```

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

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

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

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

```

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

```

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

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

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

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

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

```

434
435 \cs_new_protected:Npn \hyper@link #1 #2 #3 %#1 context, #2=destination name, #3 content
436 {
437     \bool_if:NTF \l__hyp_annot_GoTo_bool
438     {
439         \Hy@VerboseLinkStart{#1}{#2}
440         \group_begin:
441         \hook_use:n {hyp/link/#1}
442         \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
443         \__hyp_link_goto_end:

```

```

444         \group_end:
445         \Hy@VerboseLinkStop
446     }{#3}
447 }
448
449 \cs_new_protected:Npn \hyper@linkstart #1 #2 %#1 context, #2=destination name
450 {
451     \bool_if:NT \l__hyp_annot_GoTo_bool
452     {
453         \Hy@VerboseLinkStart{#1}{#2}% only for debug
454         \group_begin:
455         \hook_use:n {hyp/link/#1}
456         \__hyp_link_goto_begin:nw {#2}
457     }
458 }
459
460 \cs_new_protected:Npn \hyper@linkend
461 {
462     \bool_if:NT \l__hyp_annot_GoTo_bool
463     {
464         \__hyp_link_goto_end:
465         \group_end:
466         \Hy@VerboseLinkStop
467     }
468 }

```

10.3 URI links

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

```

469 \pdfdict_new:n {l_hyp/annot/A/URI}
470 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
471 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
472
473 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
474 {
475     \bool_if:NTF \l__hyp_annot_URI_bool
476     {
477         \group_begin:
478         \__hyp_text_pdfstring:xoN
479         { #2}
480         { \l__hyp_text_enc_uri_print_tl }
481         \l__hyp_uri_tmpa_tl
482         \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
483         \ifHy@href@ismap
484             \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
485         \fi
486         \tl_if_empty:NF \Hy@href@nextactionraw
487         {
488             \str_remove_once:Nn \Hy@href@nextactionraw {/Next}
489             \pdfdict_put:nno{l_hyp/annot/A/Next}{\Hy@href@nextactionraw}
490         }
491         \cs_set_eq:NN \# \c_hash_str
492         \cs_set_eq:NN \% \c_percent_str
493         \Hy@safe@activestrue

```

```

494     \mode_leave_vertical:
495     \pdfannot_link:nxn { URI }
496     {
497         /A
498         <<
499             \pdfdict_use:n {l_hyp/annot/A/URI}
500         >>
501     }
502     {
503         #1
504         \Hy@xspace@end
505         \Hy@VerboseLinkStop %where is the start??
506     }
507     \group_end:
508 }
509 {#1}
510 }
511

```

10.4 GoToR Links files

```

512 \pdfdict_new:n {l_hyp/annot/A/GoToR}
513 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
514 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
515
516 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
517 {
518     \bool_if:NTF \l__hyp_annot_GoToR_bool
519     {
520         \group_begin:
521         \tl_set:Nn \l__hyp_filename_tmpa_tl { #2 }
522         \Hy@CleanupFile \l__hyp_filename_tmpa_tl
523         \__hyp_text_pdfstring:ooN
524         { \l__hyp_filename_tmpa_tl }
525         { \l__hyp_text_enc_file_print_tl }
526         \l__hyp_filename_tmpa_tl
527         \pdfdict_put:nno {l_hyp/annot/A/GoToR}{F}{\l__hyp_filename_tmpa_tl}
528         \__hyp_text_pdfstring:nnN
529         { #3 }
530         { \l__hyp_text_enc_dest_print_tl }
531         \l__hyp_dest_name_tmpa_tl
532         \Hy@MakeRemoteAction
533         \tl_if_blank:eTF {#3}
534         {
535             \pdfdict_put:nnx {l_hyp/annot/A/GoToR}{D}
536             {[ \Hy@href@page/\l__hyp_dest_pdfremotestartview_tl]}
537         }
538         {
539             \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
540         }
541         \tl_if_empty:NF \Hy@href@nextactionraw
542         {
543             \str_remove_once:Nn \Hy@href@nextactionraw {/Next}
544             \pdfdict_put:nno{l_hyp/annot/A/Next}{Next}{\Hy@href@nextactionraw}

```

```

545     }
546     \mode_leave_vertical:
547     \pdfannot_link:nxn %expansion??
548     { GoToR }
549     {
550         /A<<
551         \pdfdict_use:n {l_hyp/annot/A/GoToR}
552         >>
553     }
554     {
555         #1\Hy@xspace@end
556         \Hy@VerboseLinkStop %where is the start??
557     }
558     \group_end:
559 }
560 {#1}
561 }

```

10.5 Launch links

We define \hyper@linklaunch for naming consistency

```

562 \pdfdict_new:n {l_hyp/annot/A/Launch}
563 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
564 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
565
566 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
567 {
568     \bool_if:NTF \l__hyp_annot_Launch_bool
569     {
570         \group_begin:
571         \__hyp_text_pdfstring:nnN
572         { #1 }
573         { \l__hyp_text_enc_file_print_tl }
574         \l__hyp_filename_tmpa_tl
575         \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
576         \__hyp_text_pdfstring:noN
577         { #3 }
578         { \l__hyp_text_enc_para_print_tl }
579         \l__hyp_para_tmpa_tl
580         \bool_if:nTF
581         {
582             \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
583             ||
584             \pdf_version_compare_p:Nn > {1.9}
585         }
586         {
587             \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
588         }
589         {
590             \pdfdict_put:nnx {l_hyp/annot/A/Launch}{Win}
591             {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
592         }
593         \tl_if_empty:NF \Hy@href@nextactionraw
594         {

```

```

595         \str_remove_once:Nn \Hy@href@nextactionraw {/Next}
596         \pdfdict_put:nno{l_hyp/annot/A/Next}{Next}{\Hy@href@nextactionraw}
597     }
598     \mode_leave_vertical:
599     \pdfannot_link:nxn
600     { Launch }
601     {
602         /A
603         <<
604         \pdfdict_use:n {l_hyp/annot/A/Launch}
605         >>
606     }
607     {
608         #2\Hy@xspace@end
609         \Hy@VerboseLinkStop %where is the start??
610     }
611     \group_end:
612 }
613 {#2}
614 }

```

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

```

615 \use:x
616 { % filename, anchor text, linkname
617     \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str ##1 \exp_not:N \ \ ##2 #
618 }
619 {
620     \hyper@linklaunch {#1}{#2}{#3}
621 }

```

10.6 Named links (menu)

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

```

622 \pdfdict_new:n {l_hyp/annot/A/Named}
623 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
624 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
625
626 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
627 {
628     \bool_if:NTF \l__hyp_annot_Named_bool
629     {
630         \group_begin:
631         \pdfmeta_standard_verify:nnTF {named_actions}{#1}
632         {
633             \mode_leave_vertical:
634             \pdfdict_put:nnx {l_hyp/annot/A/Named}{N}
635             {\pdf_name_from_unicode_e:n{#1}}
636             \tl_if_empty:NF \Hy@href@nextactionraw
637             {
638                 \str_remove_once:Nn \Hy@href@nextactionraw {/Next}
639                 \pdfdict_put:nno{l_hyp/annot/A/Next}{Next}{\Hy@href@nextactionraw}
640             }
641             \pdfannot_link:nxn { Named }

```

```

642         {
643             /A
644             <<
645             \pdfdict_use:n { l_hyp/annot/A/Named }
646             >>
647         }
648         {
649             #2
650             \Hy@xspace@end
651             \Hy@VerboseLinkStop
652         }
653     }
654     {
655         \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
656         #2
657     }
658     \group_end:
659 }
660 {#2}
661 }
662

```

11 Link decorations

11.1 Functions to export and select colors

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

<code>_hyp_color_export:nnN</code>	<code>_hyp_color_export:nnN {<color>} {<export format>} <tlvar></code>
<code>_hyp_colormodel_export:nnnN</code>	<code>_hyp_colormodel_export:nnnN {<color model>}{<value>} {<export format>}</code>
	<code><tlvar></code>

This exports a color as space separated numbers as needed in in /C. {<color>} should have either the format [model]{value} or be a color expression. For examples: [rgb]{1,0,.5} or red!50!blue. The export format is either space-sep-cmyk or space-sep-rgb.

`_hyp_color_export:nnN` Color keys need to parse color expressions. Two input types are supported: `color=[rgb]{1,0,.5}` and `color=red!50!blue`. The colors are exported to PDF syntax (space separated numbers).

```

663 \cs_new_protected:Npn \_hyp_color_export:nnN #1 #2 #3
664 {
665     \tl_if_head_eq_charcode:nTF {#1}[ %]
666     {
667         \_hyp_colormodel_export:wnnN #1 {#2} #3
668     }
669     {
670         \color_export:nnN {#1} {#2} #3
671     }
672 }
673
674 \cs_new_protected:Npn \_hyp_colormodel_export:wnnN [#1] #2 #3 #4
675 {

```

```

676     \color_export:nnnN {#1}{#2}{#3}#4
677   }
678
679   \cs_generate_variant:Nn \__hyp_color_export:nnN {xVN}

```

(End definition for __hyp_color_export:nnN and __hyp_colormodel_export:nnnN.)

```

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

```

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

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

```

680   \cs_new_protected:Npn \__hyp_color_select:n #1
681   {
682     \tl_if_head_eq_charcode:nNTF {#1}[ %]
683     {
684       \__hyp_colormodel_select_aux:wn #1
685     }
686     {
687       \color_select:n {#1}
688     }
689   }
690
691   \cs_new_protected:Npn \__hyp_colormodel_select_aux:wn [#1] #2
692   {
693     \color_select:nn {#1}{#2}
694   }
695
696   \cs_generate_variant:Nn \__hyp_color_select:n {e}

```

(End definition for __hyp_color_select:n and __hyp_colormodel_select_aux:wn.)

```

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

```

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

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

```

697   \cs_new_protected:Npn \__hyp_color_set:nn #1 #2
698   {
699     \tl_if_head_eq_charcode:nNTF {#2}[ %]
700     {
701       \__hyp_colormodel_set_aux:wn { #1 } #2
702     }
703     {
704       \color_set:nn {#1} {#2}
705     }
706   }
707
708   \cs_new_protected:Npn \__hyp_colormodel_set_aux:wn #1 [#2] #3
709   {

```



```

710     \color_set:nnn {#1}{#2}{#3}
711   }
712
713   \cs_generate_variant:Nn \__hyp_color_set:nn {ne}

```

(End definition for __hyp_color_set:nn and __hyp_colormodel_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.

```

714 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
715 {
716   \hook_gput_code:nnn
717     {pdfannot/link/#2/begin}
718     {hyp/color}
719     {
720       \bool_if:cT { l_hyp_annot_color#1_bool }
721       {
722         \group_begin:
723         \color_select:n { hyp/color/#1}
724       }
725     }
726   \hook_gput_code:nnn
727     {pdfannot/link/#2/end}
728     {hyp/color}
729     {
730       \bool_if:cT { l_hyp_annot_color#1_bool }
731       {
732         \group_end:
733       }
734     }
735 }

```

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

```

736 \keys_define:nn { hyp / setup }
737 {
738   ,colorlinks .meta:n =
739   {
740     ,pdfborder={0~0~0}
741     ,pdfborderstyle=
742     ,colorurl   =#1
743     ,colorlink  =#1
744     ,colorrune  =#1
745     ,colormenu  =#1
746     ,colorfile  =#1
747   }
748   ,colorlinks .default:n = {true}
749 }

```

```

colorurl (setup key)
colorlink (setup key) 750 \seq_map_inline:Nn \c__hyp_annot_types_seq
colorrune (setup key) 751 {
colormenu (setup key)
colorfile (setup key)
urlcolor (setup key)
linkcolor (setup key)
runcolor (setup key)
menucolor (setup key)
filecolor (setup key)
allcolor (setup key)

```

```

752 \keys_define:nn { hyp / setup }
753 {
754   ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
755   ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
756 }
757 }
758
759 \keys_define:nn { hyp / setup }
760 {
761   ,allcolors .meta:n =
762   {
763     ,urlcolor=#1
764     ,linkcolor=#1
765     ,runcolor=#1
766     ,filecolor=#1
767     ,menucolor=#1
768   }
769   ,allcolors .value_required:n = true
770 }

```

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*)

```

771 \keys_define:nn { hyp / setup }
772 {
773   ,bordercolormodel .choices:nn =
774   {rgb,cmyk}
775   { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
776   ,bordercolormodel .initial:n = {rgb}
777 }
778
779 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
780 {
781   \keys_define:nn { hyp / setup }
782   {
783     #1bordercolor .code:n =
784     {
785       \tl_if_empty:nTF { ##1 }
786       {
787         \pdfannot_dict_remove:nn
788         {link/#2}
789         { C }
790       }
791       {
792         \__hyp_color_export:xVN {##1}\g__hyp_bordercolormodel_str \l__hyp_tmpa_tl
793         \pdfannot_dict_put:nnx
794         {link/#2}
795         { C }
796         { [\l__hyp_tmpa_tl] }
797       }
798     }
799   }
800 }

```

```

797     }
798   }
799 }
800
801 \keys_define:nn { hyp / setup }
802 {
803   ,allbordercolors .meta:n =
804   {
805     ,linkbordercolor=#1
806     ,urlbordercolor =#1
807     ,filebordercolor=#1
808     ,menubordercolor=#1
809     ,runbordercolor =#1
810   }
811   ,allbordercolors .value_required:n = true
812 }
813

```

11.3.2 Borderwidth and -arc

```

814 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
815 {
816   \keys_define:nn { hyp / setup }
817   {
818     #1border .code:n =
819     {
820       \tl_if_empty:nTF { ##1 }
821       {
822         \pdfannot_dict_remove:nn
823         {link/#2}
824         { Border }
825       }
826       {
827         \pdfannot_dict_put:nnn
828         {link/#2}
829         { Border }
830         { [##1] }
831       }
832     }
833   }
834 }
835 \keys_define:nn { hyp / setup }
836 {
837   ,pdfborder .code:n =
838   {
839     \tl_if_empty:nTF { #1 }
840     {
841       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
842       {
843         \pdfannot_dict_remove:nn
844         {link/##2}
845         { Border }
846       }
847     }
848   }
849 }

```

```

848     {
849       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
850       {
851         \pdfannot_dict_put:nnn
852         {link/##2}
853         { Border }
854         { [#1] }
855       }
856     }
857   }
858   ,pdfborder .initial:n = {0~0~1},
859 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 860 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 861 {
runborderstyle (setup key) 862   \keys_define:nn { hyp / setup }
fileborderstyle (setup key) 863   {
menuborderstyle (setup key) 864     #1borderstyle .code:n =
865     {
866       \tl_if_empty:nTF { ##1 }
867       {
868         \pdfannot_dict_remove:nn
869         {link/#2}
870         { BS }
871       }
872       {
873         \pdfannot_dict_put:nnn
874         {link/#2}
875         { BS }
876         { <<##1>> }
877       }
878     }
879   }
880 }
881 \keys_define:nn { hyp / setup }
882 {
883   ,pdfborderstyle .code:n =
884   {
885     \tl_if_empty:nTF { #1 }
886     {
887       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
888       {
889         \pdfannot_dict_remove:nn
890         {link/##2}
891         { BS }
892       }
893     }
894     {
895       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
896       {

```

```

897         \pdfannot_dict_put:nnn
898         {link/##2}
899         { BS }
900         { <<#1>> }
901     }
902 }
903 }
904 ,pdfborderstyle .initial:n = {},
905 }

```

11.4 ocgcolorlinks

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

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

```

906 \cs_new_protected:Npn \__hyp_ocg_init:
907 {
908     \pdf_object_new:nn { l__hyp_ocg_view_dict_obj } { dict }
909     \pdf_object_new:nn { l__hyp_ocg_print_dict_obj } { dict }
910     \pdf_object_new:nn { l__hyp_ocg_config_dict_obj } { dict }
911     \pdf_object_new:nn { l__hyp_ocg_ref_array_obj } { array }
912     \pdf_object_write:nx { l__hyp_ocg_ref_array_obj }
913     {
914         \pdf_object_ref:n { l__hyp_ocg_view_dict_obj }
915         \c_space_tl
916         \pdf_object_ref:n { l__hyp_ocg_print_dict_obj }
917     }
918     \pdf_object_write:nn { l__hyp_ocg_view_dict_obj }
919     {
920         /Type/OCG
921         /Name(View)
922         /Usage
923         <<
924             /Print <</PrintState/OFF>>~
925             /View <</ViewState/ON >>~
926         >>
927     }
928     \pdf_object_write:nn { l__hyp_ocg_print_dict_obj }
929     {
930         /Type/OCG
931         /Name(Print)
932         /Usage
933         <<
934             /Print <</PrintState/ON>>~
935             /View <</ViewState/OFF>>~
936         >>
937     }
938     \pdfmanagement_add:nnx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {l__hyp_ocg_view_dict_obj} }
939     \pdfmanagement_add:nnx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {l__hyp_ocg_print_dict_obj} }
940     \pdf_object_write:nx { l__hyp_ocg_config_dict_obj }
941     {
942         /OFF[\pdf_object_ref:n { l__hyp_ocg_print_dict_obj }]

```

```

943     /AS[
944         <<
945         /Event/View
946         /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
947         /Category[/View]
948         >>
949         <<
950         /Event/Print
951         /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
952         /Category[/Print]
953         >>
954         <<
955         /Event/Export
956         /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
957         /Category[/Print]
958         >>
959     ]
960 }
961 \pdfmanagement_add:nxx { Catalog / OCPProperties }{ D }{ \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
962 \cs_gset:Npn \__hyp_ocg_init: {}
963 }

```

(End definition for __hyp_ocg_init:.)

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

```

964 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
965 {
966     \hook_gput_code:nnn
967     {pdfannot/link/#2/begin}
968     {hyp/ocg}
969     {
970         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
971         {
972             \__hyp_ocg_init:
973             \group_begin:
974             \hbox_set:Nw \l__hyp_tmpa_box
975         }
976     }
977     \hook_gput_code:nnn
978     {pdfannot/link/#2/end}
979     {hyp/ocg}
980     {
981         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
982         {
983             \hbox_set_end:
984             \mbox
985             {
986                 \pdf_bdcobject:nn {OC}{l__hyp_ocg_print_dict_obj}
987                 \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
988                 \pdf_emc:
989                 \pdf_bdcobject:nn {OC}{l__hyp_ocg_view_dict_obj}
990                 \group_begin:
991                 \color_select:n { hyp/color/#1 }
992                 \box_use_drop:N \l__hyp_tmpa_box

```

```

993         \group_end:
994         \pdf_emc:
995     }
996     \group_end:
997 }
998 }
999 }

```

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

```

1000 \pdf_version_compare:NnTF < {1.5}
1001 {
1002     \keys_define:nn { hyp / setup }
1003     {
1004         ,_ocgcolorlinks .code:n =
1005         {
1006             \msg_warning:nnxx
1007             { hyp }
1008             { ignore-deprecated-or-unknown-option-in-pdf-version }
1009             { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1010         }
1011     }
1012 }
1013 {
1014     \keys_define:nn { hyp / setup }
1015     {
1016         ,_ocgcolorlinks .meta:n =
1017         {
1018             ocgcolorlink=#1,
1019             ocgcolorurl=#1,
1020             ocgcolorfile=#1,
1021             ocgcolorrun=#1,
1022             ocgcolormenu=#1
1023         }
1024         ,_ocgcolorlinks .default:n = true
1025     }
1026 }
1027
1028 \keys_define:nn { hyp / setup }
1029 {
1030     ,ocgcolorlinks .choice:
1031     ,ocgcolorlinks / true .meta:n =
1032     {
1033         pdfborder      = {0~0~0},
1034         pdfborderstyle = {},
1035         colorlinks     = false,
1036         _ocgcolorlinks = true
1037     }
1038     ,ocgcolorlinks / false .meta:n =
1039     {
1040         _ocgcolorlinks = false
1041     }
1042     ,ocgcolorlinks .default:n = {true}
1043 }
1044

```

```

1045 \seq_map_inline:Nn \c__hyp_annot_types_seq
1046 {
1047   \pdf_version_compare:NnTF < {1.5}
1048   {
1049     \keys_define:nn { hyp / setup }
1050     {
1051       ,ocgcolor#1 .code:n=
1052       {
1053         \msg_warning:nnxx
1054         { hyp }
1055         { ignore-deprecated-or-unknown-option-in-pdf-version }
1056         { ocgcolor#1 }
1057         { \pdf_version_major:.\pdf_version_minor: }
1058       }
1059     }
1060   }
1061   {
1062     \keys_define:nn { hyp / setup }
1063     {
1064       ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1065     }
1066   }
1067 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1068 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1069 {
1070   \keys_define:nn { hyp / setup }
1071   {
1072     ,#1highlight .choices:nn =
1073     { /I, /N, /O, /P }
1074     {
1075       \pdfannot_dict_put:nnn
1076       { link/#2 }
1077       { H }
1078       { ##1 }
1079     }
1080     ,#1highlight / .code:n =
1081     {
1082       \pdfannot_dict_remove:nn
1083       { link/#2 }
1084       { H }
1085     }
1086   }
1087   ,#1highlight / unknown .code:n =
1088   {
1089     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1090     { pdfhighlight }
1091     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1092     { \exp_not:n {##1} }
1093   }
1094 }

```



```

1095     }
1096   }
1097
1098
1099   \keys_define:nn { hyp / setup }
1100   {
1101     ,pdfhighlight .choices:nn =
1102     { /I, /N, /O, /P}
1103     {
1104       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1105       {
1106         \pdfannot_dict_put:nnn
1107         {link/####2}
1108         { H }
1109         { #1 }
1110       }
1111     }
1112     ,pdfhighlight / .code:n =
1113     {
1114       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1115       {
1116         \pdfannot_dict_remove:nn
1117         {link/##2}
1118         { H }
1119       }
1120     }
1121     ,pdfhighlight .initial:n = {/I},
1122     ,pdfhighlight / unknown .code:n =
1123     {
1124       \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1125       { pdfhighlight }
1126       { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1127       { \exp_not:n {#1} }
1128     }
1129   }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1130 \keys_define:nn { hyp / setup }
hideurl (setup key) 1131 {
hidefile (setup key) 1132   hidelinks .meta:n =
hiderun (setup key) 1133   {
hidemenu (setup key) 1134     ,colorlinks      = false
1135     ,ocgcolorlinks = false
1136     ,pdfborder      = { 0~0~0 }
1137     ,pdfborderstyle=
1138   }
1139 }
1140
1141 \seq_map_inline:Nn \c__hyp_annot_types_seq
1142 {

```

```

1143 \keys_define:nn { hyp / setup }
1144 {
1145   hide#1 .meta:n =
1146   {
1147     ,color#1      = false
1148     ,ocgcolor#1   = false
1149     ,#1border     = { 0~0~0 }
1150     ,#1borderstyle =
1151   }
1152 }
1153 }

```

11.7 color schemes and settings

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

`colorscheme (setup key)`

```

1154 \keys_define:nn { hyp / setup }
1155 {
1156   colorscheme .code:n =
1157   {
1158     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1159     {
1160       \keys_set:nn { hyp /setup }
1161       {
1162         ##1 = ##2
1163       }
1164     }
1165   }
1166 }
1167 \keys_set:nn { hyp / setup } {colorscheme=phetype}

```

12 Keys

12.1 Ignored keys

The following are ignored (with or without warnings)

`unicode (setup key)`

`pdfencoding (setup key)`

`pdfversion (setup key)`

```

1168 \keys_define:nn { hyp / setup }
1169 {
1170   ,unicode      .code:n = {}
1171   ,pdfencoding  .code:n = {}
1172   ,pdfversion   .code:n =
1173   {
1174     \msg_warning:nn { hyp }{ pdfversion-disabled }
1175   }
1176 }
1177 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
  debug (setup key) 1178 \keys_define:nn { hyp / setup }
  draft (setup key) 1179 {
  final (setup key) 1180   ,verbose .choice:
1181   ,verbose / true .code:n = { \Hy@verbosetrue}
1182   ,verbose / false .code:n = { \Hy@verbosefalse}
1183   ,verbose .default:n = {true}
1184   ,debug .meta:n = {verbose=#1}
1185   ,debug .default:n = {true}
1186 }
1187 \keys_define:nn { hyp / setup }
1188 {
1189   ,draft .code:n =
1190   {
1191     \Hy@drafttrue
1192     \PassOptionsToPackage{draft}{bookmark}
1193   }
1194   ,final .code:n =
1195   {
1196     \Hy@finaltrue
1197     \PassOptionsToPackage{final}{bookmark}
1198   }
1199 }

extension (setup key)
hypertextnames (setup key) 1200 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1201 {
  pageanchor (setup key) 1202   ,extension .tl_set:N = \XR@ext
  linktoc (setup key) 1203   ,extension .initial:n= pdf
  linktocpage (setup key) 1204   ,hypertextnames .choice:
  plainpages (setup key) 1205   ,hypertextnames / true .code:n = { \Hy@hypertextnamestrue}
localanchorname (setup key) 1206   ,hypertextnames / false .code:n = { \Hy@hypertextnamesfalse}
linkfileprefix (setup key) 1207   ,hypertextnames .default:n = {true}
1208   ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1209   ,localanchorname .choice:
1210   ,localanchorname / true .code:n = { \Hy@localanchorname true }
1211   ,localanchorname / false .code:n = { \Hy@localanchorname false }
1212   ,localanchorname .default:n = {true}
1213   ,naturalnames .choice:
1214   ,naturalnames / true .code:n = { \Hy@naturalnames true}
1215   ,naturalnames / false .code:n = { \Hy@naturalnames false}
1216   ,naturalnames .default:n = {true}
1217   ,pageanchor .choice:
1218   ,pageanchor / true .code:n = { \Hy@pageanchortrue}
1219   ,pageanchor / false .code:n = { \Hy@pageanchorfalse}
1220   ,pageanchor .default:n = {true}
1221   ,plainpages .choice:
1222   ,plainpages / true .code:n = { \Hy@plainpagestrue}
1223   ,plainpages / false .code:n = { \Hy@plainpagesfalse}
1224   ,plainpages .default:n = {true}

```

```

1225 }
1226
1227 \keys_define:nn { hyp / setup }
1228 {
1229   ,linktoc .choices:nn = { none, section, all, page }
1230   {
1231     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1232   }
1233   ,linktoc / unknown .code:n =
1234   {
1235     \msg_warning:nnxxx { hyp } { unknown-choice }
1236     { linktoc }
1237     { none, section, all, page }
1238     { \exp_not:n {#1} }
1239   }
1240   ,linktocpage .choice:
1241   ,linktocpage / true .meta:n = {linktoc=page}
1242   ,linktocpage / false .meta:n = {linktoc=section}
1243   ,linktocpage .default:n = true
1244 }
1245

```

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

```

url (setup key) 1246 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1247 {
menu (setup key) 1248   \keys_define:nn { hyp / setup }
run (setup key) 1249   {
1250     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1251   }
1252 }

```

```

1253 \keys_define:nn { hyp / setup }
1254 {
1255   ,baseurl .code:n =
1256   {
1257     \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1258     \tl_if_empty:NTF \l__hyp_tmpa_tl
1259     {
1260       \pdfmanagement_remove:nn {Catalog} { URI }
1261     }
1262     {
1263       \pdfmanagement_add:nnx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1264     }
1265   }
1266   %only false does something ...
1267   ,bookmarks .choice:
1268   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1269   ,bookmarks / true .code:n = {}
1270   ,bookmarks .default:n = {true}
1271   ,bookmarksnumbered .choice:
1272   ,bookmarksnumbered / false .code:n = { \Hy@bookmarksnumberedfalse }
1273   ,bookmarksnumbered / true .code:n = { \Hy@bookmarksnumberedtrue }
1274   ,bookmarksnumbered .default:n = {true}
1275   ,bookmarksopen .choice:

```

```

1276 ,bookmarksopen / false .code:n = { \Hy@bookmarksopenfalse }
1277 ,bookmarksopen / true .code:n = { \Hy@bookmarksopentruetrue }
1278 ,bookmarksopen .default:n = {true}
1279 ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1280 ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1281 ,pdfcenterwindow .choice:
1282 ,pdfcenterwindow / false .code:n =
1283 {
1284   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1285 }
1286 ,pdfcenterwindow / true .code:n =
1287 {
1288   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1289 }
1290 ,pdfcenterwindow / .code:n =
1291 {
1292   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1293 }
1294 ,pdfcenterwindow / unknown .code:n =
1295 {
1296   \msg_warning:nnxx { hyp } { no-bool }
1297   { pdfcenterwindow }
1298   { \exp_not:n {#1} }
1299 }
1300 ,pdfcenterwindow .default:n = true
1301 ,pdfdirection .choice:
1302 ,pdfdirection / L2R .code:n =
1303 {
1304   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1305 }
1306 ,pdfdirection / R2L .code:n =
1307 {
1308   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1309 }
1310 ,pdfdirection / .code:n =
1311 {
1312   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1313 }
1314 ,pdfdirection / unknown .code:n =
1315 {
1316   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1317   { pdfdirection }
1318   { L2R , R2L }
1319   { \exp_not:n {#1} }
1320 }
1321 ,pdfdisplaydoctitle .choice:
1322 ,pdfdisplaydoctitle / false .code:n =
1323 {
1324   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1325 }
1326 ,pdfdisplaydoctitle / true .code:n =
1327 {
1328   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1329 }

```

```

1330 ,pdffdisplaydoctitle .default:n = true
1331 ,pdfduplex .choices:nn =
1332 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1333 {
1334   \pdf_version_compare:NnTF > {1.6}
1335   {
1336     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1337     { PrintDuplex } { /#1 }
1338   }
1339   {
1340     \msg_warning:nnxx
1341     {hyp}
1342     {ignore-deprecated-or-unknown-option-in-pdf-version}
1343     {pdfduplex}
1344     {\pdf_version:}
1345   }
1346 }%
1347 ,pdfduplex / .code:n =
1348 {
1349   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1350 }
1351 ,pdfduplex / unknown .code:n =
1352 {
1353   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1354   { pdfduplex }
1355   { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1356   { \exp_not:n {#1} }
1357 }
1358 ,pdffitwindow .choice:
1359 ,pdffitwindow / false .code:n =
1360 {
1361   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1362 }
1363 ,pdffitwindow / true .code:n =
1364 {
1365   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1366 }
1367 ,pdffitwindow / .code:n =
1368 {
1369   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1370 }
1371 ,pdffitwindow .default:n = true
1372 ,pdffitwindow / unknown .code:n =
1373 {
1374   \msg_warning:nnxx { hyp } { no-bool }
1375   { pdffitwindow }
1376   { \exp_not:n {#1} }
1377 }
1378 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1379 ,pdflinkmargin .initial:n = {1pt}
1380 ,pdfmenubar .choice:
1381 ,pdfmenubar / true .code:n =
1382 {
1383   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }

```

```

1384     }
1385 ,pdfmenubar / false .code:n =
1386     {
1387         \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1388         { HideMenubar } { true }
1389     }
1390 ,pdfmenubar / .code:n =
1391     {
1392         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1393     }
1394 ,pdfmenubar .default:n = true
1395 ,pdfmenubar / unknown .code:n =
1396     {
1397         \msg_warning:nnxx { hyp } { no-bool }
1398         { pdfmenubar }
1399         { \exp_not:n {#1} }
1400     }
1401 ,pdfnewwindow .choice:
1402 ,pdfnewwindow / true .code:n =
1403     {
1404         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1405         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1406     }
1407 ,pdfnewwindow / false .code:n =
1408     {
1409         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1410         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1411     }
1412 ,pdfnewwindow / .code:n =
1413     {
1414         \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1415         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1416     }
1417 ,pdfnonfullscreenpagemode .choices:nn =
1418     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1419     {
1420         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1421         { NonFullScreenPageMode } {/#1}
1422     }
1423 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1424     {
1425         \pdf_version_compare:NnTF < {1.6}
1426         {
1427             %message
1428         }
1429         {
1430             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1431             { NonFullScreenPageMode } {/UseAttachments}
1432         }
1433     }
1434 ,pdfnonfullscreenpagemode / .code:n =
1435     {
1436         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1437     }

```

```

1438 ,pdfnonfullscreenpagemode / unknown .code:n =
1439 {
1440     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1441     { pdfnonfullscreenpagemode }
1442     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1443     { \exp_not:n {#1} }
1444 }
1445 ,pdfnumcopies .code:n =
1446 {
1447     \pdf_version_compare:NnTF > {1.6}
1448     {
1449         \tl_if_empty:nTF {#1}
1450         {
1451             \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1452         }
1453         {
1454             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1455             {NumCopies}{#1}
1456         }
1457     }
1458     {
1459         \msg_warning:nnxx
1460         {hyp}
1461         {ignore-deprecated-or-unknown-option-in-pdf-version}
1462         {pdfnumcopies}
1463         {\pdf_version:}
1464     }
1465 }
1466 ,pdfpagelayout .choices:nn =
1467 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1468 { \pdfmanagement_add:nnx {Catalog} { PageLayout }{ /#1 } }
1469 ,pdfpagelayout / .code:n =
1470 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1471 ,pdfpagelayout / unknown .code:n =
1472 {
1473     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1474     { pdfpagelayout }
1475     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1476     { \exp_not:n {#1} }
1477 }
1478 ,pdfpagemode .choices:nn =
1479 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1480 { \pdfmanagement_add:nnx {Catalog} { PageMode }{ /#1 } }
1481 ,pdfpagemode / UseAttachments .code:n =
1482 {
1483     \pdf_version_compare:NnTF > {1.5}
1484     {
1485         \pdfmanagement_add:nnx {Catalog} { PageMode }{ /UseAttachments }
1486     }
1487     {
1488         \msg_warning:nnxx
1489         {hyp}
1490         {ignore-deprecated-or-unknown-value-in-pdf-version}
1491         {UseAttachments}

```



```

1492         {\pdf_version:}
1493     }
1494 }
1495 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1496 ,pdfpagemode / unknown .code:n =
1497 {
1498     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1499     { pdfpagemode }
1500     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1501     { \exp_not:n {#1} }
1502 }
1503 ,pdfpagescrop .code:n =
1504 {
1505     \tl_if_empty:nTF %or blank?
1506     {
1507         \pdfmanagement_remove:nn {Pages} { CropBox }
1508     }
1509     {
1510         \pdfmanagement_add:nnx {Pages} { CropBox } { [#1] }
1511     }
1512 }
1513 ,pdfpicktraybypdfsize .choice:
1514 ,pdfpicktraybypdfsize / true .code:n =
1515 {
1516     \pdf_version_compare:NnTF > {1.6}
1517     {
1518         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1519         { PickTrayByPDFSize } { true }
1520     }
1521     {
1522         \msg_warning:nnxx
1523         {hyp}
1524         {ignore-deprecated-or-unknown-option-in-pdf-version}
1525         {pdfpicktraybypdfsize}
1526         {\pdf_version:}
1527     }
1528 }
1529 ,pdfpicktraybypdfsize / false .code:n =
1530 {
1531     \pdf_version_compare:NnTF > {1.6}
1532     {
1533         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1534         { PickTrayByPDFSize } { false }
1535     }
1536     {
1537         \msg_warning:nnxx
1538         {hyp}
1539         {ignore-deprecated-or-unknown-option-in-pdf-version}
1540         {pdfpicktraybypdfsize}
1541         {\pdf_version:}
1542     }
1543 }
1544 ,pdfpicktraybypdfsize / .code:n =
1545 {

```

```

1546     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1547 }
1548 ,pdfpicktraybypdfsize / unknown .code:n =
1549 {
1550     \msg_warning:nnxx { hyp } { no-bool }
1551     { picktraybypdfsize }
1552     { \exp_not:n {#1} }
1553 }
1554 ,pdfprintarea .choices:nn =
1555 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1556 {
1557     \pdf_version_compare:NnTF < {2.0}
1558     {
1559         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1560         { PrintArea } { /#1 }
1561     }
1562     {
1563         \msg_warning:nnxx
1564         {hyp}
1565         {ignore-deprecated-or-unknown-option-in-pdf-version}
1566         {pdfprintarea}
1567         {\pdf_version:}
1568     }
1569 }%
1570 ,pdfprintarea / .code:n =
1571 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1572 ,pdfprintarea / unknown .code:n =
1573 {
1574     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1575     { pdfprintarea }
1576     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1577     { \exp_not:n {#1} }
1578 }
1579 ,pdfprintclip .choices:nn =
1580 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1581 {
1582     \pdf_version_compare:NnTF < {2.0}
1583     {
1584         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1585         { PrintClip } { /#1 }
1586     }
1587     {
1588         \msg_warning:nnxx
1589         {hyp}
1590         {ignore-deprecated-or-unknown-option-in-pdf-version}
1591         {pdfprintclip}
1592         {\pdf_version:}
1593     }
1594 }%
1595 ,pdfprintclip / .code:n =
1596 {
1597     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1598 }
1599 ,pdfprintclip / unknown .code:n =

```

```

1600     {
1601         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1602         { pdfprintclip }
1603         { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1604         { \exp_not:n {#1} }
1605     }
1606 ,pdfprintpagerange .code:n =
1607 {
1608     \pdf_version_compare:NnTF > {1.6}
1609     {
1610         \tl_if_empty:NTF { #1}
1611         {
1612             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1613             { PrintPageRange }
1614         }
1615         {
1616             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1617             {PrintPageRange}{[#1]}
1618         }
1619     }
1620     {
1621         \msg_warning:nnxx
1622         {hyp}
1623         {ignore-deprecated-or-unknown-option-in-pdf-version}
1624         {pdfprintpagerange}
1625         {\pdf_version:}
1626     }
1627 }
1628 ,pdfprintscaling .choices:nn =
1629 { None, AppDefault }
1630 {
1631     \pdf_version_compare:NnTF > {1.5}
1632     {
1633         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1634         { PrintScaling } { /#1 }
1635     }
1636     {
1637         \msg_warning:nnxx
1638         {hyp}
1639         {ignore-deprecated-or-unknown-option-in-pdf-version}
1640         {pdfprintscaling}
1641         {\pdf_version:}
1642     }
1643 }%
1644 ,pdfprintscaling / .code:n =
1645 {
1646     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1647 }
1648 ,pdfprintscaling / unknown .code:n =
1649 {
1650     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1651     { pdfprintarea }
1652     { None, AppDefault }
1653     { \exp_not:n {#1} }

```

```

1654     }
1655 ,pdfremotestartview .code:n =
1656 {
1657     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1658     \exp_args:NNV
1659     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1660     {
1661         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1662     }
1663     {
1664         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1665         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1666     }
1667 }
1668 ,pdfremotestartview .initial:n = {Fit}
1669 % pdfstartpage is special as it shares code with pdfstartview
1670 ,pdfstartpage .code:n =
1671 {
1672     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1673     \bool_if:nTF
1674     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1675     {
1676         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1677     }
1678     {
1679         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1680         {
1681             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1682         }
1683     }
1684 }
1685 ,pdfstartpage .initial:n =1
1686 ,pdfstartview .code:n =
1687 {
1688     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1689     \exp_args:NNV
1690     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1691     {
1692         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1693     }
1694     {
1695         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1696         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1697     }
1698     \bool_if:nTF
1699     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1700     {
1701         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1702     }
1703     {
1704         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1705         {
1706             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1707         }
1708     }

```

```

1708     }
1709 }
1710 ,pdfstartview .initial:n = Fit
1711 ,pdftoolbar .choice:
1712 ,pdftoolbar / true .code:n =
1713 {
1714     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1715 }
1716 ,pdftoolbar / false .code:n =
1717 {
1718     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1719     { HideToolbar } { true }
1720 }
1721 ,pdftoolbar / true .code:n =
1722 {
1723     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1724 }
1725 ,pdftoolbar .default:n = true
1726 ,pdftoolbar / unknown .code:n =
1727 {
1728     \msg_warning:nxxx { hyp } { no-bool }
1729     { pdftoolbar }
1730     { \exp_not:n {#1} }
1731 }
1732 % pdfview see below.
1733 ,pdfviewarea .choices:nn =
1734 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1735 {
1736     \pdf_version_compare:NnTF < {2.0}
1737     {
1738         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1739         { ViewArea } { /#1 }
1740     }
1741     {
1742         \msg_warning:nxxx
1743         {hyp}
1744         {ignore-deprecated-or-unknown-option-in-pdf-version}
1745         {pdfviewarea}
1746         {\pdf_version:}
1747     }
1748 }%
1749 ,pdfviewarea / .code:n =
1750 {
1751     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1752 }
1753 ,pdfviewarea / unknown .code:n =
1754 {
1755     \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1756     { pdfviewarea }
1757     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1758     { \exp_not:n {#1} }
1759 }
1760 ,pdfviewclip .choices:nn =
1761 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }

```

```

1762 {
1763   \pdf_version_compare:NnTF < {2.0}
1764   {
1765     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1766     { ViewClip } { /#1 }
1767   }
1768   {
1769     \msg_warning:nnxx
1770     {hyp}
1771     {ignore-deprecated-or-unknown-option-in-pdf-version}
1772     {pdfviewclip}
1773     {\pdf_version:}
1774   }
1775   }%
1776 ,pdfviewclip / .code:n =
1777 {
1778   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
1779 }
1780 ,pdfviewclip / unknown .code:n =
1781 {
1782   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1783   { pdfviewclip }
1784   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1785   { \exp_not:n {#1} }
1786 }
1787 ,pdfwindowui .choice:
1788 ,pdfwindowui / true .code:n =
1789 {
1790   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
1791 }
1792 ,pdfwindowui / false .code:n =
1793 {
1794   \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1795   { HideWindowUI } { true }
1796 }
1797 ,pdfwindowui / .code:n =
1798 {
1799   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
1800 }
1801 ,pdfwindowui / unknown .code:n =
1802 {
1803   \msg_warning:nnxx { hyp } { no-bool }
1804   { pdfwindowui }
1805   { \exp_not:n {#1} }
1806 }
1807 ,pdfwindowui .default:n = true
1808 }

```

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

```

1809 \keys_define:nn { hyp / setup }
1810 {
1811   ,pdfview .code:n =
1812   {
1813     \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}

```

```

1814 \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
1815 {
1816   { xyz }
1817   {
1818     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
1819     {
1820       \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
1821       \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
1822       {
1823         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1824       }
1825       {
1826         \tl_set:Nx \l__hyp_dest_pdfview_tl
1827         {
1828           \fp_eval:n { \l__hyp_tmpa_tl * 100 }
1829         }
1830       }
1831     }
1832     {
1833       \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1834     }
1835   }
1836   { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
1837   { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
1838   { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
1839   { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
1840   { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
1841   { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
1842   { fitr }
1843   {
1844     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
1845     {
1846       \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
1847     }
1848     {
1849       %ensure 4 values ...
1850       \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
1851       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1852       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1853       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1854       \hbox_set_to_wd:Nnn \l__hyp_dest_box
1855       {
1856         \fp_eval:n
1857         {
1858           round
1859           (
1860             abs
1861             (
1862               \seq_item:Nn\l__hyp_tmpa_seq{4}
1863               -
1864               (\seq_item:Nn\l__hyp_tmpa_seq{2})
1865             ),
1866           3
1867         )

```

```

1868         }bp
1869     }{}
1870     \box_set_dp:Nn \l__hyp_dest_box
1871     {
1872         \fp_eval:n
1873         {
1874             round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
1875         }bp
1876     }
1877     \box_set_ht:Nn \l__hyp_dest_box
1878     {
1879         \seq_item:Nn\l__hyp_tmpa_seq{5}bp
1880     }
1881 }
1882 }
1883 }
1884 {
1885     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
1886     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
1887 }
1888 }
1889 ,pdfview .initial:n = {xyz}
1890 }

```

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.

```

1891 \keys_define:nn { hyp / setup }
1892 {
1893     ,pdflang .code:n =
1894     {
1895         \tl_if_empty:nTF { #1 }
1896         {
1897             \pdfmanagement_remove:nn {Catalog} { Lang }
1898         }
1899         {
1900             \pdfmanagement_add:nnx {Catalog} { Lang } { (#1) }
1901         }
1902         \__hyp_store_metadata:nn {pdflang}{#1}
1903     }
1904 }

```

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*)

pdfcreator (*setup key*) 1905 \cs_new_protected:Npn __hyp_setup_info_key:nn #1 #2

pdfsubject (*setup key*) 1906 {

pdfproducer (*setup key*) 1907 \keys_define:nn { hyp / setup }

pdfkeywords (*setup key*) 1908 {


```

1909     pdf#1 .code:n =
1910     {
1911         \tl_if_blank:nTF {##1}
1912         {
1913             \pdfmanagement_remove:nn {Info}{#2}
1914         }
1915         {
1916             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
1917             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
1918             {
1919                 \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
1920             }
1921         }
1922         \__hyp_store_metadata:nn {pdf#1}{##1}
1923     }
1924 }
1925 \keys_define:nn { hyp / info }
1926 {
1927     #2 .code:n =
1928     {
1929         \tl_if_blank:nTF {##1}
1930         {
1931             \pdfmanagement_remove:nn {Info}{#2}
1932         }
1933         {
1934             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
1935             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
1936             {
1937                 \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
1938             }
1939         }
1940         \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
1941     }
1942 ,unknown .code:n =
1943 {
1944     \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
1945     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
1946     {
1947         \exp_args:Nno
1948         \pdfmanagement_add:nnx {Info}
1949         { \l_keys_key_str } {\l__hyp_tmpa_str}
1950     }
1951 }
1952 }
1953 }
1954 \__hyp_setup_info_key:nn {author} {Author}
1955 \__hyp_setup_info_key:nn {title} {Title}
1956 \__hyp_setup_info_key:nn {producer} {Producer}
1957 \__hyp_setup_info_key:nn {creator} {Creator}
1958 % ignored key: addtopdfcreator
1959 \__hyp_setup_info_key:nn {subject} {Subject}
1960 \__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.

```
1961 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
1962 {
1963   \keys_define:nn { hyp / setup }
1964   {
1965     pdf#1 .code:n =
1966     {
1967       \tl_if_blank:nTF {##1}
1968       {
1969         \pdfmanagement_remove:nn {Info}{#2}
1970       }
1971       {
1972         \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
1973       }
1974       \__hyp_store_metadata:nn {pdf#1}{##1}
1975     }
1976   }
1977   \keys_define:nn { hyp / info }
1978   {
1979     #2 .code:n =
1980     {
1981       \tl_if_blank:nTF {##1}
1982       {
1983         \pdfmanagement_remove:nn {Info}{#2}
1984       }
1985       {
1986         \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
1987       }
1988       \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
1989     }
1990   }
1991 }
1992
1993 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
1994 \__hyp_setup_info_date_key:nn {moddate} {ModDate}
```

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

```
1995 \keys_define:nn { hyp / setup }
1996 {
1997   ,pdftrapped .code:n =
1998   {
1999     \exp_args:Nne
2000     \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2001   }
2002   ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2003   {
2004     \pdfmanagement_add:nnx {Info}{Trapped}
2005     {/
2006       \str_uppercase:f { \str_head:n { #1 } }
2007       \str_lowercase:f { \str_tail:n { #1 } }
2008     }
2009     \__hyp_store_metadata:nx {pdftrapped}
2010     {
2011       \str_uppercase:f { \str_head:n { #1 } }
```

```

2012         \str_lowercase:f { \str_tail:n { #1 } }
2013     }
2014 }
2015 ,_pdftrapped / unknown .code:n =
2016 {
2017     \msg_warning:nnxxx { hyp } { unknown-choice }
2018     { pdftrapped }
2019     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2020     { \exp_not:n {#1} }
2021 }
2022 }

```

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

```

2023 \keys_define:nn { hyp / setup }
2024 {
2025     pdfinfo .code:n =
2026     {
2027         \keys_set:nn { hyp / info } { #1 }
2028     }
2029 }

```

Now we set some default values

```

2030 \keys_set:nn { hyp / setup } {pdfcreator = LaTeX-with-hyperref}
2031 \keys_set:nn { hyp / setup } {pdfauthor  = }
2032 \keys_set:nn { hyp / setup } {pdftitle   = }
2033 \keys_set:nn { hyp / setup } {pdfsubject = }

```

12.4 hyperxmp keys

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

```

2034 \clist_map_inline:nn
2035 {
2036     ,pdfcopyright
2037     ,pdftype
2038     ,pdflicenseurl
2039     ,pdfauthortitle
2040     ,pdfcaptionwriter
2041     ,pdfmetalang
2042     ,pdfapart
2043     ,pdfaconformance
2044     ,pdfuapart
2045     ,pdfxstandard
2046     ,pdfsource
2047     ,pdfdocumentid
2048     ,pdfinstanceid
2049     ,pdfversionid
2050     ,pdfrendition
2051     ,pdfpublication
2052     ,pdfpubtype
2053     ,pdfbytes
2054     ,pdfnumpages
2055     ,pdfissn

```

```

2056 ,pdfeissn
2057 ,pdfisbn
2058 ,pdfbookedition
2059 ,pdfpublisher
2060 ,pdfvolumenum
2061 ,pdfissuenum
2062 ,pdfpagerange
2063 ,pdfdoi
2064 ,pdfurl
2065 ,pdfidentifier
2066 ,pdfsubtitle
2067 ,pdfpubstatus
2068 ,pdfcontactaddress
2069 ,pdfcontactcity
2070 ,pdfcontactregion
2071 ,pdfcontactpostcode
2072 ,pdfcontactcountry
2073 ,pdfcontactphone
2074 ,pdfcontactemail
2075 ,pdfcontacturl
2076 ,pdfdate
2077 }
2078 {
2079   \keys_define:nn { hyp / setup }
2080   {
2081     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2082   }
2083 }

```

12.5 Transitions

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

```

2084 \keys_define:nn { hyp / setup }
2085 {
2086   pdfpageduration .code:n =
2087   {
2088     \tl_if_blank:nTF { #1 }
2089     {
2090       \pdfmanagement_remove:nn {Page}{Dur}
2091     }
2092     {
2093       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2094     }
2095   }
2096 }

```

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

setting looks e.g. like this

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

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

The keys allowed in the argument of pdfpagetransition are

```
2097 \keys_define:nn { hyp / setup }
2098 {
2099   pdfpagetransition .code:n =
2100   {
2101     \tl_if_blank:nTF {#1}
2102     {
2103       \pdfmanagement_remove:nn {Page}{Trans}
2104     }
2105     {
2106       \group_begin:
2107       \keys_set:nn { hyp / trans }{style=R,#1}
2108       \pdf_object_unnamed_write:nx { dict }
2109       {
2110         \pdfdict_use:n {l__hyp_page/Trans}
2111       }
2112       \pdfmanagement_add:nnx {Page}{Trans}{\pdf_object_ref_last:}
2113       \group_end:
2114     }
2115   }
2116 }
2117 \keys_define:nn { hyp / trans }
2118 {
2119   ,style .choices:nn =
2120   {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2121   { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2122   ,style / unknown .code:n =
2123   {
2124     \msg_warning:nnxxx { hyp } { unknown-choice }
2125     { trans / style }
2126     { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2127     { \exp_not:n {#1} }
2128   }
2129   ,duration .code:n =
```

```

2130     {
2131         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{#1}
2132     }
2133     ,direction .choices:nn =
2134     {H,V}
2135     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2136     ,direction .choices:nn =
2137     {0,90,180,270,315}
2138     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ #1 } }
2139     ,direction / None .code:n =
2140     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ /None } }
2141     ,direction / unknown .code:n =
2142     {
2143         \msg_warning:nnxxx { hyp } { unknown-choice }
2144         { trans / direction }
2145         {
2146             H~(horizontal,~only~Split,~Blinds),
2147             V~(vertical,~only~Split,~Blinds),
2148             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2149             90~(bottom~to~top,~only~Wipe),
2150             180~(right~to~left,~only~Wipe),
2151             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2152             315~(top~left~to~bottom,~only~Glitter),
2153             None~(only~Fly)
2154         }
2155         { \exp_not:n {#1} }
2156     }
2157     ,motion .choices:nn =
2158     {I,0}
2159     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2160     ,motion / unknown .code:n =
2161     {
2162         \msg_warning:nnxxx { hyp } { unknown-choice }
2163         { trans / motion }
2164         { I~(inwards) , 0~(outwards) }
2165         { \exp_not:n {#1} }
2166     }
2167     ,scale .code:n =
2168     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2169     ,opaque .choices:nn = {true,false}
2170     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2171     ,opaque / unknown .code:n =
2172     {
2173         \msg_warning:nnxxx { hyp } { unknown-choice }
2174         { trans / B }
2175         { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2176         { \exp_not:n {#1} }
2177     }
2178     % try to set unknown keys as style
2179     ,unknown .code:n =
2180     {
2181         % warning ...
2182         \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2183     }

```

```

2184 }
    Finally we process the package option list, to get most keys working
2185 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}
    Unfinished Form field code
2186 \NewDocumentCommand \MakeFieldObject { m m }
2187 {
2188   \pdfxform_new:nnn { #2 }{} { #1 }
2189 }
2190
2191
2192 \prop_new:N    \g__hyp_AcroForm_CoFields_prop
2193 \prop_new:N    \g__hyp_AcroForm_Fields_prop
2194
2195 \let\HyField@afields\ltx@empty
2196 \let\HyField@cofields\ltx@empty
2197 %% UF test for old pdftex removed
2198 %\let\HyField@AuxAddToFields\ltx@gobble
2199 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2200 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2201
2202 % the value doesn't matter, but with a prop we avoid duplicates and it is
2203 % clearly faster than removing them from a sequence
2204 \def\HyField@AuxAddToFields#1
2205 {
2206   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2207 }%
2208
2209 %fields with empty key get a value too -- lets hope that
2210 %this give the expected behaviour
2211 \def\HyField@AuxAddToCoFields #1 #2
2212 {
2213   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2214 }
2215
2216 \Hy@AtBeginDocument
2217 {
2218   \if@files
2219     \immediate\write\@mainaux{%
2220       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2221     }%
2222     \immediate\write\@mainaux{%
2223       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2224     }%
2225   \fi
2226   \let\HyField@AfterAuxOpen\@firstofone
2227 }%
2228
2229 \def\HyField@AddToFields
2230 {
2231   \exp_args:Nx\HyField__hypAddToFields
2232   {
2233     \pdfannot_box_ref_last:
2234   }

```

```

2235 \ifx\Fld@calculate@code\ltx@empty
2236 \else
2237 \begingroup
2238 \Hy@safe@activetrue
2239 \edef\Hy@temp{%
2240 \endgroup
2241 \if@files
2242 \write\@mainaux
2243 {
2244 \string\HyField@AuxAddToCoFields
2245 {
2246 \Fld@calculate@sortkey
2247 }
2248 {
2249 \pdfannot_box_ref_last:
2250 }
2251 }
2252 \fi
2253 }%
2254 \Hy@temp
2255 \fi
2256 }%
2257
2258 \def\HyField__hypAddToFields#1{
2259 \HyField@AfterAuxOpen{%
2260 \if@files
2261 \write\@mainaux{%
2262 \string\HyField@AuxAddToFields{#1}%
2263 }%
2264 \fi
2265 }%
2266 }%
2267
2268 \ExplSyntaxOff
2269 \ExplSyntaxOn
2270 \tl_new:N \l__hyp_CheckmarkYes_tl
2271 \tl_set:Nn \l__hyp_CheckmarkYes_tl { __hyp_xform_CheckMarkYes }
2272 \tl_new:N \l__hyp_CheckmarkOff_tl
2273 \tl_set:Nn \l__hyp_CheckmarkOff_tl { __hyp_xform_CheckMarkOff }
2274
2275 \def\@Form[#1]
2276 {
2277 \@ifundefined{textcolor}{\let\textcolor\@gobble}{ }
2278 \kvsetkeys{Form}{#1}
2279 \pdf@ifdraftmode{ }
2280 {
2281 \Hy@FormObjects
2282 \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2283 {
2284 \pdfmanagement_add:nxx { Catalog / AcroForm } { Fields }{##1}
2285 %\pdfmanagement_show:n { Catalog / AcroForm }
2286 }
2287 \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2288 {

```



```

2289         \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2290         {
2291             \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2292         }
2293     \seq_sort:Nn \l__hyp_tmpa_seq
2294     {
2295         \int_compare:nNnTF { \pdf@strcmp{##1}{##2} } > { 0 }
2296         { \sort_return_swapped: }
2297         { \sort_return_same: }
2298     }
2299     \seq_map_inline:Nn \l__hyp_tmpa_seq
2300     {
2301         \pdfmanagement_add:nnx { Catalog / AcroForm }
2302         { CO }
2303         {
2304             \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2305         }
2306     }
2307 }
2308 \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2309 {ZaDb} {\pdf_object_ref:n {l__hyp_font_zapdingbats_obj} }
2310 \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2311 {Helv} {\pdf_object_ref:n {l__hyp_font_helvetica_obj} }
2312 \pdfmanagement_add:nnx {Catalog /AcroForm}
2313 {DA}{(/Helv~10~Tf~0~g)}
2314 \pdfmeta_standard_verify:nTF {form_no_NeedAppearance}
2315 {
2316     \legacy_if:nT { HyField@NeedAppearances }
2317     {
2318         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2319     }
2320 }
2321 {
2322     \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2323 }
2324 }
2325 \MakeFieldObject
2326 {
2327     \group_begin:
2328     \fontfamily{pzd}
2329     \fontencoding{U}
2330     \fontseries{m}
2331     \fontshape{n}
2332     \selectfont
2333     \char123
2334     \group_end:
2335 }
2336 {__hyp_xform_Ding}
2337 \MakeFieldObject
2338 {
2339     \group_begin:
2340     \fontfamily{pzd}
2341     \fontencoding{U}
2342     \fontseries{m}

```

```

2343         \fontshape{n}
2344         \selectfont
2345         \phantom{\char123}
2346         \group_end:
2347     }
2348     {__hyp_xform_DingOff}
2349 \MakeFieldObject
2350 {
2351     \group_begin:
2352     \fontfamily{pzd}
2353     \fontencoding{U}
2354     \fontseries{m}
2355     \fontshape{n}
2356     \selectfont
2357     \char51
2358     \group_end:
2359 }
2360 {__hyp_xform_CheckMarkYes}
2361 \MakeFieldObject
2362 {
2363     \group_begin:
2364     \fontfamily{pzd}
2365     \fontencoding{U}
2366     \fontseries{m}
2367     \fontshape{n}
2368     \selectfont
2369     \phantom{\char51} %perhaps xetex needs some small glyph ..
2370     \group_end:
2371 }
2372 {__hyp_xform_CheckMarkOff}
2373 \MakeFieldObject
2374 {
2375     \fbox{\textcolor{yellow}{\textsf{Submit}}}} %color?
2376 }
2377 {__hyp_xform_Submit}
2378 \MakeFieldObject
2379 {
2380     \fbox{\textcolor{yellow}{\textsf{SubmitP}}}} %color?
2381 }
2382 {__hyp_xform_SubmitP}
2383 }
2384 \ExplSyntaxOff
2385 \let\@endForm\ltx@empty
2386 \let\HyAnn@AbsPageLabel\ltx@empty
2387 \let\Fld@pageobjref\ltx@empty
2388
2389 \ExplSyntaxOn
2390 \newcount\HyAnn@Count
2391 \HyAnn@Count=\ltx@zero
2392 \def\HyAnn@AbsPageLabel
2393 {
2394     \global\advance\HyAnn@Count by\ltx@one
2395     %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2396     %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}

```

```

2397     \%zref@refused{HyAnn@\the\HyAnn@Count}%
2398     \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
2399     \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
2400 }%
2401 \def\Fld@pageobjref
2402 {
2403     \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
2404     {
2405         /P~\pdf_pageobject_ref:n
2406         {
2407             \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
2408         }
2409     }
2410 }
2411 \ExplSyntaxOff
2412 \ExplSyntaxOn
2413 %% check if the attr should be set through
2414 %% hooks.
2415 %% check if options are missing.
2416 \def\@TextField[#1]#2{% parameters, label
2417     \def\Fld@name{#2}%
2418     \let\Fld@default\ltx@empty
2419     \let\Fld@value\@empty
2420     \def\Fld@width{\DefaultWidthofText}%
2421     \def\Fld@height{%
2422         \ifFld@multiline
2423             \DefaultHeightofTextMultiline
2424         \else
2425             \DefaultHeightofText
2426         \fi
2427     }%
2428     \begingroup
2429     \expandafter\HyField@SetKeys\expandafter{%
2430         \DefaultOptionsofText,#1%
2431     }%
2432     \PDFForm@Name
2433     \HyField@FlagsText
2434     \ifFld@hidden\def\Fld@width{1sp}\fi
2435     \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2436     \LayoutTextField{#2}{%
2437         \leavevmode
2438         \HyAnn@AbsPageLabel
2439         \Hy@escapeform\PDFForm@Text
2440         \pdfannot_box:nnnn
2441         {\Fld@width}
2442         {\Fld@height}
2443         {Opt} %is this correct?
2444         {\PDFForm@Text}
2445         \MakeTextField{\Fld@width}{\Fld@height}
2446         \HyField@AddToFields
2447     }%
2448     \endgroup
2449 }
2450 \providecommand\@curropt{}

```

```

2451 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2452 \def\Fld@name{#2}
2453 \let\Fld@default\relax
2454 \let\Fld@value\relax
2455 \def\Fld@width{\DefaultWidthofChoiceMenu}
2456 \def\Fld@height{\DefaultHeightofChoiceMenu}
2457 \begingroup
2458 \Fld@menulength=0 %
2459 \@tempdima\z@
2460 \clist_map_variable:nNn { #3 } \@curropt
2461 %\@for\@curropt:=#3\do
2462 {%
2463 \expandafter\Fld@checkequals\@curropt==\%
2464 \Hy@StepCount\Fld@menulength
2465 \settowidth{\@tempdimb}{\@currDisplay}%
2466 \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2467 }%
2468 \advance\@tempdima by~15\p@
2469 \begingroup
2470 \HyField@SetKeys{#1}
2471 \edef\x{\endgroup
2472 \noexpand\expandafter
2473 \noexpand\HyField@SetKeys
2474 \noexpand\expandafter{%
2475 \expandafter\noexpand\csname DefaultOptionsof%
2476 \ifFld@radio
2477 Radio%
2478 \else
2479 \ifFld@combo
2480 \ifFld@popdown
2481 PopdownBox%
2482 \else
2483 ComboBox%
2484 \fi
2485 \else
2486 ListBox%
2487 \fi
2488 \fi
2489 \endcsname
2490 }%
2491 }\x
2492 \HyField@SetKeys{#1}%
2493 \PDFForm@Name
2494 \ifFld@hidden\def\Fld@width{1sp}\fi
2495 \ifx\Fld@value\relax
2496 \let\Fld@value\Fld@default
2497 \fi
2498 \LayoutChoiceField{#2}{%
2499 \ifFld@radio
2500 \HyField@FlagsRadioButton
2501 \__hypRadio{#3}%
2502 \else
2503 \begingroup
2504 \HyField@FlagsChoice

```

```

2505         \ifdim\Fld@width<\@tempdima
2506         \ifdim\@tempdima<1cm\@tempdima1cm\fi
2507         \edef\Fld@width{\the\@tempdima}%
2508     \fi
2509     \ifFld@combo
2510     \else
2511         \@tempdima=\the\Fld@menulength\Fld@charsize
2512         \advance\@tempdima by~\Fld@borderwidth bp %
2513         \advance\@tempdima by~\Fld@borderwidth bp %
2514         \edef\Fld@height{\the\@tempdima}%
2515     \fi
2516     \__hypListBox{#3}%
2517 \endgroup
2518 \fi
2519 }%
2520 \endgroup
2521 }
2522 \tl_new:N \l__hyp_RadioYes_tl
2523 \tl_set:Nn \l__hyp_RadioYes_tl { __hyp_xform_Ding }
2524 \def\__hypRadio#1{%
2525     \Fld@listcount=0-%
2526     %\show\Fld@default
2527     \EdefEscapeName\Fld@default{\Fld@default}%
2528     \clist_map_variable:nNn { #1 } \@curropt
2529     %\@for\@curropt:=#1\do
2530     {%
2531         \expandafter\Fld@checkequals\@curropt==\%
2532         \EdefEscapeName\@currValue{\@currValue}%
2533         \Hy@StepCount\Fld@listcount
2534         \@currDisplay\space
2535         \leavevmode
2536         \HyAnn@AbsPageLabel
2537         \Hy@escapeform\PDFForm@Radio
2538         \pdfannot_box:nnnn
2539             {\Fld@width}
2540             {\Fld@height}
2541             {0pt} %is this correct?
2542             {
2543                 \PDFForm@Radio
2544                 /AP
2545                 <<
2546                 /N
2547                 <<
2548                 /\@currValue\c_space_tl \pdfxform_ref:o {__hyp_xform_Ding}
2549                 %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2550                 >>
2551                 >>
2552             }
2553             {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2554     \int_compare:nNnT { \Fld@listcount } = { 1 }
2555     { \HyField@AddToFields }
2556     \c_space_tl % deliberate space between radio buttons
2557     % to do: --> should be configurable
2558 }%

```

```

2559 }
2560
2561 \newcount\Fld@listcount
2562 \def\__hypListbox#1
2563 {
2564   \HyField@PDFChoices{#1}
2565   \mode_leave_vertical:
2566   \HyAnn@AbsPageLabel
2567   \Hy@escapeform\PDFForm@List
2568   \pdf_link_user:nnn
2569   {widget} %perhaps we need more types??
2570   {\PDFForm@List}
2571   {\MakeChoiceField{\Fld@width}{\Fld@height}}
2572   \HyField@AddToFields
2573 }
2574
2575
2576 \def\@PushButton[#1]#2{% parameters, label
2577   \def\Fld@name{#2}%
2578   \group_begin:
2579     \exp_args:No\HyField@SetKeys
2580     {
2581       \DefaultOptionsofPushButton,#1
2582     }
2583     \PDFForm@Name
2584     \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2585     {
2586       \HyField@FlagsPushButton
2587       \legacy_if:nT {Fld@hidden}
2588       {
2589         \def\Fld@width{1sp}
2590       }
2591       \LayoutPushButtonField
2592       {
2593         \mode_leave_vertical:
2594         \HyAnn@AbsPageLabel
2595         \Hy@escapeform\PDFForm@Push
2596         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2597         \pdfannot_box:nnnn
2598         {\box_wd:N\l_tmpa_box}
2599         {\box_ht:N\l_tmpa_box}
2600         {\box_dp:N\l_tmpa_box} %is this correct?
2601         {\PDFForm@Push}
2602         {\box_use:N\l_tmpa_box}
2603         \HyField@AddToFields
2604       }
2605     }
2606     {
2607       \msg_error:nn { hyp }{ pdfa-no-push-button }
2608       \LayoutPushButtonField
2609       {
2610         \mode_leave_vertical:
2611         \MakeButtonField{#2}
2612       }

```

```

2613     }
2614   \group_end:
2615 }
2616
2617 \def\@Submit[#1]#2
2618 {
2619   \def\Fld@width {\DefaultWidthofSubmit}
2620   \def\Fld@height{\DefaultHeightofSubmit}
2621   \group_begin:
2622     \exp_args:No\HyField@SetKeys
2623     {
2624       \DefaultOptionsofSubmit,#1
2625     }
2626   \HyField@FlagsPushButton
2627   \HyField@FlagsSubmit
2628   \legacy_if:nT { Fld@hidden }
2629   {
2630     \def\Fld@width{1sp}
2631   }
2632   \mode_leave_vertical:
2633   \HyAnn@AbsPageLabel
2634   \Hy@escapeform\PDFForm@Submit
2635   \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2636   \pdfannot_box:nnnn
2637     {\box_wd:N\l_tmpa_box}
2638     {\box_ht:N\l_tmpa_box}
2639     {\box_dp:N\l_tmpa_box} %is this correct?
2640   {
2641     \PDFForm@Submit
2642     /AP<<
2643       /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2644       /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2645     >>
2646   }
2647   \HyField@AddToFields
2648   \box_use:N\l_tmpa_box
2649
2650   \group_end:
2651 }
2652
2653 \def\@Reset[#1]#2
2654 {
2655   \def\Fld@width {\DefaultWidthofReset}
2656   \def\Fld@height{\DefaultHeightofReset}
2657   \group_begin:
2658     \exp_args:No\HyField@SetKeys
2659     {
2660       \DefaultOptionsofReset,#1
2661     }
2662   \mode_leave_vertical:
2663   \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2664   {
2665     \HyField@FlagsPushButton
2666     \legacy_if:nT { Fld@hidden }

```

```

2667         { \def\Fld@width{1sp} }
2668         \HyAnn@AbsPageLabel
2669         \Hy@escapeform\PDFForm@Reset
2670         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2671         \pdfannot_box:nnnn
2672         {\box_wd:N\l_tmpa_box}
2673         {\box_ht:N\l_tmpa_box}
2674         {\box_dp:N\l_tmpa_box} %is this correct?
2675         { \PDFForm@Reset }
2676         \HyField@AddToFields
2677         \box_use:N \l_tmpa_box
2678     }
2679     {
2680         \msg_error:nn { hyp }{ pdfa-no-reset-button }
2681         \MakeButtonField{#2}
2682     }
2683 \group_end:
2684 }
2685
2686 \def\@CheckBox[#1]#2
2687 {% parameters, label
2688     \def\Fld@name{#2}
2689     \def\Fld@default{0}
2690     \group_begin:
2691         \def\Fld@width {\DefaultWidthofCheckBox}
2692         \def\Fld@height{\DefaultHeightofCheckBox}
2693         \exp_args:No\HyField@SetKeys
2694         {
2695             \DefaultOptionsofCheckBox,#1
2696         }
2697         \PDFForm@Name
2698         \HyField@FlagsCheckBox
2699         \legacy_if:nT { Fld@hidden }
2700         {
2701             \def\Fld@width{1sp}
2702         }
2703         \LayoutCheckField{#2}
2704         {
2705             \mode_leave_vertical:
2706             \HyAnn@AbsPageLabel
2707             \Hy@escapeform\PDFForm@Check
2708             \pdfannot_box:nnnn
2709             {\Fld@width}
2710             {\Fld@height}
2711             {Opt} %is this correct?
2712             {\PDFForm@Check}
2713             \HyField@AddToFields %check if this works with xelatex ...
2714         }
2715     \group_end:
2716 }
2717 \ExplSyntaxOff
2718
2719 %hm. Should a luatex driver use type1 fonts in fields????
2720 \ExplSyntaxOn

```



```

2721 \def\Hy@FormObjects
2722 {
2723   \pdf_object_new:nn {l__hyp_encoding_pdfdoc_obj } { dict }
2724   \pdf_object_new:nn {l__hyp_font_zapfdingbats_obj } { dict }
2725   \pdf_object_new:nn {l__hyp_font_helvetica_obj } { dict }
2726   \pdf_object_write:nx {l__hyp_encoding_pdfdoc_obj }
2727   {
2728     /Type/Encoding
2729     /Differences[
2730       24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2731       /ring/tilde
2732       \c_space_tl
2733       39/quotesingle
2734       \c_space_tl
2735       96/grave %
2736       \iow_newline:
2737       128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2738       /fraction/guilsinglleft/guilsinglright/minus/perthousand
2739       /quotedblbase/quotedblleft/quotedblright/quoteleft
2740       /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2741       /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2742       \iow_newline:
2743       164/currency
2744       \c_space_tl
2745       166/brokenbar
2746       \c_space_tl
2747       168/dieresis/copyright/ordfeminine
2748       \c_space_tl
2749       172/logicalnot/.notdef/registered/macron/degree/plusminus
2750       /twosuperior/threesuperior/acute/mu
2751       \c_space_tl
2752       183/periodcentered/cedilla/onesuperior/ordmasculine
2753       \c_space_tl
2754       188/onequarter/onehalf/threequarters
2755       \iow_newline:
2756       192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
2757       /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
2758       /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
2759       /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
2760       /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
2761       /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
2762       /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
2763       /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
2764       /ocircumflex/otilde/odieresis/divide/oslash/ugrave
2765       /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
2766     ]
2767   }
2768   \pdf_object_write:nn {l__hyp_font_zapfdingbats_obj }
2769   {
2770     /Type/Font
2771     /Subtype/Type1
2772     /Name/ZaDb
2773     /BaseFont/ZapfDingbats
2774   }

```

```

2775 \pdf_object_write:nx {l__hyp_font_helvetica_obj }
2776 {
2777     /Type/Font
2778     /Subtype/Type1
2779     /Name/Helv
2780     /BaseFont/Helvetica
2781     /Encoding~\pdf_object_ref:n { l__hyp_encoding_pdfdoc_obj }
2782 }
2783 \global\let\Hy@FormObjects\relax
2784 }
2785 \ExplSyntaxOff
2786 \providecommand*{\Fld@pageobjref}{}
2787 \ifcsname pdf@escapestring\endcsname
2788     \def\Hy@escapeform#1{%
2789         \ifHy@pdfescapeform
2790             \let\Hy@escapestring\pdfescapestring
2791         \else
2792             \let\Hy@escapestring\@firstofone
2793         \fi
2794     }%
2795     \Hy@escapeform{}%
2796 \else
2797     \let\Hy@escapestring\@firstofone
2798     \def\Hy@escapeform#1{%
2799         \ifHy@pdfescapeform
2800             \def\Hy@escapestring##1{%
2801                 \noexpand\Hy@escapestring{\noexpand##1}%
2802             }%
2803             \edef\Hy@temp{#1}%
2804             \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{}\@nil
2805             \def\Hy@escapestring##1{%
2806                 \@ifundefined{Hy@esc@\string##1}{%
2807                     ##1%
2808                     \ThisShouldNotHappen
2809                 }{%
2810                     \csname Hy@esc@\string##1\endcsname
2811                 }%
2812             }%
2813         \else
2814             \let\Hy@escapestring\@firstofone
2815         \fi
2816     }%
2817     \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
2818         \ifx\\#3\\%
2819             \else
2820                 \expandafter
2821                 \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
2822                 \ltx@ReturnAfterFi{%
2823                     \Hy__hypescapeform#3\@nil
2824                 }%
2825             \fi
2826         }%
2827     \fi
2828     \def\PDFForm@Name{%

```

```

2829 \PDFForm__hypName\Fld@name
2830 \ifx\Fld@altname\relax
2831 \else
2832 \PDFForm__hypName\Fld@altname
2833 \fi
2834 \ifx\Fld@mappingname\relax
2835 \else
2836 \PDFForm__hypName\Fld@mappingname
2837 \fi
2838 }
2839 \def\PDFForm__hypName#1{%
2840 \begingroup
2841 \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
2842 \ifHy@unicode
2843 \Hy@unicodedefalse
2844 \fi
2845 \fi
2846 \pdfstringdef\Hy@gtemp#1%
2847 \endgroup
2848 \let#1\Hy@gtemp
2849 }
2850 \def\Fld@X@additionalactions{%
2851 \ifx\Fld@keystroke@code\@empty
2852 \else
2853 /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
2854 \fi
2855 \ifx\Fld@format@code\@empty
2856 \else
2857 /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
2858 \fi
2859 \ifx\Fld@validate@code\@empty
2860 \else
2861 /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
2862 \fi
2863 \ifx\Fld@calculate@code\@empty
2864 \else
2865 /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
2866 \fi
2867 \ifx\Fld@onfocus@code\@empty
2868 \else
2869 /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
2870 \fi
2871 \ifx\Fld@onblur@code\@empty
2872 \else
2873 /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
2874 \fi
2875 \ifx\Fld@onmousedown@code\@empty
2876 \else
2877 /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
2878 \fi
2879 \ifx\Fld@onmouseup@code\@empty
2880 \else
2881 /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
2882 \fi

```

```

2883 \ifx\Fld@onenter@code\@empty
2884 \else
2885   /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
2886 \fi
2887 \ifx\Fld@onexit@code\@empty
2888 \else
2889   /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
2890 \fi
2891 }
2892 \ExplSyntaxOn
2893 \def\Fld@additionalactions
2894 {%
2895   \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{}
2896   {
2897     \pdfmeta_standard_verify:nT {annot_widget_no_AA}
2898     {/AA<<\Fld@X@additionalactions>>}
2899   }
2900 }
2901 \ExplSyntaxOff
2902 \def\Fld@annotnames{%
2903   /T(\Fld@name)%
2904   \ifx\Fld@altname\relax
2905   \else
2906     /TU(\Fld@altname)%
2907   \fi
2908   \ifx\Fld@mappingname\relax
2909   \else
2910     /TM(\Fld@mappingname)%
2911   \fi
2912 }
2913 \ExplSyntaxOn
2914 \def\PDFForm@Check
2915 {
2916   /Subtype/Widget
2917   ~\Fld@annotflags
2918   ~\Fld@pageobjref
2919   ~\Fld@annotnames
2920   /FT/Btn
2921   \Fld@flags
2922   /Q~\Fld@align
2923   /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
2924   /AP
2925   <<
2926   /N
2927   <<
2928     /Yes~\pdfxform_ref:o{\l__hyp_CheckmarkYes_tl}
2929     /Off~\pdfxform_ref:o{\l__hyp_CheckmarkOff_tl}
2930   >>
2931   >>
2932   /MK<<
2933     \int_compare:nNnF {\Fld@rotation}={0}
2934     {
2935       /R~\Fld@rotation
2936     }

```

```

2937 \tl_if_empty:NF\Fld@bordercolor
2938 {
2939   /BC[\Fld@bordercolor]
2940 }
2941 \tl_if_empty:NF\Fld@bcolor
2942 {
2943   /BG[\Fld@bcolor]
2944 }
2945 /CA(\Hy@escapestring{\Fld@cbsymbol})%
2946 >>
2947 /DA
2948 (
2949   /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
2950   \tl_if_empty:NF \Fld@color
2951   {
2952     \c_space_tl \Fld@color
2953   }
2954 )
2955 /H/P
2956 \legacy_if:nTF {Fld@checked}
2957 {
2958   /V/Yes /AS/Yes
2959 }
2960 {
2961   /V/Off /AS/Off
2962 }
2963 \Fld@additionalactions
2964 }
2965 \ExplSyntaxOff
2966 \ExplSyntaxOn
2967 \def\PDFForm@Push
2968 {
2969   /Subtype/Widget
2970   ~\Fld@annotflags
2971   ~\Fld@pageobjref
2972   ~\Fld@annotnames
2973   /FT/Btn
2974   ~\Fld@flags
2975   /H/P
2976   /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
2977   \bool_if:nT
2978   {
2979     !\int_compare_p:nNn {\Fld@rotation} = {0}
2980     ||
2981     \tl_if_exist_p:N \Fld@bordercolor
2982   }
2983   {
2984     /MK
2985     <<
2986     \int_compare:nNnF {\Fld@rotation} = {0}
2987     {
2988       /R~\Fld@rotation
2989     }
2990     \tl_if_exist:NT \Fld@bordercolor

```

```

2991         {
2992             /BC[\Fld@bordercolor]
2993         }
2994     >>
2995 }
2996 /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
2997 \Fld@additionalactions
2998 }
2999
3000 \ExplSyntaxOff
3001 \def\PDFForm@List{%
3002     /Subtype/Widget%
3003     \Fld@annotflags
3004     \Fld@pageobjref
3005     \Fld@annotnames
3006     /FT/Ch%
3007     \Fld@flags
3008     /Q \Fld@align
3009     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3010     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3011         \ifx\Fld@bordercolor\relax\else 1\fi
3012         \ifx\Fld@bcolor\relax \else 1\fi
3013         \space
3014     \else
3015         /MK<<%
3016         \ifnum\Fld@rotation=\z@
3017         \else
3018             /R \Fld@rotation
3019         \fi
3020         \ifx\Fld@bordercolor\relax
3021         \else
3022             /BC[\Fld@bordercolor]%
3023         \fi
3024         \ifx\Fld@bcolor\relax
3025         \else
3026             /BG[\Fld@bcolor]%
3027         \fi
3028         >>%
3029     \fi
3030 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3031     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3032 \Fld@choices
3033 \Fld@additionalactions
3034 }
3035 \ExplSyntaxOn
3036 \def\PDFForm@Radio
3037 {
3038     /Subtype/Widget
3039     ~\Fld@annotflags
3040     ~\Fld@pageobjref
3041     ~\Fld@annotnames
3042     /FT/Btn
3043     \Fld@flags
3044     /H/P

```

```

3045 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3046 /MK<<
3047 \ifnum\Fld@rotation=\z@
3048 \else
3049 /R~\Fld@rotation
3050 \fi
3051 \ifx\Fld@bordercolor\relax
3052 \else
3053 /BC[\Fld@bordercolor]%
3054 \fi
3055 \ifx\Fld@bcolor\relax
3056 \else
3057 /BG[\Fld@bcolor]%
3058 \fi
3059 /CA(\Hy@escapestring{\Fld@radiosymbol})%
3060 >>
3061 /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3062 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3063 \ifx\Fld@default\@empty
3064 /V/Off%
3065 /DV/Off%
3066 \else
3067 /V/\Fld@default
3068 /DV/\Fld@default
3069 \fi
3070 \Fld@additionalactions
3071 }
3072 \ExplSyntaxOff
3073 \ExplSyntaxOn
3074 % Does an appeareance dict make sense here?
3075 \def\PDFForm@Text
3076 {
3077 /Subtype/Widget
3078 ~\Fld@annotflags
3079 ~\Fld@pageobjref
3080 ~\Fld@annotnames
3081 /FT/Tx
3082 ~\Fld@flags
3083 /Q~\Fld@align
3084 /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3085 \bool_if:nT
3086 {
3087 !\int_compare_p:nNn {\Fld@rotation} = {0}
3088 ||
3089 \tl_if_exist_p:N \Fld@bordercolor
3090 ||
3091 \tl_if_exist_p:N \Fld@bcolor
3092 }
3093 {
3094 /MK
3095 <<
3096 \int_compare:nNnF {\Fld@rotation} = {0}
3097 {
3098 /R~\Fld@rotation

```

```

3099         }
3100         \tl_if_exist:NT \Fld@bordercolor
3101         {
3102             /BC[\Fld@bordercolor]
3103         }
3104         \tl_if_exist:NT \Fld@bcolor
3105         {
3106             /BG[\Fld@bcolor]
3107         }
3108     >>
3109 }
3110 /DA
3111 (
3112     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3113     \tl_if_empty:NF {\c_space_tl\Fld@color}
3114 )
3115 /DV(\Hy@escapestring{\Fld@default})
3116 /V(\Hy@escapestring{\Fld@value})
3117 ~\Fld@additionalactions
3118 \int_compare:nNnT { \Fld@maxlen}>{0}
3119 {
3120     /MaxLen~\Fld@maxlen
3121 }
3122 }
3123 \ExplSyntaxOff
3124
3125 \def\PDFForm@Submit{%
3126     /Subtype/Widget%
3127     \Fld@annotflags
3128     \Fld@pageobjref
3129     \Fld@annotnames
3130     /FT/Btn%
3131     \Fld@flags
3132     /H/P%
3133     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3134     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3135         \ifx\Fld@bordercolor\relax\else 1\fi
3136         \space
3137     \else
3138         /MK<<%
3139         \ifnum\Fld@rotation=\z@
3140         \else
3141             /R \Fld@rotation
3142         \fi
3143         \ifx\Fld@bordercolor\relax
3144         \else
3145             /BC[\Fld@bordercolor]%
3146         \fi
3147     >>%
3148     \fi
3149     /A<<%
3150     /S/SubmitForm%
3151     /F<<%
3152     /FS/URL%

```



```

3153         /F(\Hy@escapestring{\Form@action})%
3154     >>%
3155     \Fld@submitflags
3156 >>%
3157     \Fld@additionalactions
3158 }
3159 \ExplSyntaxOn
3160     \def\PDFForm@Reset{%
3161         /Subtype/Widget%
3162         \Fld@annotflags
3163         \Fld@pageobjref
3164         \Fld@annotnames
3165         /FT/Btn%
3166         \Fld@flags
3167         /H/P%
3168         /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3169         \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3170             \ifx\Fld@bordercolor\relax\else 1\fi
3171             \space
3172     \else
3173         /MK<<%
3174         \ifnum\Fld@rotation=\z@
3175         \else
3176             /R~\Fld@rotation
3177         \fi
3178         \ifx\Fld@bordercolor\relax
3179         \else
3180             /BC[\Fld@bordercolor]%
3181         \fi
3182     >>%
3183     \fi
3184     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3185     /A<</S/ResetForm>>%
3186     \Fld@additionalactions
3187 }%
3188
3189
3190 %these patterns are used in hyperref checks.
3191 %it is unclear if they are really useful and if a backend support is
3192 %needed.
3193 \str_case:VnF \c_sys_backend_str
3194 {
3195     { pdfmode }
3196     {
3197         \def\HyPat@ObjRef
3198         {
3199             [0-9]*[1-9][0-9]*~0~R
3200         }
3201     }
3202     { dvipdfmx }
3203     {
3204         \def\HyPat@ObjRef
3205         {
3206             @[\~]+

```

```

3207     }
3208   }
3209   { xdvipdfmx }
3210   {
3211     \def\HyPat@ObjRef
3212     {
3213       @[\~]+
3214     }
3215   }
3216 }
3217 { %also set in hyperref sty, so probably not needed.
3218   \def\HyPat@ObjRef/{.+}
3219 }
3220
3221
3222 \ExplSyntaxOff
3223 % UF: removed Hy@writebookmark
3224 %   \Hy@currentbookmarklevel{0}
3225 %   \Hy@numberline
3226 %   \__hypwritetorep
3227 %   counter{bookmark@seq@number}
3228 % removed \HyPsd@SanitizeForOutFile, not needed
3229 % removed \currentpdfbookmark, defined by bookmark,
3230 % should use \newcommand there
3231 % removed \subpdfbookmark, defined by bookmark,
3232 % should use \newcommand there
3233 % removed \belowpdfbookmark, defined by bookmark,
3234 % should use \newcommand there
3235 % removed \pdfbookmark, defined by bookmark,
3236 % \BOOKMARK
3237 % \@BOOKMARK
3238 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3239 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3240 %% removed \ReadBookmarks / unneeded with bookmark.
3241 %% removed \Hy@OutlineName
3242 %% removed \check@bm@number
3243 %% removed \calc@bm@number
3244
3245 \ifHy@implicit
3246 \else
3247   \expandafter\endinput
3248 \fi
3249 \newlength\Hy@SectionHShift
3250 \def\Hy@SectionAnchorHref#1{%
3251   \ifx\protect\@typeset@protect
3252     \Hy__hypSectionAnchor{#1}%
3253   \fi
3254 }
3255 \DeclareRobustCommand*{\Hy__hypSectionAnchor}[1]{%
3256   \leavevmode
3257   \hbox to 0pt{%
3258     \kern-\Hy@SectionHShift
3259     \Hy@raisedlink{%
3260       \hyper@anchorstart{#1}\hyper@anchorend

```

```

3261     }%
3262     \hss
3263 }%
3264 }
3265 \let\H@old@ssect\@ssect
3266 \def\@ssect#1#2#3#4#5{%
3267   \Hy@MakeCurrentHrefAuto{section*}%
3268   \setlength{\Hy@SectionHShift}{#1}%
3269   \begingroup
3270     \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3271     \toks\tw@ \expandafter{%
3272       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3273       #5%
3274     }%
3275   \edef\x{\endgroup
3276     \the\toks@{\the\toks\tw@}%
3277   }\x
3278 }
3279 \let\H@old@schapter\@schapter
3280 \def\@schapter#1{%
3281   \begingroup
3282     \let\@mkboth\@gobbletwo
3283     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3284     \Hy@raisedlink{%
3285       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3286     }%
3287   \endgroup
3288   \H@old@schapter{#1}%
3289 }
3290 \ltx@ifundefined{@chapter}{}{%
3291   \let\Hy@org@chapter\@chapter
3292   \def\@chapter{%
3293     \def\Hy@next{%
3294       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3295       \Hy@raisedlink{%
3296         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3297       }%
3298     }%
3299     \ifnum\c@secnumdepth>\m@ne
3300       \ltx@ifundefined{if@mainmatter}%
3301       \iftrue{\csname if@mainmatter\endcsname}%
3302       \let\Hy@next\relax
3303     \fi
3304     \fi
3305     \Hy@next
3306     \Hy@org@chapter
3307   }%
3308 }
3309 \let\H@old@part\@part
3310 \begingroup\expandafter\expandafter\expandafter\endgroup
3311 \expandafter\ifx\csname chapter\endcsname\relax
3312   \let\Hy@secnum@part\z@
3313 \else
3314   \let\Hy@secnum@part\m@ne

```

```

3315 \fi
3316 \def\@part{%
3317   \ifnum\Hy@secnum@part>\c@secnumdepth
3318     \phantomsection
3319   \fi
3320   \H@old@part
3321 }
3322 \let\H@old@spart\@spart
3323 \def\@spart#1{%
3324   \Hy@MakeCurrentHrefAuto{part*}%
3325   \Hy@raisedlink{%
3326     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3327   }%
3328   \H@old@spart{#1}%
3329 }
3330 \let\H@old@sect\@sect
3331 \def\@sect#1#2#3#4#5#6[#7]#8{%
3332   \ifnum #2>\c@secnumdepth
3333     \expandafter\@firstoftwo
3334   \else
3335     \expandafter\@secondoftwo
3336   \fi
3337   {%
3338     \Hy@MakeCurrentHrefAuto{section*}%
3339     \setlength{\Hy@SectionHShift}{#3}%
3340     \begingroup
3341       \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3342       \toks\tw\expandafter{%
3343         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3344         #8%
3345       }%
3346       \edef\x{\endgroup
3347         \the\toks@{\the\toks\tw@}%
3348       }\x
3349     }{%
3350       \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3351     }%
3352 }
3353 \expandafter\def\csname Parent-4\endcsname{}
3354 \expandafter\def\csname Parent-3\endcsname{}
3355 \expandafter\def\csname Parent-2\endcsname{}
3356 \expandafter\def\csname Parent-1\endcsname{}
3357 \expandafter\def\csname Parent0\endcsname{}
3358 \expandafter\def\csname Parent1\endcsname{}
3359 \expandafter\def\csname Parent2\endcsname{}
3360 \expandafter\def\csname Parent3\endcsname{}
3361 \expandafter\def\csname Parent4\endcsname{}
3362 %%
3363 %% End of file 'hgeneric-testphase.def'.
3364 \</package>
3365 <*colorscheme>
3366 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3367 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3368 % linkcolor=

```

```

3369 %,filecolor=
3370 %,urlcolor=
3371 %,menucolor=
3372 %,runcolor=
3373 %,linkbordercolor=
3374 %,filebordercolor=
3375 %,urlbordercolor=
3376 %,menubordercolor=
3377 %,runbordercolor=
3378
3379 \prop_const_from_keyval:Nn \c__hyp_colorscheme_original_prop
3380 {
3381     linkcolor      = [rgb]{1,0,0}, %red
3382     filecolor      = [rgb]{0,1,1}, %cyan
3383     urlcolor       = [rgb]{1,0,1}, %magenta
3384     menucolor      = [rgb]{1, 0, 0}, %red
3385     runcolor       = [rgb]{0,1,1}, %cyan
3386     %-----
3387     linkbordercolor = [rgb]{1, 0 ,0 },
3388     filebordercolor = [rgb]{0, .5, .5},
3389     urlbordercolor  = [rgb]{0, 1, 1},
3390     menubordercolor = [rgb]{1, 0, 0},
3391     runbordercolor  = [rgb]{0, .7, .7}
3392 }
3393
3394 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3395 {
3396     linkcolor      = [rgb]{0,0.2,0.6},
3397     filecolor      = [rgb]{0.8,0,0.8},
3398     urlcolor       = [rgb]{0.8,0,0.8},
3399     menucolor      = [rgb]{0,0.2,0.6},
3400     runcolor       = [rgb]{0.8,0,0.8},
3401     %----- %-----
3402     linkbordercolor = [rgb]{0,0.2,0.6},
3403     filebordercolor = [rgb]{0.8,0,0.8},
3404     urlbordercolor  = [rgb]{0.8,0,0.8},
3405     menubordercolor = [rgb]{0,0.2,0.6},
3406     runbordercolor  = [rgb]{0.8,0,0.8}
3407 }
3408
3409 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3410 { %two colors: intern/extern
3411     linkcolor      = [rgb]{0.79216, 0, 0.12549},
3412     filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3413     urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3414     menucolor      = [rgb]{0.79216, 0, 0.12549 },
3415     runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3416     %----- %-----
3417     linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3418     filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3419     urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3420     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3421     runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3422 }

```

```

3423
3424 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3425 { %all darkgray
3426   linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3427   filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3428   urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3429   menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3430   runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3431   %----- %-----
3432   linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3433   filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3434   urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3435   menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3436   runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3437 }
3438
3439 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3440 { %dvipsnam.def
3441   linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3442   filecolor      = [rgb]{1, 0, 0}, %Red
3443   urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3444   menucolor      = [rgb]{1, 0, 0}, %Red
3445   runcolor       = [rgb]{1, 0, 0}, %Red
3446   %----- %-----
3447   linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3448   filebordercolor = [rgb]{1, 0, 0}, %Red
3449   urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3450   menubordercolor = [rgb]{1, 0, 0}, %Red
3451   runbordercolor  = [rgb]{1, 0, 0} %Red
3452 }
3453
3454 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3455 { %dvipsnam.def
3456   linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3457   filecolor      = [rgb]{0, 1, 0}, %Green
3458   urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3459   menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3460   runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3461   %----- %-----
3462   linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3463   filebordercolor = [rgb]{0, 1, 0}, %Green
3464   urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3465   menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3466   runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3467 }
3468
3469
3470 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3471 {
3472   linkcolor      = [rgb]{0.50196, 0, 0.02353},
3473   filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3474   urlcolor       = [rgb]{0.54118, 0, 0.52941},
3475   menucolor      = [rgb]{0.44706, 0.45882, 0},
3476   runcolor       = [rgb]{0.07451, 0.46667, 0.46275},

```

```

3477 %----- %-----
3478 linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3479 filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3480 urlbordercolor = [rgb]{0.724708, 0.4, 0.717646},
3481 menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3482 runbordercolor = [rgb]{0.444706, 0.680002, 0.67765}
3483 }
3484
3485 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3486 {
3487 linkcolor = [rgb]{0,0,0},
3488 filecolor = [rgb]{0,0,0},
3489 urlcolor = [rgb]{0,0,0},
3490 menucolor = [rgb]{0,0,0},
3491 runcolor = [rgb]{0,0,0},
3492 %----- %-----
3493 linkbordercolor = [rgb]{0,0,0},
3494 filebordercolor = [rgb]{0,0,0},
3495 urlbordercolor = [rgb]{0,0,0},
3496 menubordercolor = [rgb]{0,0,0},
3497 runbordercolor = [rgb]{0,0,0}
3498 }
3499 </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>	491
<code>\%</code>	492
<code>\.</code>	306, 310, 312
@currupt commands:	
<code>\@currupt:</code>	2461, 2529
<code>\\</code>	19, 20, 21, 28, 32, 42, 62, 69, 76, 83, 90, 105, 110, 111, 119, 120, 129, 130, 131, 132, 133, 139, 617, 2463, 2531, 2818
<code>_</code>	304, 306, 310, 312

A

<code>\A</code>	304
<code>\Acrobatmenu</code>	15, 156
<code>\addcontentsline</code>	10
<code>\AddToDocumentProperties</code>	180
<code>\AddToHook</code>	205
<code>\advance</code>	2394, 2468, 2512, 2513
<code>allcolor</code> (hypersetup key)	<u>750</u>
<code>\author</code>	2

B

<code>\b</code>	308
<code>\begingroup</code>	... 2237, 2428, 2457, 2469, 2503, 2840, 3269, 3281, 3310, 3340	
<code>\belowpdfbookmark</code>	3233
<code>\BOOKMARK</code>	3236
<code>bookmarkstype</code> (hypersetup key)	10
bool commands:		
<code>\bool_if:NTF</code> 437, 451, 462, 475, 518, 568, 628, 720, 730, 970, 981	
<code>\bool_if:nTF</code>	580, 1673, 1698, 2977, 3085	
<code>\bool_lazy_and:nnTF</code>	200
<code>\bool_new:N</code>	290, 294, 298
<code>\bool_set_true:N</code>	299
<code>bordercolormodel</code> (hypersetup key)	10, <u>771</u>	
box commands:		
<code>\box_dp:N</code>	380, 2600, 2639, 2674
<code>\box_ht:N</code>	379, 2599, 2638, 2673
<code>\box_new:N</code>	239, 301
<code>\box_set_dp:Nn</code>	1870
<code>\box_set_ht:Nn</code>	1877
<code>\box_use:N</code>	987, 2602, 2648, 2677
<code>\box_use_drop:N</code>	992
<code>\box_wd:N</code>	378, 2598, 2637, 2672
<code>\l_tmpa_box</code>	2596, 2598, 2599, 2600, 2602, 2635, 2637, 2638, 2639, 2648, 2670, 2672, 2673, 2674, 2677

C

<code>\char</code>	2333, 2345, 2357, 2369
<code>\chardef</code>	137
<code>\cite</code>	25
clist commands:	
<code>\clist_map_function:nN</code>	112, 121
<code>\clist_map_inline:nn</code>	2034
<code>\clist_map_variable:nNn</code>	2460, 2528
color commands:	
<code>\color_export:nnN</code>	670
<code>\color_export:nnnN</code>	676
<code>\color_select:n</code>	687, 723, 991
<code>\color_select:nn</code>	693
<code>\color_set:nn</code>	185, 704
<code>\color_set:nnn</code>	9, 184, 710
color names:	
<code>hyp/annot/file</code>	287
<code>hyp/annot/link</code>	287
<code>hyp/annot/menu</code>	287
<code>hyp/annot/run</code>	287
<code>hyp/annot/url</code>	287
<code>colorfile</code> (hypersetup key)	750
<code>colorlink</code> (hypersetup key)	750
<code>colorlinks</code> (hypersetup key)	736
<code>colormenu</code> (hypersetup key)	750
<code>colorrun</code> (hypersetup key)	750
<code>colorscheme</code> (hypersetup key)	1, 1154
<code>colorurl</code> (hypersetup key)	750
cs commands:	
<code>\cs_generate_variant:Nn</code>	136, 182, 218, 233, 238, 342, 679, 696, 713
<code>\cs_gset:Npn</code>	962
<code>\cs_gset_eq:NN</code>	9
<code>\cs_if_exist_p:N</code>	201
<code>\cs_new:Npn</code>	105, 234
<code>\cs_new_protected:Npn</code>	177, 212, 221, 318, 322, 326, 330, 343, 345, 349, 391, 397, 404, 408, 430, 435, 449, 460, 473, 516, 566, 626, 663, 674, 680, 691, 697, 708, 906, 1905, 1961
<code>\cs_set_eq:NN</code>	140, 149, 150, 491, 492, 1231
<code>\cs_set_protected:Npn</code>	138, 163, 359, 617
<code>\csname</code>	2475, 2810, 2821, 3301, 3311, 3353, 3354, 3355, 3356, 3357, 3358, 3359, 3360, 3361
<code>\currentpdfbookmark</code>	3229

D	
<code>\d</code>	306, 310, 312
<code>debug</code> (hypersetup key)	<u>1178</u>
<code>\DeclareDocumentMetadata</code>	
.	3, 8, <u>11</u> , <u>12</u> , <u>15</u> , 21, 29, 35
<code>\DeclareRobustCommand</code>	3255
<code>\def</code>	144,
.	354, 2200, 2204, 2211, 2229, 2258,
.	2275, 2392, 2401, 2416, 2417, 2420,
.	2421, 2434, 2435, 2451, 2452, 2455,
.	2456, 2494, 2524, 2562, 2576, 2577,
.	2589, 2617, 2619, 2620, 2630, 2653,
.	2655, 2656, 2667, 2686, 2688, 2689,
.	2691, 2692, 2701, 2721, 2788, 2798,
.	2800, 2805, 2817, 2828, 2839, 2850,
.	2893, 2902, 2914, 2967, 3001, 3036,
.	3075, 3125, 3160, 3197, 3204, 3211,
.	3218, 3250, 3266, 3280, 3292, 3293,
.	3316, 3323, 3331, 3353, 3354, 3355,
.	3356, 3357, 3358, 3359, 3360, 3361
<code>\DefaultHeightofCheckBox</code>	2692
<code>\DefaultHeightofChoiceMenu</code>	2456
<code>\DefaultHeightofReset</code>	2656
<code>\DefaultHeightofSubmit</code>	2620
<code>\DefaultHeightofText</code>	2425
<code>\DefaultHeightofTextMultiline</code>	2423
<code>\DefaultOptionsofCheckBox</code>	2695
<code>\DefaultOptionsofPushButton</code>	2581
<code>\DefaultOptionsofReset</code>	2660
<code>\DefaultOptionsofSubmit</code>	2624
<code>\DefaultOptionsofText</code>	2430
<code>\DefaultWidthofCheckBox</code>	2691
<code>\DefaultWidthofChoiceMenu</code>	2455
<code>\DefaultWidthofReset</code>	2655
<code>\DefaultWidthofSubmit</code>	2619
<code>\DefaultWidthofText</code>	2420
<code>destlabel</code> (hypersetup key)	<u>10</u>
dim commands:	
<code>\dim_to_decimal_in_bp:n</code>	9, 140
<code>\do</code>	2461, 2529
<code>\documentclass</code>	22
<code>draft</code> (hypersetup key)	<u>1178</u>
E	
<code>\edef</code> 2239, 2471, 2507, 2514, 2803, 3275, 3346	
<code>\EdefEscapeName</code>	2527, 2532
<code>\else</code>	2236, 2424, 2478,
.	2482, 2485, 2502, 2510, 2791, 2796,
.	2813, 2819, 2831, 2835, 2852, 2856,
.	2860, 2864, 2868, 2872, 2876, 2880,
.	2884, 2888, 2905, 2909, 3010, 3011,
.	3012, 3014, 3017, 3021, 3025, 3031,
.	3048, 3052, 3056, 3062, 3066, 3134,
.	3135, 3137, 3140, 3144, 3169, 3170,
.	3172, 3175, 3179, 3246, 3313, 3334
<code>\endcsname</code>	2489, 2787, 2810,
.	2821, 3301, 3311, 3353, 3354, 3355,
.	3356, 3357, 3358, 3359, 3360, 3361
<code>\endgroup</code>	2240, 2448, 2471, 2517,
.	2520, 2847, 3275, 3287, 3310, 3346
<code>\endinput</code>	3247
exp commands:	
<code>\exp_args:Ne</code>	2895
<code>\exp_args:Nne</code>	1999
<code>\exp_args:Nno</code>	1947
<code>\exp_args:NV</code>	1658, 1689
<code>\exp_args:NnV</code>	393, 400
<code>\exp_args:Nnx</code>	2182
<code>\exp_args:No</code>	
.	422, 426, 2579, 2622, 2658, 2693
<code>\exp_args:NV</code>	376, 383
<code>\exp_args:Nx</code>	1940, 1988, 2231
<code>\exp_not:N</code>	617
<code>\exp_not:n</code>	
.	1093, 1127, 1238, 1298, 1319, 1356,
.	1376, 1399, 1443, 1476, 1501, 1552,
.	1577, 1604, 1653, 1730, 1758, 1785,
.	1805, 2020, 2127, 2155, 2165, 2176
<code>\expandafter</code>	2429, 2463,
.	2472, 2474, 2475, 2531, 2804, 2820,
.	3247, 3271, 3272, 3310, 3311, 3333,
.	3335, 3342, 3343, 3353, 3354, 3355,
.	3356, 3357, 3358, 3359, 3360, 3361
<code>\ExplSyntaxOff</code>	
.	11, 2268, 2384, 2411, 2717, 2785,
.	2901, 2965, 3000, 3072, 3123, 3222
<code>\ExplSyntaxOn</code>	
.	8, 12, 2269, 2389, 2412, 2720,
.	2892, 2913, 2966, 3035, 3073, 3159
<code>extension</code> (hypersetup key)	<u>11</u> , <u>1200</u>
F	
<code>\fbox</code>	2375, 2380, 2553
<code>\fi</code>	485, 2225,
.	2252, 2255, 2264, 2426, 2434, 2435,
.	2466, 2484, 2487, 2488, 2494, 2497,
.	2506, 2508, 2515, 2518, 2793, 2815,
.	2825, 2827, 2833, 2837, 2844, 2845,
.	2854, 2858, 2862, 2866, 2870, 2874,
.	2878, 2882, 2886, 2890, 2907, 2911,
.	3010, 3011, 3012, 3019, 3023, 3027,
.	3029, 3031, 3050, 3054, 3058, 3062,
.	3069, 3134, 3135, 3142, 3146, 3148,
.	3169, 3170, 3177, 3181, 3183, 3248,
.	3253, 3303, 3304, 3315, 3319, 3336
<code>file</code> (hypersetup key)	8, <u>1246</u>

file commands:		
\file_input:n	10	
fileborderstyle (hypersetup key)	11, 860	
filecolor (hypersetup key)	750	
final (hypersetup key)	1178	
\fontencoding	2329, 2341, 2353, 2365	
\fontfamily	2328, 2340, 2352, 2364	
\fontseries	2330, 2342, 2354, 2366	
\fontshape	2331, 2343, 2355, 2367	
fp commands:		
\fp_eval:n	1828, 1856, 1872	
G		
\GetDocumentProperties	2	
\global	2394, 2783	
group commands:		
\group_begin:		
. 332, 440, 454, 477, 520, 570, 630,		
722, 973, 990, 2106, 2327, 2339,		
2351, 2363, 2578, 2621, 2657, 2690		
\group_end:		
. 339, 444, 465, 507, 558, 611, 658,		
732, 993, 996, 2113, 2334, 2346,		
2358, 2370, 2614, 2650, 2683, 2715		
H		
\hbox	3257	
hbox commands:		
\hbox_overlap_right:n	987	
\hbox_set:Nn	2596, 2635, 2670	
\hbox_set:Nw	974	
\hbox_set_end:	983	
\hbox_set_to_wd:Nnn	1854	
hidefile (hypersetup key)	1130	
hidelink (hypersetup key)	1130	
hidelinks (hypersetup key)	1130	
hidemenu (hypersetup key)	1130	
hiderun (hypersetup key)	1130	
hideurl (hypersetup key)	1130	
hook commands:		
\hook_gput_code:nnn		
. 192, 716, 726, 966, 977		
\hook_new:n	183, 317	
\hook_use:n	334, 441, 455	
\hss	3262	
Hy internal commands:		
\Hy_hyescapeform	2804, 2817, 2823	
\Hy_hypSectionAnchor	3252, 3255	
HyField internal commands:		
\HyField_hypAddToFields	2231, 2258	
hyp commands:		
\l_hyp_annot_colorfile_bool	288	
\l_hyp_annot_colorlink_bool	288	
\l_hyp_annot_colormenu_bool	288	
\l_hyp_annot_colorrund_bool	288	
\l_hyp_annot_colorurl_bool	288	
\l_hyp_annot_ocgcolorfile_bool	292	
\l_hyp_annot_ocgcolorlink_bool	292	
\l_hyp_annot_ocgcolormenu_bool	292	
\l_hyp_annot_ocgcolorrund_bool	292	
\l_hyp_annot_ocgcolorurl_bool	292	
hyp internal commands:		
\g__hyp_AcroForm_CoFields_prop		
. 2192, 2213, 2287, 2289, 2304		
\g__hyp_AcroForm_Fields_prop		
. 2193, 2206, 2282		
\l__hyp_annot_GoTo_bool	437, 451, 462	
\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	296	
\l__hyp_annot_GoToR_bool	518	
\l__hyp_annot_Launch_bool	568	
\l__hyp_annot_Named_bool	628	
\c__hyp_annot_types_seq		
. 251, 288, 292, 750, 1045, 1141		
\l__hyp_annot_URI_bool	475	
\g__hyp_bordercolormodel_str		
. 287, 775, 791		
\l__hyp_CheckmarkOff_tl		
. 2272, 2273, 2929		
\l__hyp_CheckmarkYes_tl		
. 2270, 2271, 2928		
__hyp_clist_display:n	105, 112, 121	
__hyp_color_export:nnN		
. 31, 663, 663, 679, 791		
__hyp_color_select:n	32, 680, 680, 696	
__hyp_color_select:nN	32	
__hyp_color_set:nN	32	
__hyp_color_set:nn		
. 32, 188, 190, 697, 697, 713, 755		
__hyp_colormodel_export:nnnN	31, 663	
__hyp_colormodel_export:wnnN		
. 667, 674		
__hyp_colormodel_select_aux:wn		
. 680, 684, 691		
__hyp_colormodel_set_aux:nwn		
. 697, 701, 708		
\c__hyp_colorscheme_daleif_prop	3394	
\c__hyp_colorscheme_henryford_-		
prop	3485	
\c__hyp_colorscheme_julian_prop	3409	
\c__hyp_colorscheme_original_-		
prop	3379	
\c__hyp_colorscheme_phetype_prop		
. 3470		

\c_hyp_colorscheme_szabolcsA-prop	3439	\l_hyp_text_enc_dest_print_tl ..	273, 530
\c_hyp_colorscheme_szabolcsB-prop	3454	\l_hyp_text_enc_dest_tl	24, 273, 372, 423
\c_hyp_colorscheme_tivv_prop .	3424	\l_hyp_text_enc_file_print_tl ..	273, 525, 573
\l_hyp_dest_box	24, 301, 378, 379, 380, 1854, 1870, 1877	\l_hyp_text_enc_info_print_tl ..	273, 347
\l_hyp_dest_name_tmpa_tl ..	244, 411, 412, 418, 422, 424, 427, 531, 539	\l_hyp_text_enc_para_print_tl ..	273, 578
\l_hyp_dest_pdfremotestartview_tl	270, 536, 1661, 1665	\l_hyp_text_enc_uri_print_tl ..	273, 480, 1257
\g_hyp_dest_pdfstartpage_tl ...	270, 1672, 1674, 1681, 1699, 1706	__hyp_text_pdfstring:nnN	330, 330, 342, 344, 347, 370, 421, 478, 523, 528, 571, 576, 1257
\g_hyp_dest_pdfstartview_tl ...	270, 1674, 1681, 1692, 1696, 1699, 1706	__hyp_text_pdfstring_info:nN ..	345, 345, 1916, 1934, 1944
\l_hyp_dest_pdfview_tl	286, 394, 401, 1823, 1826, 1833, 1836, 1837, 1838, 1839, 1840, 1841, 1846, 1850, 1886	__hyp_text_purify:nN .	318, 318, 335
\c_hyp_dest_startview_regex ...	302, 1659, 1690	__hyp_text_string_from_unicode:nN	326, 326, 337
\c_hyp_dest_undefined_tl	250, 417, 418	\g_hyp_text_tmpa_str .	249, 338, 340
__hyp_destination:nn	24, 365, 365, 394, 401	\l_hyp_text_tmpa_str	248, 335, 336, 337, 338
\l_hyp_filename_tmpa_tl ...	244, 521, 522, 524, 526, 527, 574, 575, 591	\l_hyp_tmpa_box ...	239, 974, 987, 992
__hyp_link_goto_begin:nw	408, 442, 456	\l_hyp_tmpa_int	239, 1659, 1661, 1690, 1692, 1813, 1814, 1818, 1820, 1844, 1851, 1852, 1853, 1862, 1864, 1874, 1879, 2291, 2293, 2299
__hyp_link_goto_end: .	430, 443, 464	\l_hyp_tmpa_str	239, 1916, 1917, 1919, 1934, 1935, 1937, 1944, 1945, 1949
\c_hyp_map_annot_hyp_prop	251	\l_hyp_tmpa_tl ..	239, 373, 377, 385, 791, 795, 1257, 1258, 1263, 1657, 1659, 1688, 1690, 1820, 1821, 1828
\c_hyp_map_hyp_annot_prop	251, 714, 778, 814, 841, 849, 860, 887, 895, 964, 1068, 1104, 1114, 1246	\l_hyp_uri_tmpa_tl ...	244, 481, 482
__hyp_ocg_init: ...	906, 906, 962, 972	hyp/annot/file (color name)	287
__hyp_PageLabels_gpush: ..	349, 349, 362	hyp/annot/link (color name)	287
\l_hyp_para_tmpa_tl ..	247, 579, 582, 591	hyp/annot/menu (color name)	287
\l_hyp_para_tmpa_tl__hyp_text_tmpa_str	244	hyp/annot/run (color name)	287
\l_hyp_RadioYes_tl	2522, 2523	hyp/annot/url (color name)	287
__hyp_ref_check:nn ..	211, 221, 233, 2399	hyp/text/pdfstring	317
__hyp_ref_if_exist:nn .	211, 219, 220	\hypercalcbp	9, 15, 140
__hyp_ref_if_exist:nnTF ..	223, 2403	\HyperDestNameFilter	371, 422
__hyp_ref_label:nn ..	211, 212, 218, 2398	\hypersetup ..	1, 2, 6, 8, 10, 16, 59, 91, 163
__hyp_ref_value:nn ..	211, 234, 238, 2407	\hypersetup keys:	
__hyp_setup_info_date_key:nn ...	1961, 1993, 1994	allcolor	750
__hyp_setup_info_key:nn ..	1905, 1954, 1955, 1956, 1957, 1959, 1960	bookmarkstype	10
__hyp_store_metadata:nn	177, 182, 1902, 1922, 1940, 1974, 1988, 2009, 2081	bordercolormodel	10, 771
__hyp_text_cleanup:N .	322, 322, 336	colorfile	750
		colorlink	750
		colorlinks	736
		colormenu	750
		colorrun	750

\keys_set:nn	166, 194, 1160, 1167, 2000, 2027, 2030, 2031, 2032, 2033, 2107, 2182
\keys_set_known:nn	2185
\kvsetkeys	165, 2278
L	
\label	6, 11
\LayoutCheckField	2703
\LayoutChoiceField	2498
\LayoutPushButtonField	2591, 2608
\LayoutTextField	2436
\leavevmode	2437, 2535, 3256
legacy commands:	
\legacy_if:nTF	151, 203, 207, 357, 2316, 2587, 2628, 2666, 2699, 2956
\let	147, 2195, 2196, 2198, 2199, 2226, 2277, 2385, 2386, 2387, 2418, 2419, 2453, 2454, 2496, 2783, 2790, 2792, 2797, 2814, 2848, 3265, 3279, 3282, 3291, 3302, 3309, 3312, 3314, 3322, 3330
link (hypersetup key)	8, <u>1246</u>
linkborder (hypersetup key)	11
linkborderstyle (hypersetup key)	11, <u>860</u>
linkcolor (hypersetup key)	<u>750</u>
linkfileprefix (hypersetup key)	<u>1200</u>
linktoc (hypersetup key)	<u>1200</u>
linktocpage (hypersetup key)	<u>1200</u>
localanchorname (hypersetup key)	<u>1200</u>
M	
\MakeButtonField	2596, 2611, 2635, 2670, 2681
\MakeChoiceField	2571
\MakeFieldObject	2186, 2325, 2337, 2349, 2361, 2373, 2378
\MakeRadioField	2553
\MakeTextField	2445
\mbox	984
menu (hypersetup key)	8, <u>1246</u>
menuborder (hypersetup key)	11
menuborderstyle (hypersetup key)	11, <u>860</u>
menucolor (hypersetup key)	<u>750</u>
mode commands:	
\mode_if_horizontal:TF	367, 389
\mode_leave_vertical:	410, 494, 546, 598, 633, 2565, 2593, 2610, 2632, 2662, 2705
msg commands:	
\msg_error:nn	202, 2607, 2680
\msg_line_context:	70
\g_msg_module_name_prop	13
\msg_new:nnn	38, 45, 50, 54, 58, 65, 72, 79, 86, 93, 99, 106, 115, 125
\msg_new:nnnn	14, 24
\msg_warning:nn	153, 1174
\msg_warning:nnn	162, 414, 655
\msg_warning:nnnn	1006, 1053, 1296, 1340, 1374, 1397, 1459, 1488, 1522, 1537, 1550, 1563, 1588, 1621, 1637, 1664, 1695, 1728, 1742, 1769, 1803, 1885
\msg_warning:nnnnn	173, 1090, 1124, 1235, 1316, 1353, 1440, 1473, 1498, 1574, 1601, 1650, 1755, 1782, 2017, 2124, 2143, 2162, 2173
N	
naturalnames (hypersetup key)	<u>1200</u>
nesting (hypersetup key)	11
\newcommand	143, 3230, 3232, 3234
\newcount	2390, 2561
\NewDocumentCommand	2186
\newlength	3249
\noexpand	2472, 2473, 2474, 2475, 2801
O	
ocgcolorfile (hypersetup key)	<u>1000</u>
ocgcolorlink (hypersetup key)	<u>1000</u>
ocgcolorlinks (hypersetup key)	<u>1000</u>
ocgcolormenu (hypersetup key)	<u>1000</u>
ocgcolorrun (hypersetup key)	<u>1000</u>
ocgcolorurl (hypersetup key)	<u>1000</u>
P	
page/Trans internal commands:	
l_hyp_page/Trans	315
pageanchor (hypersetup key)	<u>1200</u>
\paperwidth	3
\PassOptionsToPackage	209, 1192, 1197
pdf commands:	
\pdf_bdcobject:nn	986, 989
\pdf_destination:nn	20, 136, 384
\pdf_destination:nnnn	377
\pdf_emc:	988, 994
\pdf_link_user:nnn	2568
\pdf_name_from_unicode_e:n	635
\pdf_object_new:nn	908, 909, 910, 911, 2723, 2724, 2725
\pdf_object_ref:n	914, 916, 938, 939, 942, 946, 951, 956, 961, 2309, 2311, 2781
\pdf_object_ref_last:	2112
\pdf_object_unnamed_write:nn	2108
\pdf_object_write:nn	912, 918, 928, 940, 2726, 2768, 2775

\pdf_pageobject_ref:n	1681, 1706, 2405	\pdfmanagement_add:nnn
\pdf_string_from_unicode:nnN	.. 328	351, 938, 939,
\pdf_version:	961, 1263, 1288, 1304, 1308, 1328,
... 1344, 1463, 1492, 1526, 1541,		1336, 1365, 1420, 1430, 1454, 1468,
1567, 1592, 1625, 1641, 1746, 1773		1480, 1485, 1510, 1518, 1533, 1559,
\pdf_version_compare:NnTF	1584, 1616, 1633, 1679, 1704, 1718,
.....	1000, 1047,	1738, 1765, 1794, 1900, 1919, 1937,
.....	1334, 1425, 1447, 1483, 1516, 1531,	1948, 1972, 1986, 2004, 2093, 2112,
.....	1557, 1582, 1608, 1631, 1736, 1763	2284, 2301, 2308, 2310, 2312, 2318
\pdf_version_compare_p:Nn 584	\pdfmanagement_if_active_p:	... 201
\pdf_version_major: .. 150, 1009, 1057		\pdfmanagement_remove:nn
\pdf_version_minor: .. 149, 1009, 1057		... 1260, 1284, 1292, 1312, 1324,	
pdfannot commands:		1349, 1361, 1369, 1383, 1392, 1436,	
\pdfannot_box:nnnn	1451, 1470, 1507, 1546, 1571, 1597,	
.. 2440, 2538, 2597, 2636, 2671, 2708		1612, 1646, 1676, 1701, 1714, 1723,	
\pdfannot_box_ref_last: .. 2233, 2249		1751, 1778, 1790, 1799, 1897, 1913,	
\pdfannot_dict_put:nnn	1931, 1969, 1983, 2090, 2103, 2322	
.. 792, 827, 851, 873, 897, 1075, 1106		\pdfmanagement_show:n 2285
\pdfannot_dict_remove:nn	pdfmeta commands:	
.. 786, 822, 843, 868, 889, 1083, 1116		\pdfmeta_standard_verify:nnTF	...
\pdfannot_link:nnn	.. 495, 547, 599, 641	631, 2584, 2663
\pdfannot_link_goto_begin:nw	.. 427	\pdfmeta_standard_verify:nTF	...
\pdfannot_link_goto_end:	432	2314, 2897
\pdfannot_link_margin:n 6, 1378	pdfmoddate (hypersetup key) 11, 1961
\c_pdfannot_link_types_seq 296	pdfproducer (hypersetup key) 1905
pdfauthor (hypersetup key) 1905	pdfremotestartview (hypersetup key)	... 9
\pdfbookmark 3235	pdfstartview (hypersetup key) 9
pdfborder (hypersetup key) 11	\pdfstringdef 6, 7, 15, 21, 2846
pdfborderstyle (hypersetup key)	.. 11, 860	pdfsubject (hypersetup key) 1905
pdfcreationdate (hypersetup key)	11, 1961	pdftitle (hypersetup key) 1905
pdfcreator (hypersetup key) 1905	pdftrapped (hypersetup key) 1995
\pdfdest 9	pdfversion (hypersetup key) 1168
pdfdict commands:		pdfview (hypersetup key) 9, 1809
\pdfdict_new:n	315, 469, 512, 562, 622	pdfxform commands:	
\pdfdict_put:nnn	\pdfxform_new:nnn 2188
.. 316, 470, 471, 482, 484, 489, 513,		\pdfxform_ref:n
514, 527, 535, 539, 544, 563, 564,		.. 2548, 2549, 2643, 2644, 2928, 2929	
575, 590, 596, 623, 624, 634, 639,		\phantom 2345, 2369
1404, 1405, 1409, 1410, 2121, 2131,		\phantomsection 3318
2135, 2138, 2140, 2159, 2168, 2170		plainpages (hypersetup key) 1200
\pdfdict_remove:nn	... 587, 1414, 1415	prg commands:	
\pdfdict_use:n	499, 551, 604, 645, 2110	\prg_generate_conditional_-	
pdfencoding (hypersetup key) 1168	variant:Nnn 220
\pdfescapestring 2790	\prg_new_eq_conditional:NNn	... 219
PDFForm internal commands:		prop commands:	
\PDFForm__hypName	\prop_const_from_keyval:Nn
..... 2829, 2832, 2836, 2839		253, 261, 3379, 3394,
pdfinfo (hypersetup key) 2023	3409, 3424, 3439, 3454, 3470, 3485
pdfkeywords (hypersetup key) 1905	\prop_gput:Nnn 13, 2206, 2213
pdflang (hypersetup key) 11, 1891	\prop_if_empty:Nn 2287
pdflinkmargin (hypersetup key) 11	\prop_item:Nn 2304
pdfmanagement commands:		\prop_map_inline:Nn	714, 778, 814,
\pdfmanagement_add:nn 1387	841, 849, 860, 887, 895, 964, 1068,
		1104, 1114, 1158, 1246, 2282, 2289

`\prop_new:N` 2192, 2193
`\protect` 225, 3251
`\providecommand`
 141, 142, 2220, 2223, 2450, 2786
`\ProvidesFile` 3

R

`\ReadBookmarks` 3240
ref commands:
 `\ref_if_exist:nn` 219
 `\ref_label:nn` 215
 `\ref_value:nn` 18, 236
regex commands:
 `\regex_const:Nn` 302
 `\regex_extract_once:NnNTF` 1659, 1690
`\relax` 2453, 2454,
 2495, 2783, 2830, 2834, 2904, 2908,
 3011, 3012, 3020, 3024, 3051, 3055,
 3135, 3143, 3170, 3178, 3302, 3311
`\RemoveFromHook` 1268
`\RenewDocumentCommand` 156
`\RequirePackage` 7, 20, 205, 3238
`run` (hypersetup key) 8, 1246
`runborder` (hypersetup key) 11
`runborderstyle` (hypersetup key) .. 11, 860
`runcolor` (hypersetup key) 750

S

`\selectfont` 2332, 2344, 2356, 2368
seq commands:
 `\seq_const_from_clist:Nn` 251
 `\seq_count:N` 1818, 1844
 `\seq_get_right:NN` 1820
 `\seq_item:Nn` 1661,
 1692, 1814, 1862, 1864, 1874, 1879
 `\seq_map_inline:Nn`
 . 288, 292, 296, 750, 1045, 1141, 2299
 `\seq_new:N` 241
 `\seq_put_right:Nn`
 1851, 1852, 1853, 2291
 `\seq_set_split:Nnn` 1813
 `\seq_sort:Nn` 2293
`\setlength` 3268, 3339
`\setpdflinkmargin` 6
`\settowidth` 2465
`\show` 2526
sort commands:
 `\sort_return_same:` 2297
 `\sort_return_swapped:` 2296
`\space` 229, 2534, 3013, 3030,
 3031, 3061, 3062, 3136, 3168, 3171
`\spacefactor` 367, 389
`\stockwidth` 3

str commands:

`\c_colon_str` 617
`\c_hash_str` 491
`\c_percent_str` 492
`\str_case:nnTF` 3193
`\str_case_e:nnTF` 1814
`\str_gset:Nn` 775
`\str_gset_eq:NN` 338
`\str_head:n` 2006, 2011
`\str_if_eq:nnTF`
 374, 1917, 1935, 1945, 2895
`\str_if_eq_p:nn` 582
`\str_lowercase:n`
 1814, 1940, 1988, 2007, 2012
`\str_new:N` 243, 248, 249, 287
`\str_remove_once:Nn` 488, 543, 595, 638
`\str_set:Nn` 320
`\str_set_convert:Nnnn` 6
`\str_set_eq:NN` 340
`\str_tail:n` 2007, 2012
`\str_uppercase:n` 2000, 2006, 2011
`\string` 2220,
 2223, 2244, 2262, 2806, 2810, 2821
`\subpdfbookmark` 3231
sys commands:
 `\c_sys_backend_str` 3193

T

TeX and LaTeX 2_ε commands:

`\@BOOKMARK` 3237
`\@CheckBox` 2686
`\@ChoiceMenu` 2451
`\@Form` 2275
`\@PushButton` 2576
`\@Reset` 2653
`\@Submit` 2617
`\@TextField` 2416
`\@bookmarksopenlevel` 1279
`\@bsphack` 214
`\@chapter` 3291, 3292
`\@currDisplay` 2465, 2534
`\@currValue` 2532, 2548
`\@currentHref`
 3272, 3285, 3296, 3326, 3343
`\@curropt` . 2450, 2460, 2463, 2528, 2531
`\@empty` 2419, 2435, 2851,
 2855, 2859, 2863, 2867, 2871, 2875,
 2879, 2883, 2887, 3031, 3062, 3063
`\@endForm` 2385
`\@esphack` 216
`\@expl@color@set@nnn` 9
`\@firstofone` .. 2226, 2792, 2797, 2814
`\@firstoftwo` 3333
`\@for` 2461, 2529

\@gobble	2277	\Fld@default	2418, 2435, 2453, 2496, 2526, 2527, 2689, 3063, 3067, 3068, 3115
\@gobbletwo	3282	\Fld@flags	2921, 2974, 3007, 3043, 3082, 3131, 3166
\@hyper@launch	30, 617	\Fld@format@code	2855, 2857
\@ifundefined	2277, 2806	\Fld@height	2421, 2442, 2445, 2456, 2514, 2540, 2553, 2571, 2620, 2656, 2692, 2710
\@latex@warning	226	\Fld@keystroke@code	2851, 2853
\@mainaux	2219, 2222, 2242, 2261	\Fld@listcount	2525, 2533, 2554, 2561
\@mkboth	3282	\Fld@mappingname	2834, 2836, 2908, 2910
\@nil	2804, 2817, 2823	\Fld@maxlen	3118, 3120
\@part	3309, 3316	\Fld@menulength	2458, 2464, 2511
\@pdfauthor	16	\Fld@name	2417, 2452, 2577, 2688, 2829, 2903
\@pdfborder	141	\Fld@onblur@code	2871, 2873
\@pdfborderstyle	142	\Fld@onclick@code	2996
\@savsf	367, 389	\Fld@onenter@code	2883, 2885
\@schapter	3279, 3280	\Fld@onexit@code	2887, 2889
\@secondoftwo	3335	\Fld@onfocus@code	2867, 2869
\@sect	3330, 3331	\Fld@onmousedown@code	2875, 2877
\@spart	3322, 3323	\Fld@onmouseup@code	2879, 2881
\@ssect	3265, 3266	\Fld@pageobjref	2387, 2401, 2786, 2918, 2971, 3004, 3040, 3079, 3128, 3163
\@tempdima	2459, 2466, 2468, 2505, 2506, 2507, 2511, 2512, 2513, 2514	\Fld@radiosymbol	3059
\@tempdimb	2465, 2466	\Fld@rotation	2933, 2935, 2979, 2986, 2988, 3010, 3016, 3018, 3047, 3049, 3087, 3096, 3098, 3134, 3139, 3141, 3169, 3174, 3176
\@typeset@protect	3251	\Fld@submitflags	3155
\c@secnumdepth	3299, 3317, 3332	\Fld@validate@code	2859, 2861
\calc@bm@number	3243	\Fld@value	2419, 2435, 2454, 2495, 2496, 3116
\check@bm@number	3242	\Fld@width	2420, 2434, 2441, 2445, 2455, 2494, 2505, 2507, 2539, 2553, 2571, 2589, 2619, 2630, 2655, 2667, 2691, 2701, 2709
\Fld@additionalactions	2893, 2963, 2997, 3033, 3070, 3117, 3157, 3186	\Fld@X@additionalactions	2850, 2895, 2898
\Fld@align	2922, 3008, 3083	\Form@action	3153
\Fld@altname	2830, 2832, 2904, 2906	\G@refundefinedtrue	225
\Fld@annotflags	2917, 2970, 3003, 3039, 3078, 3127, 3162	\H@old@part	3309, 3320
\Fld@annotnames	2902, 2919, 2972, 3005, 3041, 3080, 3129, 3164	\H@old@schapter	3279, 3288
\Fld@bcolor	2941, 2943, 3024, 3026, 3055, 3057, 3091, 3104, 3106	\H@old@sect	3330, 3341, 3350
\fld@bcolor	3012	\H@old@spart	3322, 3328
\Fld@bordercolor	2937, 2939, 2981, 2990, 2992, 3011, 3020, 3022, 3051, 3053, 3089, 3100, 3102, 3135, 3143, 3145, 3170, 3178, 3180	\H@old@ssect	3265, 3270
\Fld@borderstyle	2923, 2976, 3009, 3045, 3084, 3133, 3184	\Hy@abspage	361
\Fld@borderwidth	2512, 2513, 2923, 2976, 3009, 3045, 3084, 3133, 3184	\Hy@activeanchorfalse	406
\Fld@calculate@code	2235, 2863, 2865	\Hy@activeanchortrue	399
\Fld@calculate@sortkey	2246	\Hy@AtBeginDocument	2200, 2216
\Fld@cbsymbol	2945	\Hy@bookmarksnumberedfalse	1272
\Fld@charsize	2511, 2949, 3030, 3061, 3112, 3168	\Hy@bookmarksnumberedtrue	1273
\Fld@chekequals	2463, 2531		
\Fld@choices	3032		
\Fld@color	2950, 2952, 3031, 3062, 3113		

\Hy@bookmarksopenfalse	1276	\Hy@secnum@part	3312, 3314, 3317
\Hy@bookmarksopentru	1277	\Hy@SectionAnchorHref	3250, 3272, 3343
\Hy@bookmarkstype	1280	\Hy@SectionHShift	3249, 3258, 3268, 3339
\Hy@chapapp	3283, 3294	\Hy@StepCount	2464, 2533
\Hy@CleanupFile	522	\Hy@temp	2239, 2254, 2803, 2804
\Hy@colorlink	26	\Hy@unicodetrue	2843
\Hy@currentbookmarklevel	3224	\Hy@unicodetrue	146
\Hy@DisableOption	148, 155	\Hy@VerboseAnchor	369
\Hy@drafttrue	1191	\Hy@verbosefalse	1182
\Hy@escapeform	2439, 2537, 2567, 2595, 2634, 2669, 2707, 2788, 2795, 2798	\Hy@VerboseLinkStart	439, 453
\Hy@escapestring	2790, 2792, 2797, 2800, 2801, 2804, 2805, 2814, 2817, 2853, 2857, 2861, 2865, 2869, 2873, 2877, 2881, 2885, 2889, 2945, 2996, 3059, 3115, 3116, 3153	\Hy@VerboseLinkStop	445, 466, 505, 556, 609, 651
\Hy@finaltrue	1196	\Hy@verbosetrue	1181
\Hy@FormObjects	2281, 2721, 2783	\Hy@VersionChecked	137
\Hy@gtemp	2846, 2848	\Hy@WrapperDef	365
\Hy@href@nextactionraw	486, 488, 489, 541, 543, 544, 593, 595, 596, 636, 638, 639	\Hy@xspace@end	442, 504, 555, 608, 650
\Hy@href@page	536	\HyAnn@AbsPageLabel	2386, 2392, 2438, 2536, 2566, 2594, 2633, 2668, 2706
\Hy@hypertextnamesfalse	1206	\HyAnn@Count	2390, 2391, 2394, 2395, 2396, 2397, 2398, 2399, 2403, 2407
\Hy@hypertextnamestrue	1205	\HyField@AddToFields	2229, 2446, 2555, 2572, 2603, 2647, 2676, 2713
\Hy@linkfileprefix	1208	\HyField@afields	2195
\Hy@linktoc	1231	\HyField@AfterAuxOpen	2200, 2226, 2259
\Hy@localanchornametrue	1211	\HyField@AuxAddToCoFields	2199, 2211, 2223, 2244
\Hy@localanchornametrue	1210	\HyField@AuxAddToFields	2198, 2204, 2220, 2262
\Hy@MakeCurrentHrefAuto	3267, 3283, 3294, 3324, 3338	\HyField@cofields	2196
\Hy@MakeRemoteAction	532	\HyField@FlagsCheckBox	2698
\Hy@naturalnamesfalse	1215	\HyField@FlagsChoice	2504
\Hy@naturalnamestrue	1214	\HyField@FlagsPushButton	2586, 2626, 2665
\Hy@next	3293, 3302, 3305	\HyField@FlagsRadioButton	2500
\Hy@numberline	144, 3225	\HyField@FlagsSubmit	2627
\Hy@org@chapter	3291, 3306	\HyField@FlagsText	2433
\Hy@OutlineName	3241	\HyField@PDFChoices	2564
\Hy@OutlineRerunCheck	3239	\HyField@SetKeys	2429, 2470, 2473, 2492, 2579, 2622, 2658, 2693
\Hy@pageanchorfalse	1219	\HyPat@ObjRef	3197, 3204, 3211, 3218
\Hy@pageanchortrue	1218	\hyper@anchor	391
\Hy@pdfmajorversion	150	\hyper@anchorend	391, 3260, 3285, 3296, 3326
\Hy@pdfminorversion	149	\hyper@anchorstart	391, 3260, 3285, 3296, 3326
\Hy@pdfstringtrue	22, 333	\hyper@link	26, 26, 435
\Hy@pdfversion	2841	\hyper@linkend	26, 26, 460
\Hy@plainpagesfalse	1223	\hyper@linkfile	516
\Hy@plainpagestrue	1222	\hyper@linklaunch	29, 566, 620
\Hy@pstringdef	343, 2821	\hyper@linknamed	30, 158, 626
\Hy@PutCatalog	349	\hyper@linkstart	26, 26, 449
\Hy@raisedlink	3259, 3284, 3295, 3325	\hyper@linkurl	473
\Hy@RestoreLastskip	388		
\Hy@safe@activetrue	493, 2238		
\Hy@SaveLastskip	368		

\HyPL@Labels	351, 361	\textsf	2375, 2380
\HyPL@StorePageLabel	349	\the	361,
\HyPsd@LoadUnicode	145		2395, 2396, 2397, 2398, 2399, 2403,
\HyPsd@pdfencoding	147		2407, 2507, 2511, 2514, 3276, 3347
\HyPsd@pdfencoding@unicode	147	\thepage	229
\HyPsd@SanitizeForOutFile	3228	\ThisShouldNotHappen	2808
\if@filesw	2218, 2241, 2260	\title	2
\ifFld@combo	2479, 2509	tl commands:	
\ifFld@hidden	2434, 2494	\c_space_tl	144, 915,
\ifFld@multiline	2422		946, 951, 956, 2548, 2549, 2556,
\ifFld@popdown	2480		2732, 2734, 2744, 2746, 2748, 2751,
\ifFld@radio	2476, 2499		2753, 2949, 2952, 3084, 3112, 3113
\ifHy@href@ismap	483	\tl_const:Nn	250
\ifHy@implicit	3245	\tl_gput_right:Nn	361
\ifHy@pdfescapeform	2789, 2799	\tl_gset:Nn	1672, 1692, 1696
\ifHy@unicode	2842	\tl_if_blank:nTF	
\kv@set@family@handler	161		533, 1911, 1929, 1967, 1981, 2088, 2101
\ltx@empty	2195,	\tl_if_empty:nTF	412, 486, 541,
	2196, 2235, 2385, 2386, 2387, 2418		593, 636, 1258, 2937, 2941, 2950, 3113
\ltx@gobble	2198	\tl_if_empty:nTF	784, 820,
\ltx@gobbletwo	2199		839, 866, 885, 1449, 1505, 1610, 1895
\ltx@ifUndefined	3290, 3300	\tl_if_empty_p:N	1674, 1699
\ltx@one	2394	\tl_if_eq:NnTF	1821
\ltx@ReturnAfterFi	2822	\tl_if_exist:nTF	2990, 3100, 3104
\ltx@zero	2391	\tl_if_exist_p:N	2981, 3089, 3091
\m@ne	3299, 3314	\tl_if_head_eq_charcode:nNTF	
\OBJ@OCG@view	143		665, 682, 699
\p@	2468	\tl_new:N	240, 244, 245, 246,
\pdf@ifdraftmode	2279		247, 270, 271, 272, 273, 274, 275,
\PDF@SetupDoc	138		276, 277, 278, 286, 2270, 2272, 2522
\pdf@strcmp	2295	\tl_set:Nn	179, 280,
\PDFForm@Check	2707, 2712, 2914		281, 282, 283, 284, 285, 521, 1657,
\PDFForm@List	2567, 2570, 3001		1661, 1665, 1688, 1823, 1826, 1833,
\PDFForm@Name			1836, 1837, 1838, 1839, 1840, 1841,
	2432, 2493, 2583, 2697, 2828		1846, 1850, 1886, 2271, 2273, 2523
\PDFForm@Push	2595, 2601, 2967	\tl_set_eq:NN	418
\PDFForm@Radio	2537, 2543, 3036	\tl_to_str:n	
\PDFForm@Reset	2669, 2675, 3160		20, 21, 22, 33, 228, 1972, 1986
\PDFForm@Submit	2634, 2641, 3125	token commands:	
\PDFForm@Text	2439, 2444, 3075	\token_to_str:N	29
\protected@edef	411	\toks	3271, 3276, 3342, 3347
\strip@pt	2949, 3030, 3061, 3112, 3168		
\toks@	3270, 3276, 3341, 3347	U	
\tw@	3271, 3276, 3342, 3347	unicode (hypersetup key)	1168
\XR@ext	1202	url (hypersetup key)	8, 1246
\z@	2459, 3010, 3016,	\url	4
	3047, 3134, 3139, 3169, 3174, 3312	urlborder (hypersetup key)	11
\zref@labelbylist	2396	urlborderstyle (hypersetup key)	11, 860
\zref@labelbyprops	2395	urlcolor (hypersetup key)	750
\zref@refused	2397	\Urlfont	4
\texorpdfstring	22	use commands:	
text commands:		\use:n	615
\text_purify:n	6, 320	V	
\textcolor	2277, 2375, 2380	verbose (hypersetup key)	1178

	W	X
\write	2219, 2222, 2242, 2261	2471, 2491, 3275, 3277, 3346, 3348