

The `hyperref-generic` module

A generic driver for `hyperref`

The L^AT_EX Project*

Version 0.95b, released 2021-03-07

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

- This driver uses `l3color` for the colors. All colors defined with `\color_set:nn` or `\color_set:nnn` will work. Colors defined with `xcolor` will work if they don't use one of the special color models not supported by `l3color` as `pdfmanagement-firstaid` contains a patch for `xcolor`. If the package `color` is used it is currently recommended to define colors after `hyperref`. 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/graphicx`, `geometry`, the KOMA-classes, memoir, ... all try to set this.

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

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

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

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

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

The next compilation can then make use of it. The two-pass could be avoided in the future with a better labeling system, where the name 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 v0.95b %
5 generic Hyperref driver for the LaTeX PDF management testphase bundle]
6
7 \RequirePackage{etoolbox} %why?
8 \ExplSyntaxOn
9 \file_input:n {hyperref-colorschemes.def}
10 \ExplSyntaxOff

```

1 messages

Redirect the message name:

```

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

```

At first a message for the testing of the resource management

```

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

```

The pdfversion should be set in \DeclareDocumentMetadata

```

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

```

A generic message for ignored keys.

```

37 \msg_new:nnn

```

```

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

```

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

```

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

```

pdf/A message for not allowed Named actions

```

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

```

A message if the destination name is empty.

```

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

```

A message if the destination check fails

```

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

```

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

```

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

```

```

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

```

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:

```

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

```

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

```

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

```

```

133     No~value~is~equivalent~to~using~‘true‘.
134 }

```

2 Variants

```

135 \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

```

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

137 %\cs_set_protected:Npn \PDF@SetupDoc{}
138 %\PDF@FinishDoc{}% dummy needed for hyperref ...

```

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

```

139 \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:

```

140 \providecommand\@pdfborder{}
141 \providecommand\@pdfborderstyle{}
142 \newcommand\OBJ@OCG@view {} % needed in hyperref
143 \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`.

```

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

```

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

```

148 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
149 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:

150 \legacy_if:nT { Hy@setpdfversion }
151 {
152     \msg_warning:nn { hyp }{ pdfversion-disabled }
153 }
154 \Hy@DisableOption{pdfversion}

```

`\Acrobatmenu` should use the new internal link command

```

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

```

`\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!

```

159 % TODO should go at some time ...
160 % \kv@set@family@handler{Hyp}
161 % { \msg_warning:nnx {hyp}{unknown-key-to-Hyp}{#1} }

```

```

162 \cs_set_protected:Npn \hypersetup #1
163 {
164     %\kvsetkeys{Hyp} {#1}
165     \keys_set:nn { hyp / setup }{ #1 }
166 }
167 % TODO for now unknown keys should only give warnings.
168 \keys_define:nn { hyp / setup }
169 {
170     unknown .code:n =
171     {
172         \msg_warning:nnxxx { hyp } { unknown-key }
173         { hyp~/~setup }{ \l_keys_key_str } { #1 }
174     }
175 }

```

4 Compability commands

4.1 Metadata

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

```

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

```

182 \hook_new:n{hyp/link/cite}
183 \color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
184 \color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
185 \keys_define:nn { hyp / setup }
186 {
187     ,citecolor .code:n = {\__hyp_color_set:nn {hyp/color/cite}{#1}}
188     ,citebordercolor
189     .code:n = {\__hyp_color_set:nn {hyp/color/citeborder}{#1}}
190 }
191 \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
192 {
193     \keys_set:nn { hyp / setup }
194     {
195         ,linkbordercolor= hyp/color/citeborder
196         ,linkcolor      = hyp/color/cite
197     }
198 }

```


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

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

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

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

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 210 %
\__hyp_ref_check:nn    211 \cs_new_protected:Npn \__hyp_ref_label:nn #1 #2 %label/attributes
\__hyp_ref_value:nn    212 {
213   \bsphack
214   \ref_label:nn{#1}{#2}
215   \esphack
216 }
217 \cs_generate_variant:Nn \__hyp_ref_label:nn {en}
```

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

```
218 \prg_new_eq_conditional:Nn \__hyp_ref_if_exist:nn \ref_if_exist:nn { p , T , F , TF }
219 \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.

```
220 \cs_new_protected:Npn \__hyp_ref_check:nn #1 #2 %label/attribute
221 {
222   \__hyp_ref_if_exist:nnF {#1}{#2}
223   {
224     \protect\G@refundefinedtrue
225     \@latex@warning
226     {
227       Reference~'\tl_to_str:n {#1}'~with~attribute~'\tl_to_str:n {#2}'~
228       on~page~\thepage~\space undefined
229     }
230   }
```

```

230     }
231   }
232   \cs_generate_variant:Nn \__hyp_ref_check:nn {en}

```

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

```

233   \cs_new:Npn \__hyp_ref_value:nn #1 #2
234   {
235     \ref_value:nn{#1}{#2}
236   }
237   \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
238 \box_new:N \l__hyp_tmpa_box
239 \tl_new:N \l__hyp_tmpa_tl
240 \seq_new:N \l__hyp_tmpa_seq
241 \int_new:N \l__hyp_tmpa_int
242 \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
\__hyp_text_tmpa_str\g__hyp_text_tmpa_str
243 \tl_new:N \l__hyp_dest_name_tmpa_tl
244 \tl_new:N \l__hyp_uri_tmpa_tl
245 \tl_new:N \l__hyp_filename_tmpa_tl
246 \tl_new:N \l__hyp_para_tmpa_tl
247 \str_new:N \l__hyp_text_tmpa_str
248 \str_new:N \g__hyp_text_tmpa_str

```

TODO: document and check use!

(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.

```

249 \tl_const:Nn \c__hyp_dest_undefined_tl {UNDEFINED}

```

(End definition for \c__hyp_dest_undefined_tl.)

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

```

\c__hyp_map_hyp_annot_prop
250 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
251 {url,link,file,menu,run}
252 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
253 {
254   URI    = url,
255   GoTo   = link,
256   GoToR  = file,

```

```

257     Named = menu,
258     Launch= run
259 }
260 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
261 {
262     url    = URI,
263     link   = GoTo,
264     file   = GoToR,
265     menu   = Named,
266     run    = Launch
267 }
268

```

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

7.3 Variables

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

```

269 \tl_new:N \g__hyp_dest_pdfstartpage_tl
270 \tl_new:N \g__hyp_dest_pdfstartview_tl
271 \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

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

```

285 \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` (*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`

```
286 \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_colorrrun_bool`

`\l_hyp_annot_colormenu_bool`

```
287 \seq_map_inline:Nn \c__hyp_annot_types_seq
```

```
288 {
```

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

```
290 }
```

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

`\l_hyp_annot_ocgcolorlink_bool`

`\l_hyp_annot_ocgcolorurl_bool`

`\l_hyp_annot_ocgcolorfile_bool`

`\l_hyp_annot_ocgcolorrrun_bool`

`\l_hyp_annot_ocgcolormenu_bool`

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

```
291 \seq_map_inline:Nn \c__hyp_annot_types_seq
```

```
292 {
```

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

```
294 }
```

(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.

```
295 \seq_map_inline:Nn \c_pdfannot_link_types_seq
```

```
296 {
```

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

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

```
299 }
```

(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.

```
300 \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.

```

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

(End definition for `\c__hyp_dest_startview_regex`.)

7.7 PDF dictionaries

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

```

314 \pdfdict_new:n {l__hyp_page/Trans}
315 \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`

```

316 \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`

```

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

(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`

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

(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`

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

(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`

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

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

This is a special version for info keys:

```

\__hyp_text_pdfstring_info:nN

```

```

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

```

(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

```

```

348 \cs_new_protected:Npn \__hyp_PageLabels_gpush:
349 {
350   \pdfmanagement_add:nnx {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
351 }
352
353 \def\Hy@PutCatalog #1 {}
354
355
356 \legacy_if:nT { Hy@pdfpagelabels }
357 {

```

```

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

```

(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

```

```

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

```



```

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

```

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

```

(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:

```

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

```

```

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

```

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.

```

433
434 \cs_new_protected:Npn \hyper@link #1 #2 #3 %#1 context, #2=destination name, #3 content
435 {
436     \bool_if:NTF \l__hyp_annot_GoTo_bool
437     {
438         \Hy@VerboseLinkStart{#1}{#2}
439         \group_begin:
440         \hook_use:n {hyp/link/#1}
441         \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
442         \__hyp_link_goto_end:

```

```

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

```

10.3 URI links

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

```

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

```

```

493         \mode_leave_vertical:
494         \pdfannot_dict_put:nxn {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
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:nxn {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_dict_put:nmx {link/GoToR}{A}{<<\pdfdict_use:n {l_hyp/annot/A/GoToR}>>}
548     \pdfannot_link:nmx %expansion??
549     { GoToR }
550     {
551         % /A<<
552         % \pdfdict_use:n {l_hyp/annot/A/GoToR}
553         % >>
554     }
555     {
556         #1\Hy@xspace@end
557         \Hy@VerboseLinkStop %where is the start??
558     }
559     \group_end:
560 }
561 {#1}
562 }

```

10.5 Launch links

We define \hyper@linklaunch for naming consistency

```

563 \pdfdict_new:n {l_hyp/annot/A/Launch}
564 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
565 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
566
567 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
568 {
569     \bool_if:NTF \l__hyp_annot_Launch_bool
570     {
571         \group_begin:
572         \__hyp_text_pdfstring:nnN
573         { #1 }
574         { \l__hyp_text_enc_file_print_tl }
575         \l__hyp_filename_tmpa_tl
576         \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
577         \__hyp_text_pdfstring:noN
578         { #3 }
579         { \l__hyp_text_enc_para_print_tl }
580         \l__hyp_para_tmpa_tl
581         \bool_if:nTF
582         {
583             \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
584             ||
585             \pdf_version_compare_p:Nn > {1.9}
586         }
587         {
588             \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
589         }
590         {
591             \pdfdict_put:nmx {l_hyp/annot/A/Launch}{Win}
592             {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
593         }
594     }
595 }

```

```

594 \tl_if_empty:NF \Hy@href@nextactionraw
595 {
596   \str_remove_once:Nn \Hy@href@nextactionraw {/Next}
597   \pdfdict_put:nno{l_hyp/annot/A/Next}{Next}{\Hy@href@nextactionraw}
598 }
599 \mode_leave_vertical:
600 \pdfannot_dict_put:nnx {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}>>}
601 \pdfannot_link:nxn
602 { Launch }
603 {
604   % /A
605   % <<
606   % \pdfdict_use:n {l_hyp/annot/A/Launch}
607   % >>
608 }
609 {
610   #2\Hy@xspace@end
611   \Hy@VerboseLinkStop %where is the start??
612 }
613 \group_end:
614 }
615 {#2}
616 }

```

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

```

617 \use:x
618 { % filename, anchor text, linkname
619   \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str ##1 \exp_not:N \ \ ##2 #
620 }
621 {
622   \hyper@linklaunch {#1}{#2}{#3}
623 }

```

10.6 Named links (menu)

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

```

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

```

```

641         \pdfdict_put:nno{l_hyp/annot/A/Next}{Next}{\Hy@href@nextactionraw}
642     }
643     \pdfannot_dict_put:nrx {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}>>
644     \pdfannot_link:nxn { Named }
645     {
646     %         /A
647     %         <<
648     %         \pdfdict_use:n { l_hyp/annot/A/Named }
649     %         >>
650     }
651     {
652     #2
653     \Hy@xspace@end
654     \Hy@VerboseLinkStop
655     }
656     }
657     {
658     \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
659     #2
660     }
661     \group_end:
662 }
663 {#2}
664 }
665

```

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).

```

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

```

```

675 }
676
677 \cs_new_protected:Npn \__hyp_colormodel_export:wnnN [#1] #2 #3 #4
678 {
679     \color_export:nnnN {#1}{#2}{#3}#4
680 }
681
682 \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 color=red!50!blue.

```

683 \cs_new_protected:Npn \__hyp_color_select:n #1
684 {
685     \tl_if_head_eq_charcode:nNTF {#1}[ %]
686     {
687         \__hyp_colormodel_select_aux:wn #1
688     }
689     {
690         \color_select:n {#1}
691     }
692 }
693
694 \cs_new_protected:Npn \__hyp_colormodel_select_aux:wn [#1] #2
695 {
696     \color_select:nn {#1}{#2}
697 }
698
699 \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 color=red!50!blue.

```

700 \cs_new_protected:Npn \__hyp_color_set:nn #1 #2
701 {
702     \tl_if_head_eq_charcode:nNTF {#2}[ %]
703     {
704         \__hyp_colormodel_set_aux:nwn { #1 } #2
705     }
706     {
707         \color_set:nn {#1} {#2}
708     }

```



```

709 }
710
711 \cs_new_protected:Npn \__hyp_colormodel_set_aux:nwn #1 [#2] #3
712 {
713   \color_set:nnn {#1}{#2}{#3}
714 }
715
716 \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.

```

717 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
718 {
719   \hook_gput_code:nnn
720     {pdfannot/link/#2/begin}
721     {hyp/color}
722     {
723       \bool_if:cT { l_hyp_annot_color#1_bool }
724       {
725         \group_begin:
726         \color_select:n { hyp/color/#1}
727       }
728     }
729   \hook_gput_code:nnn
730     {pdfannot/link/#2/end}
731     {hyp/color}
732     {
733       \bool_if:cT { l_hyp_annot_color#1_bool }
734       {
735         \group_end:
736       }
737     }
738 }

```

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

```

739 \keys_define:nn { hyp / setup }
740 {
741   ,colorlinks .meta:n =
742   {
743     ,pdfborder={0~0~0}
744     ,pdfborderstyle=
745     ,colorurl =#1
746     ,colorlink =#1
747     ,colorrun =#1
748     ,colormenu =#1
749     ,colorfile =#1
750   }
751   ,colorlinks .default:n = {true}
752 }

```

```

colorurl (setup key)
colorlink (setup key) 753 \seq_map_inline:Nn \c__hyp_annot_types_seq
colorrun (setup key) 754 {
colormenu (setup key) 755 \keys_define:nn { hyp / setup }
colorfile (setup key) 756 {
urlcolor (setup key) 757 ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
linkcolor (setup key) 758 ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
runcolor (setup key) 759 }
menucolor (setup key) 760 }
filecolor (setup key) 761
allcolor (setup key) 762 \keys_define:nn { hyp / setup }
763 {
764 ,allcolors .meta:n =
765 {
766 ,urlcolor=#1
767 ,linkcolor=#1
768 ,runcolor=#1
769 ,filecolor=#1
770 ,menucolor=#1
771 }
772 ,allcolors .value_required:n = true
773 }

```

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

```

```

796             {link/#2}
797             { C }
798             { [\l__hyp_tmpa_tl] }
799         }
800     }
801 }
802 }
803
804 \keys_define:nn { hyp / setup }
805 {
806     ,allbordercolors .meta:n =
807     {
808         ,linkbordercolor=#1
809         ,urlbordercolor =#1
810         ,filebordercolor=#1
811         ,menubordercolor=#1
812         ,runbordercolor =#1
813     }
814     ,allbordercolors .value_required:n = true
815 }
816

```

11.3.2 Borderwidth and -arc

```

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

```

```

847         {link/##2}
848         { Border }
849     }
850 }
851 {
852     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
853     {
854         \pdfannot_dict_put:nnn
855         {link/##2}
856         { Border }
857         { [#1] }
858     }
859 }
860 }
861 ,pdfborder .initial:n = {0~0~1},
862 }

```

11.3.3 Borderstyle

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

```

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

```

```

896     }
897     {
898         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
899         {
900             \pdfannot_dict_put:nnn
901             {link/##2}
902             { BS }
903             { <<#1>> }
904         }
905     }
906 }
907 ,pdfborderstyle .initial:n = {},
908 }

```

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.

```

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

```

```

942 \pdfmanagement_add:nnx { Catalog / OCPProperties }{OCGs }{ \pdf_object_ref:n {l__hyp_ocg} }
943 \pdf_object_write:nx { l__hyp_ocg_config_dict_obj }
944 {
945   /OFF[\pdf_object_ref:n { l__hyp_ocg_print_dict_obj }]
946   /AS[
947     <<
948     /Event/View
949     /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
950     /Category[/View]
951     >>
952     <<
953     /Event/Print
954     /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
955     /Category[/Print]
956     >>
957     <<
958     /Event/Export
959     /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
960     /Category[/Print]
961     >>
962     ]
963   }
964 \pdfmanagement_add:nnx { Catalog / OCPProperties }{ D }{ \pdf_object_ref:n { l__hyp_ocg} }
965 \cs_gset:Npn \__hyp_ocg_init: {}
966 }

```

(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.

```

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

```

```

992             \pdf_bdcobject:nn {0C}{1__hyp_ocg_view_dict_obj}
993             \group_begin:
994             \color_select:n { hyp/color/#1 }
995             \box_use_drop:N \l__hyp_tmpa_box
996             \group_end:
997             \pdf_emc:
998         }
999     \group_end:
1000 }
1001 }
1002 }

```

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

```

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

```

```

1044     }
1045     ,ocgcolorlinks .default:n = {true}
1046   }
1047
1048   \seq_map_inline:Nn \c__hyp_annot_types_seq
1049   {
1050     \pdf_version_compare:NnTF < {1.5}
1051     {
1052       \keys_define:nn { hyp / setup }
1053       {
1054         ,ocgcolor#1 .code:n=
1055         {
1056           \msg_warning:nnxx
1057             { hyp }
1058             { ignore-deprecated-or-unknown-option-in-pdf-version }
1059             { ocgcolor#1 }
1060             { \pdf_version_major:.\pdf_version_minor: }
1061         }
1062       }
1063     }
1064     {
1065       \keys_define:nn { hyp / setup }
1066       {
1067         ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1068       }
1069     }
1070   }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1071   \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1072   {
1073     \keys_define:nn { hyp / setup }
1074     {
1075       ,#1highlight .choices:nn =
1076       { /I, /N, /O, /P }
1077       {
1078         \pdfannot_dict_put:nnn
1079           {link/#2}
1080           { H }
1081           { ##1 }
1082       }
1083     }
1084     ,#1highlight / .code:n =
1085     {
1086       \pdfannot_dict_remove:nn
1087         {link/#2}
1088         { H }
1089     }
1090   }
1091   ,#1highlight / unknown .code:n =
1092   {
1093     \msg_warning:nnxxx { hyp } { unknown-choice+empty }

```



```

1094         { pdfhighlight }
1095         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1096         { \exp_not:n {##1} }
1097     }
1098 }
1099 }
1100
1101
1102 \keys_define:nn { hyp / setup }
1103 {
1104     ,pdfhighlight .choices:nn =
1105     { /I, /N, /O, /P }
1106     {
1107         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1108         {
1109             \pdfannot_dict_put:nnn
1110             {link/###2}
1111             { H }
1112             { #1 }
1113         }
1114     }
1115     ,pdfhighlight / .code:n =
1116     {
1117         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1118         {
1119             \pdfannot_dict_remove:nn
1120             {link/##2}
1121             { H }
1122         }
1123     }
1124     ,pdfhighlight .initial:n = {/I},
1125     ,pdfhighlight / unknown .code:n =
1126     {
1127         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1128         { pdfhighlight }
1129         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1130         { \exp_not:n {#1} }
1131     }
1132 }

```

11.6 Hiding links

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

```

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

```

```

1142 }
1143
1144 \seq_map_inline:Nn \c__hyp_annot_types_seq
1145 {
1146   \keys_define:nn { hyp / setup }
1147   {
1148     hide#1 .meta:n =
1149     {
1150       ,color#1      = false
1151       ,ocgcolor#1   = false
1152       ,#1border     = { 0~0~0 }
1153       ,#1borderstyle =
1154     }
1155   }
1156 }

```

11.7 color schemes and settings

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

`colorscheme (setup key)`

```

1157 \keys_define:nn { hyp / setup }
1158 {
1159   colorscheme .code:n =
1160   {
1161     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1162     {
1163       \keys_set:nn { hyp /setup }
1164       {
1165         ##1 = ##2
1166       }
1167     }
1168   }
1169 }
1170 \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)`

```

1171 \keys_define:nn { hyp / setup }
1172 {
1173   ,unicode      .code:n = {}
1174   ,pdfencoding  .code:n = {}
1175   ,pdfversion   .code:n =
1176   {
1177     \msg_warning:nn { hyp }{ pdfversion-disabled }
1178   }
1179 }
1180 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
  debug (setup key) 1181 \keys_define:nn { hyp / setup }
  draft (setup key) 1182 {
  final (setup key) 1183   ,verbose .choice:
1184   ,verbose / true .code:n = { \Hy@verbosetrue}
1185   ,verbose / false .code:n = { \Hy@verbosefalse}
1186   ,verbose .default:n = {true}
1187   ,debug .meta:n = {verbose=#1}
1188   ,debug .default:n = {true}
1189 }
1190 \keys_define:nn { hyp / setup }
1191 {
1192   ,draft .code:n =
1193   {
1194     \Hy@drafttrue
1195     \PassOptionsToPackage{draft}{bookmark}
1196   }
1197   ,final .code:n =
1198   {
1199     \Hy@finaltrue
1200     \PassOptionsToPackage{final}{bookmark}
1201   }
1202 }

extension (setup key)
hypertextnames (setup key) 1203 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1204 {
  pageanchor (setup key) 1205   ,extension .tl_set:N = \XR@ext
  linktoc (setup key) 1206   ,extension .initial:n= pdf
  linktocpage (setup key) 1207   ,hypertextnames .choice:
  plainpages (setup key) 1208   ,hypertextnames / true .code:n = { \Hy@hypertextnamestrue}
localanchorname (setup key) 1209   ,hypertextnames / false .code:n = { \Hy@hypertextnamesfalse}
linkfileprefix (setup key) 1210   ,hypertextnames .default:n = {true}
1211   ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1212   ,localanchorname .choice:
1213   ,localanchorname / true .code:n = { \Hy@localanchorname true }
1214   ,localanchorname / false .code:n = { \Hy@localanchorname false }
1215   ,localanchorname .default:n = {true}
1216   ,naturalnames .choice:
1217   ,naturalnames / true .code:n = { \Hy@naturalnames true}
1218   ,naturalnames / false .code:n = { \Hy@naturalnames false}
1219   ,naturalnames .default:n = {true}
1220   ,pageanchor .choice:
1221   ,pageanchor / true .code:n = { \Hy@pageanchortrue}
1222   ,pageanchor / false .code:n = { \Hy@pageanchorfalse}
1223   ,pageanchor .default:n = {true}
1224   ,plainpages .choice:
1225   ,plainpages / true .code:n = { \Hy@plainpagestrue}
1226   ,plainpages / false .code:n = { \Hy@plainpagesfalse}
1227   ,plainpages .default:n = {true}

```

```

1228 }
1229
1230 \keys_define:nn { hyp / setup }
1231 {
1232   ,linktoc .choices:nn = { none, section, all, page }
1233   {
1234     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1235   }
1236   ,linktoc / unknown .code:n =
1237   {
1238     \msg_warning:nnxxx { hyp } { unknown-choice }
1239     { linktoc }
1240     { none, section, all, page }
1241     { \exp_not:n {#1} }
1242   }
1243   ,linktocpage .choice:
1244   ,linktocpage / true .meta:n = {linktoc=page}
1245   ,linktocpage / false .meta:n = {linktoc=section}
1246   ,linktocpage .default:n = true
1247 }
1248

```

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

```

url (setup key) 1249 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1250 {
menu (setup key) 1251   \keys_define:nn { hyp / setup }
run (setup key) 1252   {
1253     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1254   }
1255 }

```

```

1256 \keys_define:nn { hyp / setup }
1257 {
1258   ,baseurl .code:n =
1259   {
1260     \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1261     \tl_if_empty:NTF \l__hyp_tmpa_tl
1262     {
1263       \pdfmanagement_remove:nn {Catalog} { URI }
1264     }
1265     {
1266       \pdfmanagement_add:nnx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1267     }
1268   }
1269   %only false does something ...
1270   ,bookmarks .choice:
1271   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1272   ,bookmarks / true .code:n = {}
1273   ,bookmarks .default:n = {true}
1274   ,bookmarksnumbered .choice:
1275   ,bookmarksnumbered / false .code:n = { \Hy@bookmarksnumberedfalse }
1276   ,bookmarksnumbered / true .code:n = { \Hy@bookmarksnumberedtrue }
1277   ,bookmarksnumbered .default:n = {true}
1278   ,bookmarksopen .choice:

```

```

1279 ,bookmarksopen / false .code:n = { \Hy@bookmarksopenfalse }
1280 ,bookmarksopen / true .code:n = { \Hy@bookmarksopentruetrue }
1281 ,bookmarksopen .default:n = {true}
1282 ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1283 ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1284 ,pdfcenterwindow .choice:
1285 ,pdfcenterwindow / false .code:n =
1286 {
1287     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1288 }
1289 ,pdfcenterwindow / true .code:n =
1290 {
1291     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1292 }
1293 ,pdfcenterwindow / .code:n =
1294 {
1295     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1296 }
1297 ,pdfcenterwindow / unknown .code:n =
1298 {
1299     \msg_warning:nnxx { hyp } { no-bool }
1300     { pdfcenterwindow }
1301     { \exp_not:n {#1} }
1302 }
1303 ,pdfcenterwindow .default:n = true
1304 ,pdfdirection .choice:
1305 ,pdfdirection / L2R .code:n =
1306 {
1307     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1308 }
1309 ,pdfdirection / R2L .code:n =
1310 {
1311     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1312 }
1313 ,pdfdirection / .code:n =
1314 {
1315     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1316 }
1317 ,pdfdirection / unknown .code:n =
1318 {
1319     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1320     { pdfdirection }
1321     { L2R , R2L }
1322     { \exp_not:n {#1} }
1323 }
1324 ,pdfdisplaydoctitle .choice:
1325 ,pdfdisplaydoctitle / false .code:n =
1326 {
1327     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1328 }
1329 ,pdfdisplaydoctitle / true .code:n =
1330 {
1331     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1332 }

```

```

1333 ,pdffdisplaydoctitle .default:n = true
1334 ,pdfduplex .choices:nn =
1335 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1336 {
1337     \pdf_version_compare:NnTF > {1.6}
1338     {
1339         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1340         { PrintDuplex } { /#1 }
1341     }
1342     {
1343         \msg_warning:nnxx
1344         {hyp}
1345         {ignore-deprecated-or-unknown-option-in-pdf-version}
1346         {pdfduplex}
1347         {\pdf_version:}
1348     }
1349 }%
1350 ,pdfduplex / .code:n =
1351 {
1352     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1353 }
1354 ,pdfduplex / unknown .code:n =
1355 {
1356     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1357     { pdfduplex }
1358     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1359     { \exp_not:n {#1} }
1360 }
1361 ,pdffitwindow .choice:
1362 ,pdffitwindow / false .code:n =
1363 {
1364     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1365 }
1366 ,pdffitwindow / true .code:n =
1367 {
1368     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1369 }
1370 ,pdffitwindow / .code:n =
1371 {
1372     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1373 }
1374 ,pdffitwindow .default:n = true
1375 ,pdffitwindow / unknown .code:n =
1376 {
1377     \msg_warning:nnxx { hyp } { no-bool }
1378     { pdffitwindow }
1379     { \exp_not:n {#1} }
1380 }
1381 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1382 ,pdflinkmargin .initial:n = {1pt}
1383 ,pdfmenubar .choice:
1384 ,pdfmenubar / true .code:n =
1385 {
1386     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }

```

```

1387     }
1388 ,pdfmenubar / false .code:n =
1389     {
1390         \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1391         { HideMenubar } { true }
1392     }
1393 ,pdfmenubar / .code:n =
1394     {
1395         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1396     }
1397 ,pdfmenubar .default:n = true
1398 ,pdfmenubar / unknown .code:n =
1399     {
1400         \msg_warning:nnxx { hyp } { no-bool }
1401         { pdfmenubar }
1402         { \exp_not:n {#1} }
1403     }
1404 ,pdfnewwindow .choice:
1405 ,pdfnewwindow / true .code:n =
1406     {
1407         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1408         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1409     }
1410 ,pdfnewwindow / false .code:n =
1411     {
1412         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1413         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1414     }
1415 ,pdfnewwindow / .code:n =
1416     {
1417         \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1418         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1419     }
1420 ,pdfnonfullscreenpagemode .choices:nn =
1421     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1422     {
1423         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1424         { NonFullScreenPageMode } {/#1}
1425     }
1426 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1427     {
1428         \pdf_version_compare:NnTF < {1.6}
1429         {
1430             %message
1431         }
1432         {
1433             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1434             {NonFullScreenPageMode}{/UseAttachments}
1435         }
1436     }
1437 ,pdfnonfullscreenpagemode / .code:n =
1438     {
1439         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1440     }

```

```

1441 ,pdfnonfullscreenpagemode / unknown .code:n =
1442 {
1443   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1444   { pdfnonfullscreenpagemode }
1445   { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1446   { \exp_not:n {#1} }
1447 }
1448 ,pdfnumcopies .code:n =
1449 {
1450   \pdf_version_compare:NnTF > {1.6}
1451   {
1452     \tl_if_empty:nTF {#1}
1453     {
1454       \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1455     }
1456     {
1457       \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1458       {NumCopies}{#1}
1459     }
1460   }
1461   {
1462     \msg_warning:nnxx
1463     {hyp}
1464     {ignore-deprecated-or-unknown-option-in-pdf-version}
1465     {pdfnumcopies}
1466     {\pdf_version:}
1467   }
1468 }
1469 ,pdfpagelayout .choices:nn =
1470 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1471 { \pdfmanagement_add:nnx {Catalog} { PageLayout }{ /#1 } }
1472 ,pdfpagelayout / .code:n =
1473 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1474 ,pdfpagelayout / unknown .code:n =
1475 {
1476   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1477   { pdfpagelayout }
1478   { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1479   { \exp_not:n {#1} }
1480 }
1481 ,pdfpagemode .choices:nn =
1482 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1483 { \pdfmanagement_add:nnx {Catalog} { PageMode }{ /#1 } }
1484 ,pdfpagemode / UseAttachments .code:n =
1485 {
1486   \pdf_version_compare:NnTF > {1.5}
1487   {
1488     \pdfmanagement_add:nnx {Catalog} { PageMode }{ /UseAttachments }
1489   }
1490   {
1491     \msg_warning:nnxx
1492     {hyp}
1493     {ignore-deprecated-or-unknown-value-in-pdf-version}
1494     {UseAttachments}

```



```

1495         {\pdf_version:}
1496     }
1497 }
1498 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1499 ,pdfpagemode / unknown .code:n =
1500 {
1501     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1502     { pdfpagemode }
1503     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1504     { \exp_not:n {#1} }
1505 }
1506 ,pdfpagescrop .code:n =
1507 {
1508     \tl_if_empty:nTF %or blank?
1509     {
1510         \pdfmanagement_remove:nn {Pages} { CropBox }
1511     }
1512     {
1513         \pdfmanagement_add:nnx {Pages} { CropBox } { [#1] }
1514     }
1515 }
1516 ,pdfpicktraybypdfsize .choice:
1517 ,pdfpicktraybypdfsize / true .code:n =
1518 {
1519     \pdf_version_compare:NnTF > {1.6}
1520     {
1521         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1522         { PickTrayByPDFSize } { true }
1523     }
1524     {
1525         \msg_warning:nnxx
1526         {hyp}
1527         {ignore-deprecated-or-unknown-option-in-pdf-version}
1528         {pdfpicktraybypdfsize}
1529         {\pdf_version:}
1530     }
1531 }
1532 ,pdfpicktraybypdfsize / false .code:n =
1533 {
1534     \pdf_version_compare:NnTF > {1.6}
1535     {
1536         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1537         { PickTrayByPDFSize } { false }
1538     }
1539     {
1540         \msg_warning:nnxx
1541         {hyp}
1542         {ignore-deprecated-or-unknown-option-in-pdf-version}
1543         {pdfpicktraybypdfsize}
1544         {\pdf_version:}
1545     }
1546 }
1547 ,pdfpicktraybypdfsize / .code:n =
1548 {

```

```

1549     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1550 }
1551 ,pdfpicktraybypdfsize / unknown .code:n =
1552 {
1553     \msg_warning:nxxx { hyp } { no-bool }
1554     { picktraybypdfsize }
1555     { \exp_not:n {#1} }
1556 }
1557 ,pdfprintarea .choices:nn =
1558 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1559 {
1560     \pdf_version_compare:NnTF < {2.0}
1561     {
1562         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1563         { PrintArea } { /#1 }
1564     }
1565     {
1566         \msg_warning:nxxx
1567         {hyp}
1568         {ignore-deprecated-or-unknown-option-in-pdf-version}
1569         {pdfprintarea}
1570         {\pdf_version:}
1571     }
1572 }%
1573 ,pdfprintarea / .code:n =
1574 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1575 ,pdfprintarea / unknown .code:n =
1576 {
1577     \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1578     { pdfprintarea }
1579     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1580     { \exp_not:n {#1} }
1581 }
1582 ,pdfprintclip .choices:nn =
1583 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1584 {
1585     \pdf_version_compare:NnTF < {2.0}
1586     {
1587         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1588         { PrintClip } { /#1 }
1589     }
1590     {
1591         \msg_warning:nxxx
1592         {hyp}
1593         {ignore-deprecated-or-unknown-option-in-pdf-version}
1594         {pdfprintclip}
1595         {\pdf_version:}
1596     }
1597 }%
1598 ,pdfprintclip / .code:n =
1599 {
1600     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1601 }
1602 ,pdfprintclip / unknown .code:n =

```

```

1603     {
1604         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1605         { pdfprintclip }
1606         { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1607         { \exp_not:n {#1} }
1608     }
1609 ,pdfprintpagerange .code:n =
1610 {
1611     \pdf_version_compare:NnTF > {1.6}
1612     {
1613         \tl_if_empty:nTF { #1}
1614         {
1615             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1616             { PrintPageRange }
1617         }
1618         {
1619             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1620             {PrintPageRange}{[#1]}
1621         }
1622     }
1623     {
1624         \msg_warning:nnxx
1625         {hyp}
1626         {ignore-deprecated-or-unknown-option-in-pdf-version}
1627         {pdfprintpagerange}
1628         {\pdf_version:}
1629     }
1630 }
1631 ,pdfprintscaling .choices:nn =
1632 { None, AppDefault }
1633 {
1634     \pdf_version_compare:NnTF > {1.5}
1635     {
1636         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1637         { PrintScaling } { /#1 }
1638     }
1639     {
1640         \msg_warning:nnxx
1641         {hyp}
1642         {ignore-deprecated-or-unknown-option-in-pdf-version}
1643         {pdfprintscaling}
1644         {\pdf_version:}
1645     }
1646 }%
1647 ,pdfprintscaling / .code:n =
1648 {
1649     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1650 }
1651 ,pdfprintscaling / unknown .code:n =
1652 {
1653     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1654     { pdfprintarea }
1655     { None, AppDefault }
1656     { \exp_not:n {#1} }

```

```

1657     }
1658 ,pdfremotestartview .code:n =
1659 {
1660     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1661     \exp_args:NNV
1662     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1663     {
1664         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1665     }
1666     {
1667         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1668         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1669     }
1670 }
1671 ,pdfremotestartview .initial:n = {Fit}
1672 % pdfstartpage is special as it shares code with pdfstartview
1673 ,pdfstartpage .code:n =
1674 {
1675     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1676     \bool_if:nTF
1677     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1678     {
1679         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1680     }
1681     {
1682         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1683         {
1684             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1685         }
1686     }
1687 }
1688 ,pdfstartpage .initial:n =1
1689 ,pdfstartview .code:n =
1690 {
1691     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1692     \exp_args:NNV
1693     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1694     {
1695         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1696     }
1697     {
1698         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1699         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1700     }
1701     \bool_if:nTF
1702     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1703     {
1704         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1705     }
1706     {
1707         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1708         {
1709             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1710         }

```

```

1711     }
1712 }
1713 ,pdfstartview .initial:n = Fit
1714 ,pdftoolbar .choice:
1715 ,pdftoolbar / true .code:n =
1716 {
1717     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1718 }
1719 ,pdftoolbar / false .code:n =
1720 {
1721     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1722     { HideToolbar } { true }
1723 }
1724 ,pdftoolbar / true .code:n =
1725 {
1726     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1727 }
1728 ,pdftoolbar .default:n = true
1729 ,pdftoolbar / unknown .code:n =
1730 {
1731     \msg_warning:nxxx { hyp } { no-bool }
1732     { pdftoolbar }
1733     { \exp_not:n {#1} }
1734 }
1735 % pdfview see below.
1736 ,pdfviewarea .choices:nn =
1737 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1738 {
1739     \pdf_version_compare:NnTF < {2.0}
1740     {
1741         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1742         { ViewArea } { /#1 }
1743     }
1744     {
1745         \msg_warning:nxxx
1746         {hyp}
1747         {ignore-deprecated-or-unknown-option-in-pdf-version}
1748         {pdfviewarea}
1749         {\pdf_version:}
1750     }
1751 }%
1752 ,pdfviewarea / .code:n =
1753 {
1754     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1755 }
1756 ,pdfviewarea / unknown .code:n =
1757 {
1758     \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1759     { pdfviewarea }
1760     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1761     { \exp_not:n {#1} }
1762 }
1763 ,pdfviewclip .choices:nn =
1764 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }

```

```

1765 {
1766   \pdf_version_compare:NnTF < {2.0}
1767   {
1768     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1769     { ViewClip } { /#1 }
1770   }
1771   {
1772     \msg_warning:nnxx
1773     {hyp}
1774     {ignore-deprecated-or-unknown-option-in-pdf-version}
1775     {pdfviewclip}
1776     {\pdf_version:}
1777   }
1778   }%
1779 ,pdfviewclip / .code:n =
1780 {
1781   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
1782 }
1783 ,pdfviewclip / unknown .code:n =
1784 {
1785   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1786   { pdfviewclip }
1787   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1788   { \exp_not:n {#1} }
1789 }
1790 ,pdfwindowui .choice:
1791 ,pdfwindowui / true .code:n =
1792 {
1793   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
1794 }
1795 ,pdfwindowui / false .code:n =
1796 {
1797   \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1798   { HideWindowUI } { true }
1799 }
1800 ,pdfwindowui / .code:n =
1801 {
1802   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
1803 }
1804 ,pdfwindowui / unknown .code:n =
1805 {
1806   \msg_warning:nnxx { hyp } { no-bool }
1807   { pdfwindowui }
1808   { \exp_not:n {#1} }
1809 }
1810 ,pdfwindowui .default:n = true
1811 }

```

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

```

1812 \keys_define:nn { hyp / setup }
1813 {
1814   ,pdfview .code:n =
1815   {
1816     \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}

```

```

1817 \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
1818 {
1819   { xyz }
1820   {
1821     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
1822     {
1823       \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
1824       \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
1825       {
1826         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1827       }
1828       {
1829         \tl_set:Nx \l__hyp_dest_pdfview_tl
1830         {
1831           \fp_eval:n { \l__hyp_tmpa_tl * 100 }
1832         }
1833       }
1834     }
1835     {
1836       \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1837     }
1838   }
1839   { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
1840   { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
1841   { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
1842   { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
1843   { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
1844   { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
1845   { fitr }
1846   {
1847     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
1848     {
1849       \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
1850     }
1851     {
1852       %ensure 4 values ...
1853       \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
1854       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1855       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1856       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1857       \hbox_set_to_wd:Nnn \l__hyp_dest_box
1858       {
1859         \fp_eval:n
1860         {
1861           round
1862           (
1863             abs
1864             (
1865               \seq_item:Nn\l__hyp_tmpa_seq{4}
1866               -
1867               (\seq_item:Nn\l__hyp_tmpa_seq{2})
1868             ),
1869           3
1870         }

```

```

1871         }bp
1872     }{}
1873     \box_set_dp:Nn \l__hyp_dest_box
1874     {
1875         \fp_eval:n
1876         {
1877             round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
1878         }bp
1879     }
1880     \box_set_ht:Nn \l__hyp_dest_box
1881     {
1882         \seq_item:Nn\l__hyp_tmpa_seq{5}bp
1883     }
1884 }
1885 }
1886 }
1887 {
1888     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
1889     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
1890 }
1891 }
1892 ,pdfview .initial:n = {xyz}
1893 }

```

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.

```

1894 \keys_define:nn { hyp / setup }
1895 {
1896     ,pdflang .code:n =
1897     {
1898         \tl_if_empty:nTF { #1 }
1899         {
1900             \pdfmanagement_remove:nn {Catalog} { Lang }
1901         }
1902         {
1903             \pdfmanagement_add:nnx {Catalog} { Lang } { (#1) }
1904         }
1905         \__hyp_store_metadata:nn {pdflang}{#1}
1906     }
1907 }

```

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*) 1908 \cs_new_protected:Npn __hyp_setup_info_key:nn #1 #2

pdfsubject (*setup key*) 1909 {

pdfproducer (*setup key*) 1910 \keys_define:nn { hyp / setup }

pdfkeywords (*setup key*) 1911 {


```

1912 pdf#1 .code:n =
1913 {
1914   \tl_if_blank:nTF {##1}
1915   {
1916     \pdfmanagement_remove:nn {Info}{#2}
1917   }
1918   {
1919     \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
1920     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
1921     {
1922       \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
1923     }
1924   }
1925   \__hyp_store_metadata:nn {pdf#1}{##1}
1926 }
1927 }
1928 \keys_define:nn { hyp / info }
1929 {
1930   #2 .code:n =
1931   {
1932     \tl_if_blank:nTF {##1}
1933     {
1934       \pdfmanagement_remove:nn {Info}{#2}
1935     }
1936     {
1937       \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
1938       \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
1939       {
1940         \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
1941       }
1942     }
1943     \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
1944   }
1945   ,unknown .code:n =
1946   {
1947     \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
1948     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
1949     {
1950       \exp_args:Nno
1951       \pdfmanagement_add:nnx {Info}
1952       { \l_keys_key_str } {\l__hyp_tmpa_str}
1953     }
1954   }
1955 }
1956 }
1957 \__hyp_setup_info_key:nn {author} {Author}
1958 \__hyp_setup_info_key:nn {title} {Title}
1959 \__hyp_setup_info_key:nn {producer} {Producer}
1960 \__hyp_setup_info_key:nn {creator} {Creator}
1961 % ignored key: addtopdfcreator
1962 \__hyp_setup_info_key:nn {subject} {Subject}
1963 \__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.

```
1964 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
1965 {
1966   \keys_define:nn { hyp / setup }
1967   {
1968     pdf#1 .code:n =
1969     {
1970       \tl_if_blank:nTF {##1}
1971       {
1972         \pdfmanagement_remove:nn {Info}{#2}
1973       }
1974       {
1975         \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
1976       }
1977       \__hyp_store_metadata:nn {pdf#1}{##1}
1978     }
1979   }
1980   \keys_define:nn { hyp / info }
1981   {
1982     #2 .code:n =
1983     {
1984       \tl_if_blank:nTF {##1}
1985       {
1986         \pdfmanagement_remove:nn {Info}{#2}
1987       }
1988       {
1989         \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
1990       }
1991       \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
1992     }
1993   }
1994 }
1995
1996 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
1997 \__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 ...

```
1998 \keys_define:nn { hyp / setup }
1999 {
2000   ,pdftrapped .code:n =
2001   {
2002     \exp_args:Nne
2003     \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2004   }
2005   ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2006   {
2007     \pdfmanagement_add:nnx {Info}{Trapped}
2008     {/
2009       \str_uppercase:f { \str_head:n { #1 } }
2010       \str_lowercase:f { \str_tail:n { #1 } }
2011     }
2012     \__hyp_store_metadata:nx {pdftrapped}
2013     {
2014       \str_uppercase:f { \str_head:n { #1 } }
```

```

2015         \str_lowercase:f { \str_tail:n { #1 } }
2016     }
2017 }
2018 ,_pdftrapped / unknown .code:n =
2019 {
2020     \msg_warning:nnxxx { hyp } { unknown-choice }
2021     { pdftrapped }
2022     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2023     { \exp_not:n {#1} }
2024 }
2025 }

```

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

```

2026 \keys_define:nn { hyp / setup }
2027 {
2028     pdfinfo .code:n =
2029     {
2030         \keys_set:nn { hyp / info } { #1 }
2031     }
2032 }

```

Now we set some default values

```

2033 \keys_set:nn { hyp / setup } {pdfcreator = LaTeX-with-hyperref}
2034 \keys_set:nn { hyp / setup } {pdfauthor  = }
2035 \keys_set:nn { hyp / setup } {pdftitle   = }
2036 \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:

```

2037 \clist_map_inline:nn
2038 {
2039     ,pdfcopyright
2040     ,pdftype
2041     ,pdflicenseurl
2042     ,pdfauthortitle
2043     ,pdfcaptionwriter
2044     ,pdfmetalang
2045     ,pdfapart
2046     ,pdfaconformance
2047     ,pdfuapart
2048     ,pdfxstandard
2049     ,pdfsource
2050     ,pdfdocumentid
2051     ,pdfinstanceid
2052     ,pdfversionid
2053     ,pdfrendition
2054     ,pdfpublication
2055     ,pdfpubtype
2056     ,pdfbytes
2057     ,pdfnumpages
2058     ,pdfissn

```

```

2059 ,pdfeissn
2060 ,pdfisbn
2061 ,pdfbookedition
2062 ,pdfpublisher
2063 ,pdfvolumenum
2064 ,pdfissuenum
2065 ,pdfpagerange
2066 ,pdfdoi
2067 ,pdfurl
2068 ,pdfidentifier
2069 ,pdfsubtitle
2070 ,pdfpubstatus
2071 ,pdfcontactaddress
2072 ,pdfcontactcity
2073 ,pdfcontactregion
2074 ,pdfcontactpostcode
2075 ,pdfcontactcountry
2076 ,pdfcontactphone
2077 ,pdfcontactemail
2078 ,pdfcontacturl
2079 ,pdfdate
2080 }
2081 {
2082   \keys_define:nn { hyp / setup }
2083   {
2084     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2085   }
2086 }

```

12.5 Transitions

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

```

2087 \keys_define:nn { hyp / setup }
2088 {
2089   pdfpageduration .code:n =
2090   {
2091     \tl_if_blank:nTF { #1 }
2092     {
2093       \pdfmanagement_remove:nn {Page}{Dur}
2094     }
2095     {
2096       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2097     }
2098   }
2099 }

```

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

```
2100 \keys_define:nn { hyp / setup }
2101 {
2102   pdfpagetransition .code:n =
2103   {
2104     \tl_if_blank:nTF {#1}
2105     {
2106       \pdfmanagement_remove:nn {Page}{Trans}
2107     }
2108     {
2109       \group_begin:
2110       \keys_set:nn { hyp / trans }{style=R,#1}
2111       \pdf_object_unnamed_write:nx { dict }
2112       {
2113         \pdfdict_use:n {l__hyp_page/Trans}
2114       }
2115       \pdfmanagement_add:nnx {Page}{Trans}{\pdf_object_ref_last:}
2116       \group_end:
2117     }
2118   }
2119 }
2120 \keys_define:nn { hyp / trans }
2121 {
2122   ,style .choices:nn =
2123   {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2124   { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2125   ,style / unknown .code:n =
2126   {
2127     \msg_warning:nnxxx { hyp } { unknown-choice }
2128     { trans / style }
2129     { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2130     { \exp_not:n {#1} }
2131   }
2132   ,duration .code:n =
```

```

2133     {
2134         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{#1}
2135     }
2136     ,direction .choices:nn =
2137     {H,V}
2138     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2139     ,direction .choices:nn =
2140     {0,90,180,270,315}
2141     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ #1 } }
2142     ,direction / None .code:n =
2143     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ /None } }
2144     ,direction / unknown .code:n =
2145     {
2146         \msg_warning:nnxxx { hyp } { unknown-choice }
2147         { trans / direction }
2148         {
2149             H~(horizontal,~only~Split,~Blinds),
2150             V~(vertical,~only~Split,~Blinds),
2151             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2152             90~(bottom~to~top,~only~Wipe),
2153             180~(right~to~left,~only~Wipe),
2154             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2155             315~(top~left~to~bottom,~only~Glitter),
2156             None~(only~Fly)
2157         }
2158         { \exp_not:n {#1} }
2159     }
2160     ,motion .choices:nn =
2161     {I,0}
2162     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2163     ,motion / unknown .code:n =
2164     {
2165         \msg_warning:nnxxx { hyp } { unknown-choice }
2166         { trans / motion }
2167         { I~(inwards) , 0~(outwards) }
2168         { \exp_not:n {#1} }
2169     }
2170     ,scale .code:n =
2171     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2172     ,opaque .choices:nn = {true,false}
2173     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2174     ,opaque / unknown .code:n =
2175     {
2176         \msg_warning:nnxxx { hyp } { unknown-choice }
2177         { trans / B }
2178         { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2179         { \exp_not:n {#1} }
2180     }
2181     % try to set unknown keys as style
2182     ,unknown .code:n =
2183     {
2184         % warning ...
2185         \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2186     }

```

```

2187 }
      Finally we process the package option list, to get most keys working
2188 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}
      Unfinished Form field code
2189 \NewDocumentCommand \MakeFieldObject { m m }
2190 {
2191   \pdfxform_new:nnn { #2 }{} { #1 }
2192 }
2193
2194
2195 \prop_new:N \g__hyp_AcroForm_CoFields_prop
2196 \prop_new:N \g__hyp_AcroForm_Fields_prop
2197
2198 \let\HyField@afields\ltx@empty
2199 \let\HyField@cofields\ltx@empty
2200 %% UF test for old pdftex removed
2201 %\let\HyField@AuxAddToFields\ltx@gobble
2202 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2203 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2204
2205 % the value doesn't matter, but with a prop we avoid duplicates and it is
2206 % clearly faster than removing them from a sequence
2207 \def\HyField@AuxAddToFields#1
2208 {
2209   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2210 }%
2211
2212 %fields with empty key get a value too -- lets hope that
2213 %this give the expected behaviour
2214 \def\HyField@AuxAddToCoFields #1 #2
2215 {
2216   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2217 }
2218
2219 \Hy@AtBeginDocument
2220 {
2221   \if@files
2222     \immediate\write\@mainaux{%
2223       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2224     }%
2225     \immediate\write\@mainaux{%
2226       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2227     }%
2228   \fi
2229   \let\HyField@AfterAuxOpen\@firstofone
2230 }%
2231
2232 \def\HyField@AddToFields
2233 {
2234   \exp_args:Nx\HyField__hypAddToFields
2235   {
2236     \pdfannot_box_ref_last:
2237   }

```

```

2238 \ifx\Fld@calculate@code\ltx@empty
2239 \else
2240 \begingroup
2241 \Hy@safe@activetrue
2242 \edef\Hy@temp{%
2243 \endgroup
2244 \if@filesw
2245 \write\@mainaux
2246 {
2247 \string\HyField@AuxAddToCoFields
2248 {
2249 \Fld@calculate@sortkey
2250 }
2251 {
2252 \pdfannot_box_ref_last:
2253 }
2254 }
2255 \fi
2256 }%
2257 \Hy@temp
2258 \fi
2259 }%
2260
2261 \def\HyField__hypAddToFields#1{
2262 \HyField@AfterAuxOpen{%
2263 \if@filesw
2264 \write\@mainaux{%
2265 \string\HyField@AuxAddToFields{#1}%
2266 }%
2267 \fi
2268 }%
2269 }%
2270
2271 \ExplSyntaxOff
2272 \ExplSyntaxOn
2273 \tl_new:N \l__hyp_CheckmarkYes_tl
2274 \tl_set:Nn \l__hyp_CheckmarkYes_tl { __hyp_xform_CheckMarkYes }
2275 \tl_new:N \l__hyp_CheckmarkOff_tl
2276 \tl_set:Nn \l__hyp_CheckmarkOff_tl { __hyp_xform_CheckMarkOff }
2277
2278 \def\@Form[#1]
2279 {
2280 \@ifundefined{textcolor}{\let\textcolor\gobble}{}
2281 \kvsetkeys{Form}{#1}
2282 \pdf@ifdraftmode{}
2283 {
2284 \Hy@FormObjects
2285 \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2286 {
2287 \pdfmanagement_add:nnx { Catalog / AcroForm } { Fields }{##1}
2288 %\pdfmanagement_show:n { Catalog / AcroForm }
2289 }
2290 \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2291 {

```



```

2292         \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2293         {
2294             \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2295         }
2296     \seq_sort:Nn \l__hyp_tmpa_seq
2297     {
2298         \int_compare:nNnTF { \pdf@strcmp{##1}{##2} } > { 0 }
2299         { \sort_return_swapped: }
2300         { \sort_return_same: }
2301     }
2302     \seq_map_inline:Nn \l__hyp_tmpa_seq
2303     {
2304         \pdfmanagement_add:nnx { Catalog / AcroForm }
2305         { CO }
2306         {
2307             \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2308         }
2309     }
2310 }
2311 \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2312 {ZaDb} {\pdf_object_ref:n {l__hyp_font_zapdingbats_obj} }
2313 \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2314 {Helv} {\pdf_object_ref:n {l__hyp_font_helvetica_obj} }
2315 \pdfmanagement_add:nnx {Catalog /AcroForm}
2316 {DA}{(/Helv~10~Tf~0~g)}
2317 \pdfmeta_standard_verify:nTF {form_no_NeedAppearance}
2318 {
2319     \legacy_if:nT { HyField@NeedAppearances }
2320     {
2321         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2322     }
2323 }
2324 {
2325     \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2326 }
2327 }
2328 \MakeFieldObject
2329 {
2330     \group_begin:
2331     \fontfamily{pzd}
2332     \fontencoding{U}
2333     \fontseries{m}
2334     \fontshape{n}
2335     \selectfont
2336     \char123
2337     \group_end:
2338 }
2339 {__hyp_xform_Ding}
2340 \MakeFieldObject
2341 {
2342     \group_begin:
2343     \fontfamily{pzd}
2344     \fontencoding{U}
2345     \fontseries{m}

```

```

2346         \fontshape{n}
2347         \selectfont
2348         \phantom{\char123}
2349         \group_end:
2350     }
2351     {__hyp_xform_DingOff}
2352 \MakeFieldObject
2353 {
2354     \group_begin:
2355     \fontfamily{pzd}
2356     \fontencoding{U}
2357     \fontseries{m}
2358     \fontshape{n}
2359     \selectfont
2360     \char51
2361     \group_end:
2362 }
2363 {__hyp_xform_CheckMarkYes}
2364 \MakeFieldObject
2365 {
2366     \group_begin:
2367     \fontfamily{pzd}
2368     \fontencoding{U}
2369     \fontseries{m}
2370     \fontshape{n}
2371     \selectfont
2372     \phantom{\char51} %perhaps xetex needs some small glyph ..
2373     \group_end:
2374 }
2375 {__hyp_xform_CheckMarkOff}
2376 \MakeFieldObject
2377 {
2378     \fbox{\textcolor{yellow}{\textsf{Submit}}}} %color?
2379 }
2380 {__hyp_xform_Submit}
2381 \MakeFieldObject
2382 {
2383     \fbox{\textcolor{yellow}{\textsf{SubmitP}}}} %color?
2384 }
2385 {__hyp_xform_SubmitP}
2386 }
2387 \ExplSyntaxOff
2388 \let\@endForm\ltx@empty
2389 \let\HyAnn@AbsPageLabel\ltx@empty
2390 \let\Fld@pageobjref\ltx@empty
2391
2392 \ExplSyntaxOn
2393 \newcount\HyAnn@Count
2394 \HyAnn@Count=\ltx@zero
2395 \def\HyAnn@AbsPageLabel
2396 {
2397     \global\advance\HyAnn@Count by\ltx@one
2398     %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2399     %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}

```

```

2400     \%zref@refused{HyAnn@\the\HyAnn@Count}%
2401     \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
2402     \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
2403 }%
2404 \def\Fld@pageobjref
2405 {
2406     \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
2407     {
2408         /P~\pdf_pageobject_ref:n
2409         {
2410             \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
2411         }
2412     }
2413 }
2414 \ExplSyntaxOff
2415 \ExplSyntaxOn
2416 %% check if the attr should be set through
2417 %% hooks.
2418 %% check if options are missing.
2419 \def\@TextField[#1]#2{% parameters, label
2420     \def\Fld@name{#2}%
2421     \let\Fld@default\ltx@empty
2422     \let\Fld@value\@empty
2423     \def\Fld@width{\DefaultWidthofText}%
2424     \def\Fld@height{%
2425         \ifFld@multiline
2426             \DefaultHeightofTextMultiline
2427         \else
2428             \DefaultHeightofText
2429         \fi
2430 }%
2431 \begingroup
2432     \expandafter\HyField@SetKeys\expandafter{%
2433         \DefaultOptionsofText,#1%
2434     }%
2435     \PDFForm@Name
2436     \HyField@FlagsText
2437     \ifFld@hidden\def\Fld@width{1sp}\fi
2438     \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2439     \LayoutTextField{#2}{%
2440         \leavevmode
2441         \HyAnn@AbsPageLabel
2442         \Hy@escapeform\PDFForm@Text
2443         \pdfannot_box:nnnn
2444         {\Fld@width}
2445         {\Fld@height}
2446         {Opt} %is this correct?
2447         {\PDFForm@Text}
2448         \MakeTextField{\Fld@width}{\Fld@height}
2449         \HyField@AddToFields
2450     }%
2451 \endgroup
2452 }
2453 \providecommand\@curropt{}

```

```

2454 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2455 \def\Fld@name{#2}
2456 \let\Fld@default\relax
2457 \let\Fld@value\relax
2458 \def\Fld@width{\DefaultWidthofChoiceMenu}
2459 \def\Fld@height{\DefaultHeightofChoiceMenu}
2460 \beginngroup
2461 \Fld@menulength=0 %
2462 \@tempdima\z@
2463 \clist_map_variable:nNn { #3 } \@curropt
2464 %\@for\@curropt:=#3\do
2465 {%
2466 \expandafter\Fld@checkequals\@curropt==\%
2467 \Hy@StepCount\Fld@menulength
2468 \settowidth{\@tempdimb}{\@currDisplay}%
2469 \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2470 }%
2471 \advance\@tempdima by~15\p@
2472 \beginngroup
2473 \HyField@SetKeys{#1}
2474 \edef\x{\endgroup
2475 \noexpand\expandafter
2476 \noexpand\HyField@SetKeys
2477 \noexpand\expandafter{%
2478 \expandafter\noexpand\csname DefaultOptionsof%
2479 \ifFld@radio
2480 Radio%
2481 \else
2482 \ifFld@combo
2483 \ifFld@popdown
2484 PopdownBox%
2485 \else
2486 ComboBox%
2487 \fi
2488 \else
2489 ListBox%
2490 \fi
2491 \fi
2492 \endcsname
2493 }%
2494 }\x
2495 \HyField@SetKeys{#1}%
2496 \PDFForm@Name
2497 \ifFld@hidden\def\Fld@width{1sp}\fi
2498 \ifx\Fld@value\relax
2499 \let\Fld@value\Fld@default
2500 \fi
2501 \LayoutChoiceField{#2}{%
2502 \ifFld@radio
2503 \HyField@FlagsRadioButton
2504 \__hypRadio{#3}%
2505 \else
2506 \beginngroup
2507 \HyField@FlagsChoice

```

```

2508         \ifdim\Fld@width<\@tempdima
2509         \ifdim\@tempdima<1cm\@tempdima1cm\fi
2510         \edef\Fld@width{\the\@tempdima}%
2511     \fi
2512     \ifFld@combo
2513     \else
2514         \@tempdima=\the\Fld@menulength\Fld@charsize
2515         \advance\@tempdima by~\Fld@borderwidth bp %
2516         \advance\@tempdima by~\Fld@borderwidth bp %
2517         \edef\Fld@height{\the\@tempdima}%
2518     \fi
2519     \__hypListBox{#3}%
2520 \endgroup
2521 \fi
2522 }%
2523 \endgroup
2524 }
2525 \tl_new:N \l__hyp_RadioYes_tl
2526 \tl_set:Nn \l__hyp_RadioYes_tl { __hyp_xform_Ding }
2527 \def\__hypRadio#1{%
2528     \Fld@listcount=0-%
2529     %\show\Fld@default
2530     \EdefEscapeName\Fld@default{\Fld@default}%
2531     \clist_map_variable:nNn { #1 } \@curropt
2532     %\@for\@curropt:=#1\do
2533     {%
2534         \expandafter\Fld@checkequals\@curropt==\%
2535         \EdefEscapeName\@currValue{\@currValue}%
2536         \Hy@StepCount\Fld@listcount
2537         \@currDisplay\space
2538         \leavevmode
2539         \HyAnn@AbsPageLabel
2540         \Hy@escapeform\PDFForm@Radio
2541         \pdfannot_box:nnnn
2542             {\Fld@width}
2543             {\Fld@height}
2544             {Opt} %is this correct?
2545             {
2546                 \PDFForm@Radio
2547                 /AP
2548                 <<
2549                 /N
2550                 <<
2551                 /\@currValue\c_space_tl \pdfxform_ref:o {__hyp_xform_Ding}
2552                 %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2553                 >>
2554                 >>
2555             }
2556             {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2557     \int_compare:nNnT { \Fld@listcount } = { 1 }
2558     { \HyField@AddToFields }
2559     \c_space_tl % deliberate space between radio buttons
2560     % to do: --> should be configurable
2561 }%

```

```

2562 }
2563
2564 \newcount\Fld@listcount
2565 \def\__hypListbox#1
2566 {
2567   \HyField@PDFChoices{#1}
2568   \mode_leave_vertical:
2569   \HyAnn@AbsPageLabel
2570   \Hy@escapeform\PDFForm@List
2571   \pdf_link_user:nnn
2572   {widget} %perhaps we need more types??
2573   {\PDFForm@List}
2574   {\MakeChoiceField{\Fld@width}{\Fld@height}}
2575   \HyField@AddToFields
2576 }
2577
2578
2579 \def\@PushButton[#1]#2{% parameters, label
2580   \def\Fld@name{#2}%
2581   \group_begin:
2582     \exp_args:No\HyField@SetKeys
2583     {
2584       \DefaultOptionsofPushButton,#1
2585     }
2586     \PDFForm@Name
2587     \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2588     {
2589       \HyField@FlagsPushButton
2590       \legacy_if:nT {Fld@hidden}
2591       {
2592         \def\Fld@width{1sp}
2593       }
2594       \LayoutPushButtonField
2595       {
2596         \mode_leave_vertical:
2597         \HyAnn@AbsPageLabel
2598         \Hy@escapeform\PDFForm@Push
2599         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2600         \pdfannot_box:nnnn
2601         {\box_wd:N\l_tmpa_box}
2602         {\box_ht:N\l_tmpa_box}
2603         {\box_dp:N\l_tmpa_box} %is this correct?
2604         {\PDFForm@Push}
2605         {\box_use:N\l_tmpa_box}
2606         \HyField@AddToFields
2607       }
2608     }
2609     {
2610       \msg_error:nn { hyp }{ pdfa-no-push-button }
2611       \LayoutPushButtonField
2612       {
2613         \mode_leave_vertical:
2614         \MakeButtonField{#2}
2615       }

```

```

2616     }
2617     \group_end:
2618 }
2619
2620 \def\@Submit[#1]#2
2621 {
2622     \def\Fld@width {\DefaultWidthofSubmit}
2623     \def\Fld@height{\DefaultHeightofSubmit}
2624     \group_begin:
2625         \exp_args:No\HyField@SetKeys
2626         {
2627             \DefaultOptionsofSubmit,#1
2628         }
2629     \HyField@FlagsPushButton
2630     \HyField@FlagsSubmit
2631     \legacy_if:nT { Fld@hidden }
2632     {
2633         \def\Fld@width{1sp}
2634     }
2635     \mode_leave_vertical:
2636     \HyAnn@AbsPageLabel
2637     \Hy@escapeform\PDFForm@Submit
2638     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2639     \pdfannot_box:nnnn
2640         {\box_wd:N\l_tmpa_box}
2641         {\box_ht:N\l_tmpa_box}
2642         {\box_dp:N\l_tmpa_box} %is this correct?
2643     {
2644         \PDFForm@Submit
2645         /AP<<
2646             /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2647             /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2648         >>
2649     }
2650     \HyField@AddToFields
2651     \box_use:N\l_tmpa_box
2652
2653     \group_end:
2654 }
2655
2656 \def\@Reset[#1]#2
2657 {
2658     \def\Fld@width {\DefaultWidthofReset}
2659     \def\Fld@height{\DefaultHeightofReset}
2660     \group_begin:
2661         \exp_args:No\HyField@SetKeys
2662         {
2663             \DefaultOptionsofReset,#1
2664         }
2665     \mode_leave_vertical:
2666     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2667     {
2668         \HyField@FlagsPushButton
2669         \legacy_if:nT { Fld@hidden }

```

```

2670         { \def\Fld@width{1sp} }
2671         \HyAnn@AbsPageLabel
2672         \Hy@escapeform\PDFForm@Reset
2673         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2674         \pdfannot_box:nnnn
2675         {\box_wd:N\l_tmpa_box}
2676         {\box_ht:N\l_tmpa_box}
2677         {\box_dp:N\l_tmpa_box} %is this correct?
2678         { \PDFForm@Reset }
2679         \HyField@AddToFields
2680         \box_use:N \l_tmpa_box
2681     }
2682     {
2683         \msg_error:nn { hyp }{ pdfa-no-reset-button }
2684         \MakeButtonField{#2}
2685     }
2686 \group_end:
2687 }
2688
2689 \def\@CheckBox[#1]#2
2690 {% parameters, label
2691     \def\Fld@name{#2}
2692     \def\Fld@default{0}
2693     \group_begin:
2694         \def\Fld@width {\DefaultWidthofCheckBox}
2695         \def\Fld@height{\DefaultHeightofCheckBox}
2696         \exp_args:No\HyField@SetKeys
2697         {
2698             \DefaultOptionsofCheckBox,#1
2699         }
2700         \PDFForm@Name
2701         \HyField@FlagsCheckBox
2702         \legacy_if:nT { Fld@hidden }
2703         {
2704             \def\Fld@width{1sp}
2705         }
2706         \LayoutCheckField{#2}
2707         {
2708             \mode_leave_vertical:
2709             \HyAnn@AbsPageLabel
2710             \Hy@escapeform\PDFForm@Check
2711             \pdfannot_box:nnnn
2712             {\Fld@width}
2713             {\Fld@height}
2714             {Opt} %is this correct?
2715             {\PDFForm@Check}
2716             \HyField@AddToFields %check if this works with xelatex ...
2717         }
2718     \group_end:
2719 }
2720 \ExplSyntaxOff
2721
2722 %hm. Should a luatex driver use type1 fonts in fields????
2723 \ExplSyntaxOn

```



```

2724 \def\Hy@FormObjects
2725 {
2726   \pdf_object_new:nn {l__hyp_encoding_pdfdoc_obj } { dict }
2727   \pdf_object_new:nn {l__hyp_font_zapfdingbats_obj } { dict }
2728   \pdf_object_new:nn {l__hyp_font_helvetica_obj } { dict }
2729   \pdf_object_write:nx {l__hyp_encoding_pdfdoc_obj }
2730   {
2731     /Type/Encoding
2732     /Differences[
2733       24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2734       /ring/tilde
2735       \c_space_tl
2736       39/quotesingle
2737       \c_space_tl
2738       96/grave %
2739       \iow_newline:
2740       128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2741       /fraction/guilsinglleft/guilsinglright/minus/perthousand
2742       /quotedblbase/quotedblleft/quotedblright/quoteleft
2743       /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2744       /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2745       \iow_newline:
2746       164/currency
2747       \c_space_tl
2748       166/brokenbar
2749       \c_space_tl
2750       168/dieresis/copyright/ordfeminine
2751       \c_space_tl
2752       172/logicalnot/.notdef/registered/macron/degree/plusminus
2753       /twosuperior/threesuperior/acute/mu
2754       \c_space_tl
2755       183/periodcentered/cedilla/onesuperior/ordmasculine
2756       \c_space_tl
2757       188/onequarter/onehalf/threequarters
2758       \iow_newline:
2759       192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
2760       /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
2761       /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
2762       /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
2763       /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
2764       /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
2765       /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
2766       /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
2767       /ocircumflex/otilde/odieresis/divide/oshlash/ugrave
2768       /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
2769     ]
2770   }
2771   \pdf_object_write:nn {l__hyp_font_zapfdingbats_obj }
2772   {
2773     /Type/Font
2774     /Subtype/Type1
2775     /Name/ZaDb
2776     /BaseFont/ZapfDingbats
2777   }

```

```

2778 \pdf_object_write:nx {l__hyp_font_helvetica_obj }
2779 {
2780     /Type/Font
2781     /Subtype/Type1
2782     /Name/Helv
2783     /BaseFont/Helvetica
2784     /Encoding~\pdf_object_ref:n { l__hyp_encoding_pdfdoc_obj }
2785 }
2786 \global\let\Hy@FormObjects\relax
2787 }
2788 \ExplSyntaxOff
2789 \providecommand*{\Fld@pageobjref}{}
2790 \ifcsname pdf@escapestring\endcsname
2791     \def\Hy@escapeform#1{%
2792         \ifHy@pdfescapeform
2793             \let\Hy@escapestring\pdfescapestring
2794         \else
2795             \let\Hy@escapestring\@firstofone
2796         \fi
2797     }%
2798     \Hy@escapeform{}%
2799 \else
2800     \let\Hy@escapestring\@firstofone
2801     \def\Hy@escapeform#1{%
2802         \ifHy@pdfescapeform
2803             \def\Hy@escapestring##1{%
2804                 \noexpand\Hy@escapestring{\noexpand##1}%
2805             }%
2806             \edef\Hy@temp{#1}%
2807             \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{}\@nil
2808             \def\Hy@escapestring##1{%
2809                 \@ifundefined{Hy@esc@\string##1}{%
2810                     ##1%
2811                     \ThisShouldNotHappen
2812                 }{%
2813                     \csname Hy@esc@\string##1\endcsname
2814                 }%
2815             }%
2816         \else
2817             \let\Hy@escapestring\@firstofone
2818         \fi
2819     }%
2820     \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
2821         \ifx\\#3\\%
2822         \else
2823             \expandafter
2824             \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
2825             \ltx@ReturnAfterFi{%
2826                 \Hy__hypescapeform#3\@nil
2827             }%
2828         \fi
2829     }%
2830 \fi
2831 \def\PDFForm@Name{%

```

```

2832 \PDFForm__hypName\Fld@name
2833 \ifx\Fld@altname\relax
2834 \else
2835 \PDFForm__hypName\Fld@altname
2836 \fi
2837 \ifx\Fld@mappingname\relax
2838 \else
2839 \PDFForm__hypName\Fld@mappingname
2840 \fi
2841 }
2842 \def\PDFForm__hypName#1{%
2843 \begingroup
2844 \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
2845 \ifHy@unicode
2846 \Hy@unicodedefalse
2847 \fi
2848 \fi
2849 \pdfstringdef\Hy@gtemp#1%
2850 \endgroup
2851 \let#1\Hy@gtemp
2852 }
2853 \def\Fld@X@additionalactions{%
2854 \ifx\Fld@keystroke@code\@empty
2855 \else
2856 /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
2857 \fi
2858 \ifx\Fld@format@code\@empty
2859 \else
2860 /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
2861 \fi
2862 \ifx\Fld@validate@code\@empty
2863 \else
2864 /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
2865 \fi
2866 \ifx\Fld@calculate@code\@empty
2867 \else
2868 /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
2869 \fi
2870 \ifx\Fld@onfocus@code\@empty
2871 \else
2872 /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
2873 \fi
2874 \ifx\Fld@onblur@code\@empty
2875 \else
2876 /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
2877 \fi
2878 \ifx\Fld@onmousedown@code\@empty
2879 \else
2880 /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
2881 \fi
2882 \ifx\Fld@onmouseup@code\@empty
2883 \else
2884 /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
2885 \fi

```

```

2886 \ifx\Fld@onenter@code\@empty
2887 \else
2888   /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
2889 \fi
2890 \ifx\Fld@onexit@code\@empty
2891 \else
2892   /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
2893 \fi
2894 }
2895 \ExplSyntaxOn
2896 \def\Fld@additionalactions
2897 {%
2898   \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{}
2899   {
2900     \pdfmeta_standard_verify:nT {annot_widget_no_AA}
2901     {/AA<<\Fld@X@additionalactions>>}
2902   }
2903 }
2904 \ExplSyntaxOff
2905 \def\Fld@annotnames{%
2906   /T(\Fld@name)%
2907   \ifx\Fld@altname\relax
2908   \else
2909     /TU(\Fld@altname)%
2910   \fi
2911   \ifx\Fld@mappingname\relax
2912   \else
2913     /TM(\Fld@mappingname)%
2914   \fi
2915 }
2916 \ExplSyntaxOn
2917 \def\PDFForm@Check
2918 {
2919   /Subtype/Widget
2920   ~\Fld@annotflags
2921   ~\Fld@pageobjref
2922   ~\Fld@annotnames
2923   /FT/Btn
2924   \Fld@flags
2925   /Q~\Fld@align
2926   /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
2927   /AP
2928   <<
2929   /N
2930   <<
2931     /Yes~\pdfxform_ref:o{\l__hyp_CheckmarkYes_t1}
2932     /Off~\pdfxform_ref:o{\l__hyp_CheckmarkOff_t1}
2933   >>
2934   >>
2935 /MK<<
2936   \int_compare:nNnF {\Fld@rotation}={0}
2937   {
2938     /R~\Fld@rotation
2939   }

```

```

2940 \tl_if_empty:NF\Fld@bordercolor
2941 {
2942   /BC[\Fld@bordercolor]
2943 }
2944 \tl_if_empty:NF\Fld@bcolor
2945 {
2946   /BG[\Fld@bcolor]
2947 }
2948 /CA(\Hy@escapestring{\Fld@cbsymbol})%
2949 >>
2950 /DA
2951 (
2952   /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
2953   \tl_if_empty:NF \Fld@color
2954   {
2955     \c_space_tl \Fld@color
2956   }
2957 )
2958 /H/P
2959 \legacy_if:nTF {Fld@checked}
2960 {
2961   /V/Yes /AS/Yes
2962 }
2963 {
2964   /V/Off /AS/Off
2965 }
2966 \Fld@additionalactions
2967 }
2968 \ExplSyntaxOff
2969 \ExplSyntaxOn
2970 \def\PDFForm@Push
2971 {
2972   /Subtype/Widget
2973   ~\Fld@annotflags
2974   ~\Fld@pageobjref
2975   ~\Fld@annotnames
2976   /FT/Btn
2977   ~\Fld@flags
2978   /H/P
2979   /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
2980   \bool_if:nT
2981   {
2982     !\int_compare_p:nNn {\Fld@rotation} = {0}
2983     ||
2984     \tl_if_exist_p:N \Fld@bordercolor
2985   }
2986   {
2987     /MK
2988     <<
2989     \int_compare:nNnF {\Fld@rotation} = {0}
2990     {
2991       /R~\Fld@rotation
2992     }
2993     \tl_if_exist:NT \Fld@bordercolor

```

```

2994         {
2995             /BC[\Fld@bordercolor]
2996         }
2997     >>
2998 }
2999 /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3000 \Fld@additionalactions
3001 }
3002
3003 \ExplSyntaxOff
3004 \def\PDFForm@List{%
3005     /Subtype/Widget%
3006     \Fld@annotflags
3007     \Fld@pageobjref
3008     \Fld@annotnames
3009     /FT/Ch%
3010     \Fld@flags
3011     /Q \Fld@align
3012     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3013     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3014         \ifx\Fld@bordercolor\relax\else 1\fi
3015         \ifx\Fld@bcolor\relax \else 1\fi
3016         \space
3017 \else
3018     /MK<<%
3019     \ifnum\Fld@rotation=\z@
3020     \else
3021         /R \Fld@rotation
3022     \fi
3023     \ifx\Fld@bordercolor\relax
3024     \else
3025         /BC[\Fld@bordercolor]%
3026     \fi
3027     \ifx\Fld@bcolor\relax
3028     \else
3029         /BG[\Fld@bcolor]%
3030     \fi
3031     >>%
3032 \fi
3033 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3034     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3035 \Fld@choices
3036 \Fld@additionalactions
3037 }
3038 \ExplSyntaxOn
3039 \def\PDFForm@Radio
3040 {
3041     /Subtype/Widget
3042     ~\Fld@annotflags
3043     ~\Fld@pageobjref
3044     ~\Fld@annotnames
3045     /FT/Btn
3046     \Fld@flags
3047     /H/P

```

```

3048 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3049 /MK<<
3050 \ifnum\Fld@rotation=\z@
3051 \else
3052 /R~\Fld@rotation
3053 \fi
3054 \ifx\Fld@bordercolor\relax
3055 \else
3056 /BC[\Fld@bordercolor]%
3057 \fi
3058 \ifx\Fld@bcolor\relax
3059 \else
3060 /BG[\Fld@bcolor]%
3061 \fi
3062 /CA(\Hy@escapestring{\Fld@radiosymbol})%
3063 >>
3064 /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3065 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3066 \ifx\Fld@default\@empty
3067 /V/Off%
3068 /DV/Off%
3069 \else
3070 /V/\Fld@default
3071 /DV/\Fld@default
3072 \fi
3073 \Fld@additionalactions
3074 }
3075 \ExplSyntaxOff
3076 \ExplSyntaxOn
3077 % Does an appeareance dict make sense here?
3078 \def\PDFForm@Text
3079 {
3080 /Subtype/Widget
3081 ~\Fld@annotflags
3082 ~\Fld@pageobjref
3083 ~\Fld@annotnames
3084 /FT/Tx
3085 ~\Fld@flags
3086 /Q~\Fld@align
3087 /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3088 \bool_if:nT
3089 {
3090 !\int_compare_p:nNn {\Fld@rotation} = {0}
3091 ||
3092 \tl_if_exist_p:N \Fld@bordercolor
3093 ||
3094 \tl_if_exist_p:N \Fld@bcolor
3095 }
3096 {
3097 /MK
3098 <<
3099 \int_compare:nNnF {\Fld@rotation} = {0}
3100 {
3101 /R~\Fld@rotation

```

```

3102         }
3103         \tl_if_exist:NT \Fld@bordercolor
3104         {
3105             /BC[\Fld@bordercolor]
3106         }
3107         \tl_if_exist:NT \Fld@bcolor
3108         {
3109             /BG[\Fld@bcolor]
3110         }
3111     >>
3112 }
3113 /DA
3114 (
3115     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3116     \tl_if_empty:NF {\c_space_tl\Fld@color}
3117 )
3118 /DV(\Hy@escapestring{\Fld@default})
3119 /V(\Hy@escapestring{\Fld@value})
3120 ~\Fld@additionalactions
3121 \int_compare:nNnT { \Fld@maxlen}>{0}
3122 {
3123     /MaxLen~\Fld@maxlen
3124 }
3125 }
3126 \ExplSyntaxOff
3127
3128 \def\PDFForm@Submit{%
3129     /Subtype/Widget%
3130     \Fld@annotflags
3131     \Fld@pageobjref
3132     \Fld@annotnames
3133     /FT/Btn%
3134     \Fld@flags
3135     /H/P%
3136     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3137     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3138         \ifx\Fld@bordercolor\relax\else 1\fi
3139         \space
3140 \else
3141     /MK<<%
3142         \ifnum\Fld@rotation=\z@
3143         \else
3144             /R \Fld@rotation
3145         \fi
3146         \ifx\Fld@bordercolor\relax
3147         \else
3148             /BC[\Fld@bordercolor]%
3149         \fi
3150     >>%
3151 \fi
3152 /A<<%
3153     /S/SubmitForm%
3154     /F<<%
3155     /FS/URL%

```



```

3156         /F(\Hy@escapestring{\Form@action})%
3157     >>%
3158     \Fld@submitflags
3159 >>%
3160     \Fld@additionalactions
3161 }
3162 \ExplSyntaxOn
3163     \def\PDFForm@Reset{%
3164         /Subtype/Widget%
3165         \Fld@annotflags
3166         \Fld@pageobjref
3167         \Fld@annotnames
3168         /FT/Btn%
3169         \Fld@flags
3170         /H/P%
3171         /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3172         \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3173             \ifx\Fld@bordercolor\relax\else 1\fi
3174             \space
3175         \else
3176             /MK<<%
3177             \ifnum\Fld@rotation=\z@
3178             \else
3179                 /R~\Fld@rotation
3180             \fi
3181             \ifx\Fld@bordercolor\relax
3182             \else
3183                 /BC[\Fld@bordercolor]%
3184             \fi
3185             >>%
3186         \fi
3187         /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3188         /A<</S/ResetForm>>%
3189         \Fld@additionalactions
3190     }%
3191
3192
3193 %these patterns are used in hyperref checks.
3194 %it is unclear if they are really useful and if a backend support is
3195 %needed.
3196 \str_case:VnF \c_sys_backend_str
3197 {
3198     { pdfmode }
3199     {
3200         \def\HyPat@ObjRef
3201         {
3202             [0-9]*[1-9][0-9]*~0~R
3203         }
3204     }
3205     { dvipdfmx }
3206     {
3207         \def\HyPat@ObjRef
3208         {
3209             @[\~]+

```

```

3210     }
3211   }
3212   { xdvipdfmx }
3213   {
3214     \def\HyPat@ObjRef
3215     {
3216       @[\~]+
3217     }
3218   }
3219 }
3220 { %also set in hyperref sty, so probably not needed.
3221   \def\HyPat@ObjRef/{.+}
3222 }
3223
3224
3225 \ExplSyntaxOff
3226 % UF: removed Hy@writebookmark
3227 %   \Hy@currentbookmarklevel{0}
3228 %   \Hy@numberline
3229 %   \__hypwritetorep
3230 %   counter{bookmark@seq@number}
3231 % removed \HyPsd@SanitizeForOutFile, not needed
3232 % removed \currentpdfbookmark, defined by bookmark,
3233 % should use \newcommand there
3234 % removed \subpdfbookmark, defined by bookmark,
3235 % should use \newcommand there
3236 % removed \belowpdfbookmark, defined by bookmark,
3237 % should use \newcommand there
3238 % removed \pdfbookmark, defined by bookmark,
3239 % \BOOKMARK
3240 % \@BOOKMARK
3241 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3242 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3243 %% removed \ReadBookmarks / unneeded with bookmark.
3244 %% removed \Hy@OutlineName
3245 %% removed \check@bm@number
3246 %% removed \calc@bm@number
3247
3248 \ifHy@implicit
3249 \else
3250   \expandafter\endinput
3251 \fi
3252 \newlength\Hy@SectionHShift
3253 \def\Hy@SectionAnchorHref#1{%
3254   \ifx\protect\@typeset@protect
3255     \Hy__hypSectionAnchor{#1}%
3256   \fi
3257 }
3258 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%
3259   \leavevmode
3260   \hbox to 0pt{%
3261     \kern-\Hy@SectionHShift
3262     \Hy@raisedlink{%
3263       \hyper@anchorstart{#1}\hyper@anchorend

```

```

3264     }%
3265     \hss
3266 }%
3267 }
3268 \let\H@old@ssect\@ssect
3269 \def\@ssect#1#2#3#4#5{%
3270   \Hy@MakeCurrentHrefAuto{section*}%
3271   \setlength{\Hy@SectionHShift}{#1}%
3272   \begingroup
3273     \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3274     \toks\tw@ \expandafter{%
3275       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3276       #5%
3277     }%
3278   \edef\x{\endgroup
3279     \the\toks@\the\toks\tw@}%
3280 } \x
3281 }
3282 \let\H@old@schapter\@schapter
3283 \def\@schapter#1{%
3284   \begingroup
3285     \let\@mkboth\@gobbletwo
3286     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3287     \Hy@raisedlink{%
3288       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3289     }%
3290   \endgroup
3291   \H@old@schapter{#1}%
3292 }
3293 \ltx@ifundefined{@chapter}{}{%
3294   \let\Hy@org@chapter\@chapter
3295   \def\@chapter{%
3296     \def\Hy@next{%
3297       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3298       \Hy@raisedlink{%
3299         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3300       }%
3301     }%
3302     \ifnum\c@secnumdepth>\m@ne
3303       \ltx@ifundefined{if@mainmatter}%
3304       \iftrue{\csname if@mainmatter\endcsname}%
3305       \let\Hy@next\relax
3306     \fi
3307     \fi
3308     \Hy@next
3309     \Hy@org@chapter
3310   }%
3311 }
3312 \let\H@old@part\@part
3313 \begingroup\expandafter\expandafter\expandafter\endgroup
3314 \expandafter\ifx\csname chapter\endcsname\relax
3315   \let\Hy@secnum@part\z@
3316 \else
3317   \let\Hy@secnum@part\m@ne

```

```

3318 \fi
3319 \def\@part{%
3320   \ifnum\Hy@secnum@part>\c@secnumdepth
3321     \phantomsection
3322   \fi
3323   \H@old@part
3324 }
3325 \let\H@old@spart\@spart
3326 \def\@spart#1{%
3327   \Hy@MakeCurrentHrefAuto{part*}%
3328   \Hy@raisedlink{%
3329     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3330   }%
3331   \H@old@spart{#1}%
3332 }
3333 \let\H@old@sect\@sect
3334 \def\@sect#1#2#3#4#5#6[#7]#8{%
3335   \ifnum #2>\c@secnumdepth
3336     \expandafter\@firstoftwo
3337   \else
3338     \expandafter\@secondoftwo
3339   \fi
3340   {%
3341     \Hy@MakeCurrentHrefAuto{section*}%
3342     \setlength{\Hy@SectionHShift}{#3}%
3343     \begingroup
3344       \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3345       \toks\tw\expandafter{%
3346         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3347         #8%
3348       }%
3349       \edef\x{\endgroup
3350         \the\toks@{\the\toks\tw@}%
3351       }\x
3352     }{%
3353       \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3354     }%
3355   }
3356   \expandafter\def\csname Parent-4\endcsname{}
3357   \expandafter\def\csname Parent-3\endcsname{}
3358   \expandafter\def\csname Parent-2\endcsname{}
3359   \expandafter\def\csname Parent-1\endcsname{}
3360   \expandafter\def\csname Parent0\endcsname{}
3361   \expandafter\def\csname Parent1\endcsname{}
3362   \expandafter\def\csname Parent2\endcsname{}
3363   \expandafter\def\csname Parent3\endcsname{}
3364   \expandafter\def\csname Parent4\endcsname{}
3365   %%
3366   %% End of file 'hgeneric-testphase.def'.
3367   \</package>
3368   \<colorscheme>
3369   % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3370   % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3371   % linkcolor=

```

```

3372 %,filecolor=
3373 %,urlcolor=
3374 %,menucolor=
3375 %,runcolor=
3376 %,linkbordercolor=
3377 %,filebordercolor=
3378 %,urlbordercolor=
3379 %,menubordercolor=
3380 %,runbordercolor=
3381
3382 \prop_const_from_keyval:Nn \c__hyp_colorscheme_original_prop
3383 {
3384     linkcolor      = [rgb]{1,0,0}, %red
3385     filecolor      = [rgb]{0,1,1}, %cyan
3386     urlcolor       = [rgb]{1,0,1}, %magenta
3387     menucolor      = [rgb]{1, 0, 0}, %red
3388     runcolor       = [rgb]{0,1,1}, %cyan
3389     %-----
3390     linkbordercolor = [rgb]{1, 0 ,0 },
3391     filebordercolor = [rgb]{0, .5, .5},
3392     urlbordercolor  = [rgb]{0, 1, 1},
3393     menubordercolor = [rgb]{1, 0, 0},
3394     runbordercolor  = [rgb]{0, .7, .7}
3395 }
3396
3397 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3398 {
3399     linkcolor      = [rgb]{0,0.2,0.6},
3400     filecolor      = [rgb]{0.8,0,0.8},
3401     urlcolor       = [rgb]{0.8,0,0.8},
3402     menucolor      = [rgb]{0,0.2,0.6},
3403     runcolor       = [rgb]{0.8,0,0.8},
3404     %----- %-----
3405     linkbordercolor = [rgb]{0,0.2,0.6},
3406     filebordercolor = [rgb]{0.8,0,0.8},
3407     urlbordercolor  = [rgb]{0.8,0,0.8},
3408     menubordercolor = [rgb]{0,0.2,0.6},
3409     runbordercolor  = [rgb]{0.8,0,0.8}
3410 }
3411
3412 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3413 { %two colors: intern/extern
3414     linkcolor      = [rgb]{0.79216, 0, 0.12549},
3415     filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3416     urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3417     menucolor      = [rgb]{0.79216, 0, 0.12549 },
3418     runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3419     %----- %-----
3420     linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3421     filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3422     urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3423     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3424     runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3425 }

```

```

3426
3427 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3428 { %all darkgray
3429   linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3430   filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3431   urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3432   menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3433   runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3434   %----- %-----
3435   linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3436   filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3437   urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3438   menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3439   runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3440 }
3441
3442 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3443 { %dvipsnam.def
3444   linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3445   filecolor      = [rgb]{1, 0, 0}, %Red
3446   urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3447   menucolor      = [rgb]{1, 0, 0}, %Red
3448   runcolor       = [rgb]{1, 0, 0}, %Red
3449   %----- %-----
3450   linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3451   filebordercolor = [rgb]{1, 0, 0}, %Red
3452   urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3453   menubordercolor = [rgb]{1, 0, 0}, %Red
3454   runbordercolor  = [rgb]{1, 0, 0} %Red
3455 }
3456
3457 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3458 { %dvipsnam.def
3459   linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3460   filecolor      = [rgb]{0, 1, 0}, %Green
3461   urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3462   menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3463   runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3464   %----- %-----
3465   linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3466   filebordercolor = [rgb]{0, 1, 0}, %Green
3467   urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3468   menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3469   runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3470 }
3471
3472
3473 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3474 {
3475   linkcolor      = [rgb]{0.50196, 0, 0.02353},
3476   filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3477   urlcolor       = [rgb]{0.54118, 0, 0.52941},
3478   menucolor      = [rgb]{0.44706, 0.45882, 0},
3479   runcolor       = [rgb]{0.07451, 0.46667, 0.46275},

```

```

3480 %----- %-----
3481 linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3482 filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3483 urlbordercolor = [rgb]{0.724708, 0.4, 0.717646},
3484 menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3485 runbordercolor = [rgb]{0.444706, 0.680002, 0.67765}
3486 }
3487
3488 \prop_const_from_keyval:Nn \c_hyp_colorscheme_henryford_prop
3489 {
3490 linkcolor = [rgb]{0,0,0},
3491 filecolor = [rgb]{0,0,0},
3492 urlcolor = [rgb]{0,0,0},
3493 menucolor = [rgb]{0,0,0},
3494 runcolor = [rgb]{0,0,0},
3495 %----- %-----
3496 linkbordercolor = [rgb]{0,0,0},
3497 filebordercolor = [rgb]{0,0,0},
3498 urlbordercolor = [rgb]{0,0,0},
3499 menubordercolor = [rgb]{0,0,0},
3500 runbordercolor = [rgb]{0,0,0}
3501 }
3502 </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		C	
\#	490	\char	2336, 2348, 2360, 2372
\%	491	\chardef	136
\.	305, 309, 311	\cite	25
@curropt commands:		clist commands:	
\@curropt:	2464, 2532	\clist_map_function:nN	111, 120
\\	18, 19, 20,	\clist_map_inline:nn	2037
	27, 31, 41, 61, 68, 75, 82, 89, 104,	\clist_map_variable:nNn	2463, 2531
	109, 110, 118, 119, 128, 129, 130,	color commands:	
	131, 132, 138, 619, 2466, 2534, 2821	\color_export:nnN	673
_	303, 305, 309, 311	\color_export:nnnN	679
A		\color_select:n	690, 726, 994
\A	303	\color_select:nn	696
\Acrobatmenu	15, 155	\color_set:nn	2, 184, 707
\addcontentsline	10	\color_set:nnn	2, 183, 713
\AddToDocumentProperties	179	color names:	
\AddToHook	204	hyp/annot/file	286
\advance	2397, 2471, 2515, 2516	hyp/annot/link	286
allcolor (hypersetup key)	753	hyp/annot/menu	286
\author	2	hyp/annot/run	286
B		hyp/annot/url	286
\b	307	colorfile (hypersetup key)	753
\begingroup	2240, 2431, 2460, 2472,	colorlink (hypersetup key)	753
	2506, 2843, 3272, 3284, 3313, 3343	colorlinks (hypersetup key)	739
\belowpdfbookmark	3236	colormenu (hypersetup key)	753
\BOOKMARK	3239	colorrun (hypersetup key)	753
bookmarkstype (hypersetup key)	10	colorscheme (hypersetup key)	1, 1157
bool commands:		colorurl (hypersetup key)	753
\bool_if:NTF	436, 450, 461,	cs commands:	
	474, 518, 569, 630, 723, 733, 973, 984	\cs_generate_variant:Nn	135,
\bool_if:nTF	581, 1676, 1701, 2980, 3088		181, 217, 232, 237, 341, 682, 699, 716
\bool_lazy_and:nnTF	199	\cs_gset:Npn	965
\bool_new:N	289, 293, 297	\cs_if_exist_p:N	200
\bool_set_true:N	298	\cs_new:Npn	104, 233
bordercolormodel (hypersetup key)	10, 774	\cs_new_protected:Npn	176, 211,
box commands:			220, 317, 321, 325, 329, 342, 344,
\box_dp:N	379, 2603, 2642, 2677		348, 390, 396, 403, 407, 429, 434,
\box_ht:N	378, 2602, 2641, 2676		448, 459, 472, 516, 567, 628, 666,
\box_new:N	238, 300		677, 683, 694, 700, 711, 909, 1908, 1964
\box_set_dp:Nn	1873	\cs_set_eq:NN	
\box_set_ht:Nn	1880		139, 148, 149, 490, 491, 1234
\box_use:N	990, 2605, 2651, 2680	\cs_set_protected:Npn	
\box_use_drop:N	995		137, 162, 358, 619
\box_wd:N	377, 2601, 2640, 2675	\csname	2478, 2813,
\l_tmpa_box	2599, 2601, 2602,		2824, 3304, 3314, 3356, 3357, 3358,
	2603, 2605, 2638, 2640, 2641, 2642,		3359, 3360, 3361, 3362, 3363, 3364
	2651, 2673, 2675, 2676, 2677, 2680	\currentpdfbookmark	3232

D	
<code>\d</code>	305, 309, 311
<code>debug</code> (hypersetup key)	<u>1181</u>
<code>\DeclareDocumentMetadata</code>	
.	3, 8, <u>11</u> , <u>12</u> , <u>15</u> , 20, 28, 34
<code>\DeclareRobustCommand</code>	3258
<code>\def</code>	143,
353, 2203, 2207, 2214, 2232, 2261,	
2278, 2395, 2404, 2419, 2420, 2423,	
2424, 2437, 2438, 2454, 2455, 2458,	
2459, 2497, 2527, 2565, 2579, 2580,	
2592, 2620, 2622, 2623, 2633, 2656,	
2658, 2659, 2670, 2689, 2691, 2692,	
2694, 2695, 2704, 2724, 2791, 2801,	
2803, 2808, 2820, 2831, 2842, 2853,	
2896, 2905, 2917, 2970, 3004, 3039,	
3078, 3128, 3163, 3200, 3207, 3214,	
3221, 3253, 3269, 3283, 3295, 3296,	
3319, 3326, 3334, 3356, 3357, 3358,	
3359, 3360, 3361, 3362, 3363, 3364	
<code>\DefaultHeightofCheckBox</code>	2695
<code>\DefaultHeightofChoiceMenu</code>	2459
<code>\DefaultHeightofReset</code>	2659
<code>\DefaultHeightofSubmit</code>	2623
<code>\DefaultHeightofText</code>	2428
<code>\DefaultHeightofTextMultiline</code>	2426
<code>\DefaultOptionsofCheckBox</code>	2698
<code>\DefaultOptionsofPushButton</code>	2584
<code>\DefaultOptionsofReset</code>	2663
<code>\DefaultOptionsofSubmit</code>	2627
<code>\DefaultOptionsofText</code>	2433
<code>\DefaultWidthofCheckBox</code>	2694
<code>\DefaultWidthofChoiceMenu</code>	2458
<code>\DefaultWidthofReset</code>	2658
<code>\DefaultWidthofSubmit</code>	2622
<code>\DefaultWidthofText</code>	2423
<code>destlabel</code> (hypersetup key)	<u>10</u>
dim commands:	
<code>\dim_to_decimal_in_bp:n</code>	9, 139
<code>\do</code>	2464, 2532
<code>\documentclass</code>	21
<code>draft</code> (hypersetup key)	<u>1181</u>
E	
<code>\edef</code> 2242, 2474, 2510, 2517, 2806, 3278, 3349	
<code>\EdefEscapeName</code>	2530, 2535
<code>\else</code>	2239, 2427, 2481,
2485, 2488, 2505, 2513, 2794, 2799,	
2816, 2822, 2834, 2838, 2855, 2859,	
2863, 2867, 2871, 2875, 2879, 2883,	
2887, 2891, 2908, 2912, 3013, 3014,	
3015, 3017, 3020, 3024, 3028, 3034,	
3051, 3055, 3059, 3065, 3069, 3137,	
3138, 3140, 3143, 3147, 3172, 3173,	
3175, 3178, 3182, 3249, 3316, 3337	
<code>\endcsname</code>	2492, 2790, 2813,
2824, 3304, 3314, 3356, 3357, 3358,	
3359, 3360, 3361, 3362, 3363, 3364	
<code>\endgroup</code>	2243, 2451, 2474, 2520,
2523, 2850, 3278, 3290, 3313, 3349	
<code>\endinput</code>	3250
exp commands:	
<code>\exp_args:Ne</code>	2898
<code>\exp_args:Nne</code>	2002
<code>\exp_args:Nno</code>	1950
<code>\exp_args:NNV</code>	1661, 1692
<code>\exp_args:NnV</code>	392, 399
<code>\exp_args:Nnx</code>	2185
<code>\exp_args:No</code>	
.	421, 425, 2582, 2625, 2661, 2696
<code>\exp_args:NV</code>	375, 382
<code>\exp_args:Nx</code>	1943, 1991, 2234
<code>\exp_not:N</code>	619
<code>\exp_not:n</code>	
.	1096, 1130, 1241, 1301, 1322, 1359,
.	1379, 1402, 1446, 1479, 1504, 1555,
.	1580, 1607, 1656, 1733, 1761, 1788,
.	1808, 2023, 2130, 2158, 2168, 2179
<code>\expandafter</code>	2432, 2466,
2475, 2477, 2478, 2534, 2807, 2823,	
3250, 3274, 3275, 3313, 3314, 3336,	
3338, 3345, 3346, 3356, 3357, 3358,	
3359, 3360, 3361, 3362, 3363, 3364	
<code>\ExplSyntaxOff</code>	
.	10, 2271, 2387, 2414, 2720, 2788,
.	2904, 2968, 3003, 3075, 3126, 3225
<code>\ExplSyntaxOn</code>	
.	8, 11, 2272, 2392, 2415, 2723,
.	2895, 2916, 2969, 3038, 3076, 3162
<code>extension</code> (hypersetup key)	<u>11</u> , <u>1203</u>
F	
<code>\fbox</code>	2378, 2383, 2556
<code>\fi</code>	484, 2228,
2255, 2258, 2267, 2429, 2437, 2438,	
2469, 2487, 2490, 2491, 2497, 2500,	
2509, 2511, 2518, 2521, 2796, 2818,	
2828, 2830, 2836, 2840, 2847, 2848,	
2857, 2861, 2865, 2869, 2873, 2877,	
2881, 2885, 2889, 2893, 2910, 2914,	
3013, 3014, 3015, 3022, 3026, 3030,	
3032, 3034, 3053, 3057, 3061, 3065,	
3072, 3137, 3138, 3145, 3149, 3151,	
3172, 3173, 3180, 3184, 3186, 3251,	
3256, 3306, 3307, 3318, 3322, 3339	
<code>file</code> (hypersetup key)	8, <u>1249</u>

file commands:		
\file_input:n	9	
fileborderstyle (hypersetup key)	11, 863	
filecolor (hypersetup key)	753	
final (hypersetup key)	1181	
\fontencoding	2332, 2344, 2356, 2368	
\fontfamily	2331, 2343, 2355, 2367	
\fontseries	2333, 2345, 2357, 2369	
\fontshape	2334, 2346, 2358, 2370	
fp commands:		
\fp_eval:n	1831, 1859, 1875	
G		
\GetDocumentProperties	2	
\global	2397, 2786	
group commands:		
\group_begin:		
. 331, 439, 453, 476, 520, 571, 632,		
725, 976, 993, 2109, 2330, 2342,		
2354, 2366, 2581, 2624, 2660, 2693		
\group_end:		
. 338, 443, 464, 507, 559, 613, 661,		
735, 996, 999, 2116, 2337, 2349,		
2361, 2373, 2617, 2653, 2686, 2718		
H		
\hbox	3260	
hbox commands:		
\hbox_overlap_right:n	990	
\hbox_set:Nn	2599, 2638, 2673	
\hbox_set:Nw	977	
\hbox_set_end:	986	
\hbox_set_to_wd:Nnn	1857	
hidefile (hypersetup key)	1133	
hidelink (hypersetup key)	1133	
hidelinks (hypersetup key)	1133	
hidemenu (hypersetup key)	1133	
hiderun (hypersetup key)	1133	
hideurl (hypersetup key)	1133	
hook commands:		
\hook_gput_code:nnn		
. 191, 719, 729, 969, 980		
\hook_new:n	182, 316	
\hook_use:n	333, 440, 454	
\hss	3265	
Hy internal commands:		
\Hy_hyescapeform	2807, 2820, 2826	
\Hy_hypSectionAnchor	3255, 3258	
HyField internal commands:		
\HyField_hypAddToFields	2234, 2261	
hyp commands:		
\l_hyp_annot_colorfile_bool	287	
\l_hyp_annot_colorlink_bool	287	
\l_hyp_annot_colormenu_bool	287	
\l_hyp_annot_colorrund_bool	287	
\l_hyp_annot_colorurl_bool	287	
\l_hyp_annot_ocgcolorfile_bool	291	
\l_hyp_annot_ocgcolorlink_bool	291	
\l_hyp_annot_ocgcolormenu_bool	291	
\l_hyp_annot_ocgcolorrund_bool	291	
\l_hyp_annot_ocgcolorurl_bool	291	
hyp internal commands:		
\g__hyp_AcroForm_CoFields_prop		
. 2195, 2216, 2290, 2292, 2307		
\g__hyp_AcroForm_Fields_prop		
. 2196, 2209, 2285		
\l__hyp_annot_GoTo_bool	436, 450, 461	
\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	295	
\l__hyp_annot_GoToR_bool	518	
\l__hyp_annot_Launch_bool	569	
\l__hyp_annot_Named_bool	630	
\c__hyp_annot_types_seq		
. 250, 287, 291, 753, 1048, 1144		
\l__hyp_annot_URI_bool	474	
\g__hyp_bordercolormodel_str		
. 286, 778, 794		
\l__hyp_CheckmarkOff_tl		
. 2275, 2276, 2932		
\l__hyp_CheckmarkYes_tl		
. 2273, 2274, 2931		
__hyp_clist_display:n	104, 111, 120	
__hyp_color_export:nnN		
. 31, 666, 666, 682, 794		
__hyp_color_select:n	32, 683, 683, 699	
__hyp_color_select:nN	32	
__hyp_color_set:nN	32	
__hyp_color_set:nn		
. 32, 187, 189, 700, 700, 716, 758		
__hyp_colormodel_export:nnnN	31, 666	
__hyp_colormodel_export:wnnN		
. 670, 677		
__hyp_colormodel_select_aux:wn		
. 683, 687, 694		
__hyp_colormodel_set_aux:nwn		
. 700, 704, 711		
\c__hyp_colorscheme_daleif_prop	3397	
\c__hyp_colorscheme_henryford_-		
prop	3488	
\c__hyp_colorscheme_julian_prop	3412	
\c__hyp_colorscheme_original_-		
prop	3382	
\c__hyp_colorscheme_phetype_prop		
. 3473		

\c_hyp_colorscheme_szabolcsA-prop	3442	\l_hyp_text_enc_dest_print_tl	272, 530
\c_hyp_colorscheme_szabolcsB-prop	3457	\l_hyp_text_enc_dest_tl	24, 272, 371, 422
\c_hyp_colorscheme_tivv_prop	3427	\l_hyp_text_enc_file_print_tl	272, 525, 574
\l_hyp_dest_box	24, 300, 377, 378, 379, 1857, 1873, 1880	\l_hyp_text_enc_info_print_tl	272, 346
\l_hyp_dest_name_tmpa_tl	243, 410, 411, 417, 421, 423, 426, 531, 539	\l_hyp_text_enc_para_print_tl	272, 579
\l_hyp_dest_pdfremotestartview_tl	269, 536, 1664, 1668	\l_hyp_text_enc_uri_print_tl	272, 479, 1260
\g_hyp_dest_pdfstartpage_tl	269, 1675, 1677, 1684, 1702, 1709	_hyp_text_pdfstring:nnN	329, 329, 341, 343, 346, 369, 420, 477, 523, 528, 572, 577, 1260
\g_hyp_dest_pdfstartview_tl	269, 1677, 1684, 1695, 1699, 1702, 1709	_hyp_text_pdfstring_info:nN	344, 344, 1919, 1937, 1947
\l_hyp_dest_pdfview_tl	285, 393, 400, 1826, 1829, 1836, 1839, 1840, 1841, 1842, 1843, 1844, 1849, 1853, 1889	_hyp_text_purify:nN	317, 317, 334
\c_hyp_dest_startview_regex	301, 1662, 1693	_hyp_text_string_from_unicode:nN	325, 325, 336
\c_hyp_dest_undefined_tl	249, 416, 417	\g_hyp_text_tmpa_str	248, 337, 339
_hyp_destination:nn	24, 364, 364, 393, 400	\l_hyp_text_tmpa_str	247, 334, 335, 336, 337
\l_hyp_filename_tmpa_tl	243, 521, 522, 524, 526, 527, 575, 576, 592	\l_hyp_tmpa_box	238, 977, 990, 995
_hyp_link_goto_begin:nw	407, 441, 455	\l_hyp_tmpa_int	238, 1662, 1664, 1693, 1695, 1816, 1817, 1821, 1823, 1847, 1854, 1855, 1856, 1865, 1867, 1877, 1882, 2294, 2296, 2302
_hyp_link_goto_end:	429, 442, 463	\l_hyp_tmpa_str	238, 1919, 1920, 1922, 1937, 1938, 1940, 1947, 1948, 1952
\c_hyp_map_annot_hyp_prop	250	\l_hyp_tmpa_tl	238, 372, 376, 384, 794, 798, 1260, 1261, 1266, 1660, 1662, 1691, 1693, 1823, 1824, 1831
\c_hyp_map_hyp_annot_prop	250, 717, 781, 817, 844, 852, 863, 890, 898, 967, 1071, 1107, 1117, 1249	\l_hyp_uri_tmpa_tl	243, 480, 481
_hyp_ocg_init:	909, 909, 965, 975	hyp/annot/file (color name)	286
_hyp_PageLabels_gpush:	348, 348, 361	hyp/annot/link (color name)	286
\l_hyp_para_tmpa_tl	246, 580, 583, 592	hyp/annot/menu (color name)	286
\l_hyp_para_tmpa_tl_uuu\l_hyp_text_tmpa_str	243	hyp/annot/run (color name)	286
\l_hyp_RadioYes_tl	2525, 2526	hyp/annot/url (color name)	286
_hyp_ref_check:nn	210, 220, 232, 2402	hyp/text/pdfstring	316
_hyp_ref_if_exist:nn	210, 218, 219	\hypercalcbp	9, 15, 139
_hyp_ref_if_exist:nnTF	222, 2406	\HyperDestNameFilter	370, 421
_hyp_ref_label:nn	210, 211, 217, 2401	\hypersetup	1, 2, 6, 8, 10, 15, 59, 91, 162
_hyp_ref_value:nn	210, 233, 237, 2410	\hypersetup keys:	
_hyp_setup_info_date_key:nn	1964, 1996, 1997	allcolor	753
_hyp_setup_info_key:nn	1908, 1957, 1958, 1959, 1960, 1962, 1963	bookmarkstype	10
_hyp_store_metadata:nn	176, 181, 1905, 1925, 1943, 1977, 1991, 2012, 2084	bordercolormodel	10, 774
_hyp_text_cleanup:N	321, 321, 335	colorfile	753
		colorlink	753
		colorlinks	739
		colormenu	753
		colorrun	753

\keys_set:nn
 . 165, 193, 1163, 1170, 2003, 2030,
 2033, 2034, 2035, 2036, 2110, 2185
 \keys_set_known:nn 2188
 \kvsetkeys 164, 2281

L

\label 6, 11
 \LayoutCheckField 2706
 \LayoutChoiceField 2501
 \LayoutPushButtonField 2594, 2611
 \LayoutTextField 2439
 \leavevmode 2440, 2538, 3259
 legacy commands:

 \legacy_if:nTF 150, 202, 206,
 356, 2319, 2590, 2631, 2669, 2702, 2959
 \let 146,
 2198, 2199, 2201, 2202, 2229, 2280,
 2388, 2389, 2390, 2421, 2422, 2456,
 2457, 2499, 2786, 2793, 2795, 2800,
 2817, 2851, 3268, 3282, 3285, 3294,
 3305, 3312, 3315, 3317, 3325, 3333
 link (hypersetup key) 8, [1249](#)
 linkborder (hypersetup key) 11
 linkborderstyle (hypersetup key) . 11, [863](#)
 linkcolor (hypersetup key) [753](#)
 linkfileprefix (hypersetup key) ... [1203](#)
 linktoc (hypersetup key) [1203](#)
 linktocpage (hypersetup key) [1203](#)
 localanchorname (hypersetup key) . [1203](#)

M

\MakeButtonField
 2599, 2614, 2638, 2673, 2684
 \MakeChoiceField 2574
 \MakeFieldObject 2189,
 2328, 2340, 2352, 2364, 2376, 2381
 \MakeRadioField 2556
 \MakeTextField 2448
 \mbox 987
 menu (hypersetup key) 8, [1249](#)
 menuborder (hypersetup key) 11
 menuborderstyle (hypersetup key) . 11, [863](#)
 menucolor (hypersetup key) [753](#)
 mode commands:
 \mode_if_horizontal:TF 366, 388
 \mode_leave_vertical:
 409, 493, 546, 599,
 635, 2568, 2596, 2613, 2635, 2665, 2708
 msg commands:
 \msg_error:nn 201, 2610, 2683
 \msg_line_context: 69
 \g_msg_module_name_prop 12

\msg_new:nnn 37, 44, 49, 53,
 57, 64, 71, 78, 85, 92, 98, 105, 114, 124
 \msg_new:nnnn 13, 23
 \msg_warning:nn 152, 1177
 \msg_warning:nnn 161, 413, 658
 \msg_warning:nnnn
 1009, 1056, 1299, 1343,
 1377, 1400, 1462, 1491, 1525, 1540,
 1553, 1566, 1591, 1624, 1640, 1667,
 1698, 1731, 1745, 1772, 1806, 1888
 \msg_warning:nnnnn 172,
 1093, 1127, 1238, 1319, 1356, 1443,
 1476, 1501, 1577, 1604, 1653, 1758,
 1785, 2020, 2127, 2146, 2165, 2176

N

naturalnames (hypersetup key) [1203](#)
 nesting (hypersetup key) 11
 \newcommand 142, 3233, 3235, 3237
 \newcount 2393, 2564
 \NewDocumentCommand 2189
 \newlength 3252
 \noexpand ... 2475, 2476, 2477, 2478, 2804

O

ocgcolorfile (hypersetup key) [1003](#)
 ocgcolorlink (hypersetup key) [1003](#)
 ocgcolorlinks (hypersetup key) [1003](#)
 ocgcolormenu (hypersetup key) [1003](#)
 ocgcolorrun (hypersetup key) [1003](#)
 ocgcolorurl (hypersetup key) [1003](#)

P

page/Trans internal commands:
 l_hyp_page/Trans 314
 pageanchor (hypersetup key) [1203](#)
 \paperwidth 3
 \PassOptionsToPackage ... 208, 1195, 1200
 pdf commands:
 \pdf_bdcobject:nn 989, 992
 \pdf_destination:nn 19, 135, 383
 \pdf_destination:nnnn 376
 \pdf_emc: 991, 997
 \pdf_link_user:nnn 2571
 \pdf_name_from_unicode_e:n 637
 \pdf_object_new:nn
 . 911, 912, 913, 914, 2726, 2727, 2728
 \pdf_object_ref:n
 917, 919, 941, 942, 945,
 949, 954, 959, 964, 2312, 2314, 2784
 \pdf_object_ref_last: 2115
 \pdf_object_unnamed_write:nn . 2111
 \pdf_object_write:nn
 . 915, 921, 931, 943, 2729, 2771, 2778

\pdf_pageobject_ref:n	1684, 1709, 2408	pdfmanagement commands:	
\pdf_string_from_unicode:nnN	.. 327	\pdfmanagement_add:nn 1390
\pdf_version:		\pdfmanagement_add:nnn
...	1347, 1466, 1495, 1529, 1544,	350, 941, 942,
1570, 1595, 1628, 1644, 1749, 1776		964, 1266, 1291, 1307, 1311, 1331,	
\pdf_version_compare:NnTF	1339, 1368, 1423, 1433, 1457, 1471,	
.....	1003, 1050,	1483, 1488, 1513, 1521, 1536, 1562,	
1337, 1428, 1450, 1486, 1519, 1534,		1587, 1619, 1636, 1682, 1707, 1721,	
1560, 1585, 1611, 1634, 1739, 1766		1741, 1768, 1797, 1903, 1922, 1940,	
\pdf_version_compare_p:Nn 585	1951, 1975, 1989, 2007, 2096, 2115,	
\pdf_version_major: ..	149, 1012, 1060	2287, 2304, 2311, 2313, 2315, 2321	
\pdf_version_minor: ..	148, 1012, 1060	\pdfmanagement_if_active_p: ..	200
pdfannot commands:		\pdfmanagement_remove:nn
\pdfannot_box:nnnn	1263, 1287, 1295, 1315, 1327,
..	2443, 2541, 2600, 2639, 2674, 2711	1352, 1364, 1372, 1386, 1395, 1439,	
\pdfannot_box_ref_last: ..	2236, 2252	1454, 1473, 1510, 1549, 1574, 1600,	
\pdfannot_dict_put:nnn	1615, 1649, 1679, 1704, 1717, 1726,	
.....	494, 547, 600,	1754, 1781, 1793, 1802, 1900, 1916,	
643, 795, 830, 854, 876, 900, 1078, 1109		1934, 1972, 1986, 2093, 2106, 2325	
\pdfannot_dict_remove:nn	\pdfmanagement_show:n 2288
..	789, 825, 846, 871, 892, 1086, 1119	pdfmeta commands:	
\pdfannot_link:nnn	.. 495, 548, 601, 644	\pdfmeta_standard_verify:nnTF	...
\pdfannot_link_goto_begin:nw	.. 426	633, 2587, 2666
\pdfannot_link_goto_end:	431	\pdfmeta_standard_verify:nTF	...
\pdfannot_link_margin:n 6, 1381	2317, 2900
\c_pdfannot_link_types_seq 295	pdfmoddate (hypersetup key) 11, 1964
pdfauthor (hypersetup key) 1908	pdfproducer (hypersetup key) 1908
\pdfbookmark 3238	pdfremotestartview (hypersetup key)	... 9
pdfborder (hypersetup key) 11	pdfstartview (hypersetup key) 9
pdfborderstyle (hypersetup key)	.. 11, 863	\pdfstringdef 6, 7, 15, 21, 2849
pdfcreationdate (hypersetup key)	11, 1964	pdfsubject (hypersetup key) 1908
pdfcreator (hypersetup key) 1908	pdftitle (hypersetup key) 1908
\pdfdest 9	pdftrapped (hypersetup key) 1998
pdfdict commands:		pdfversion (hypersetup key) 1171
\pdfdict_new:n	314, 468, 512, 563, 624	pdfview (hypersetup key) 9, 1812
\pdfdict_put:nnn	pdfxform commands:	
..	315, 469, 470, 481, 483, 488, 513,	\pdfxform_new:nnn 2191
514, 527, 535, 539, 544, 564, 565,		\pdfxform_ref:n
576, 591, 597, 625, 626, 636, 641,		..	2551, 2552, 2646, 2647, 2931, 2932
1407, 1408, 1412, 1413, 2124, 2134,		\phantom 2348, 2372
2138, 2141, 2143, 2162, 2171, 2173		\phantomsection 3321
\pdfdict_remove:nn	... 588, 1417, 1418	plainpages (hypersetup key) 1203
\pdfdict_use:n 494,	prg commands:	
499, 547, 552, 600, 606, 643, 648, 2113		\prg_generate_conditional_-	
pdfencoding (hypersetup key) 1171	variant:Nnn 219
\pdfescapestring 2793	\prg_new_eq_conditional:NNn	... 218
PDFForm internal commands:		prop commands:	
\PDFForm_hypName	\prop_const_from_keyval:Nn
.....	2832, 2835, 2839, 2842	252, 260, 3382, 3397,
pdfinfo (hypersetup key) 2026	3412, 3427, 3442, 3457, 3473, 3488	
pdfkeywords (hypersetup key) 1908	\prop_gput:Nnn 12, 2209, 2216
pdflang (hypersetup key) 11, 1894	\prop_if_empty:Nn 2290
pdflinkmargin (hypersetup key) 11	\prop_item:Nn 2307

<code>\prop_map_inline:Nn</code>	717, 781, 817, 844, 852, 863, 890, 898, 967, 1071, 1107, 1117, 1161, 1249, 2285, 2292
<code>\prop_new:N</code>	2195, 2196
<code>\protect</code>	224, 3254
<code>\providecommand</code>	140, 141, 2223, 2226, 2453, 2789
<code>\ProvidesFile</code>	3
R	
<code>\ReadBookmarks</code>	3243
ref commands:	
<code>\ref_if_exist:nn</code>	218
<code>\ref_label:nn</code>	214
<code>\ref_value:nn</code>	18, 235
regex commands:	
<code>\regex_const:Nn</code>	301
<code>\regex_extract_once:NnNTF</code>	1662, 1693
<code>\relax</code>	2456, 2457, 2498, 2786, 2833, 2837, 2907, 2911, 3014, 3015, 3023, 3027, 3054, 3058, 3138, 3146, 3173, 3181, 3305, 3314
<code>\RemoveFromHook</code>	1271
<code>\RenewDocumentCommand</code>	155
<code>\RequirePackage</code>	7, 19, 204, 3241
<code>run</code> (hypersetup key)	8, 1249
<code>runborder</code> (hypersetup key)	11
<code>runborderstyle</code> (hypersetup key)	11, 863
<code>runcolor</code> (hypersetup key)	753
S	
<code>\selectfont</code>	2335, 2347, 2359, 2371
seq commands:	
<code>\seq_const_from_clist:Nn</code>	250
<code>\seq_count:N</code>	1821, 1847
<code>\seq_get_right:NN</code>	1823
<code>\seq_item:Nn</code>	1664, 1695, 1817, 1865, 1867, 1877, 1882
<code>\seq_map_inline:Nn</code>	287, 291, 295, 753, 1048, 1144, 2302
<code>\seq_new:N</code>	240
<code>\seq_put_right:Nn</code>	1854, 1855, 1856, 2294
<code>\seq_set_split:Nnn</code>	1816
<code>\seq_sort:Nn</code>	2296
<code>\setlength</code>	3271, 3342
<code>\setpdflinkmargin</code>	6
<code>\settowidth</code>	2468
<code>\show</code>	2529
sort commands:	
<code>\sort_return_same:</code>	2300
<code>\sort_return_swapped:</code>	2299
<code>\space</code>	228, 2537, 3016, 3033, 3034, 3064, 3065, 3139, 3171, 3174
<code>\spacefactor</code>	366, 388
<code>\stockwidth</code>	3
str commands:	
<code>\c_colon_str</code>	619
<code>\c_hash_str</code>	490
<code>\c_percent_str</code>	491
<code>\str_case:nnTF</code>	3196
<code>\str_case_e:nnTF</code>	1817
<code>\str_gset:Nn</code>	778
<code>\str_gset_eq:NN</code>	337
<code>\str_head:n</code>	2009, 2014
<code>\str_if_eq:nnTF</code>	373, 1920, 1938, 1948, 2898
<code>\str_if_eq_p:nn</code>	583
<code>\str_lowercase:n</code>	1817, 1943, 1991, 2010, 2015
<code>\str_new:N</code>	242, 247, 248, 286
<code>\str_remove_once:Nn</code>	487, 543, 596, 640
<code>\str_set:Nn</code>	319
<code>\str_set_convert:Nnnn</code>	6
<code>\str_set_eq:NN</code>	339
<code>\str_tail:n</code>	2010, 2015
<code>\str_uppercase:n</code>	2003, 2009, 2014
<code>\string</code>	2223, 2226, 2247, 2265, 2809, 2813, 2824
<code>\subpdfbookmark</code>	3234
sys commands:	
<code>\c_sys_backend_str</code>	3196
T	
TeX and L ^A T _E X 2 _ε commands:	
<code>\@BOOKMARK</code>	3240
<code>\@CheckBox</code>	2689
<code>\@ChoiceMenu</code>	2454
<code>\@Form</code>	2278
<code>\@PushButton</code>	2579
<code>\@Reset</code>	2656
<code>\@Submit</code>	2620
<code>\@TextField</code>	2419
<code>\@bookmarksopenlevel</code>	1282
<code>\@bsphack</code>	213
<code>\@chapter</code>	3294, 3295
<code>\@currDisplay</code>	2468, 2537
<code>\@currValue</code>	2535, 2551
<code>\@currentHref</code>	3275, 3288, 3299, 3329, 3346
<code>\@curopt</code>	2453, 2463, 2466, 2531, 2534
<code>\@empty</code>	2422, 2438, 2854, 2858, 2862, 2866, 2870, 2874, 2878, 2882, 2886, 2890, 3034, 3065, 3066
<code>\@endForm</code>	2388
<code>\@esphack</code>	215
<code>\@firstofone</code>	2229, 2795, 2800, 2817
<code>\@firstoftwo</code>	3336

<code>\@for</code>	2464, 2532	<code>\Fld@default</code>
<code>\@gobble</code>	2280	...	2421, 2438, 2456, 2499, 2529,
<code>\@gobbletwo</code>	3285		2530, 2692, 3066, 3070, 3071, 3118
<code>\@hyper@launch</code>	30, 619	<code>\Fld@flags</code>	2924,
<code>\@ifundefined</code>	2280, 2809		2977, 3010, 3046, 3085, 3134, 3169
<code>\@latex@warning</code>	225	<code>\Fld@format@code</code>	2858, 2860
<code>\@mainaux</code>	2222, 2225, 2245, 2264	<code>\Fld@height</code>
<code>\@mkboth</code>	3285		2424, 2445, 2448, 2459, 2517, 2543,
<code>\@nil</code>	2807, 2820, 2826		2556, 2574, 2623, 2659, 2695, 2713
<code>\@part</code>	3312, 3319	<code>\Fld@keystroke@code</code>	2854, 2856
<code>\@pdfauthor</code>	16	<code>\Fld@listcount</code>	2528, 2536, 2557, 2564
<code>\@pdfborder</code>	140	<code>\Fld@mappingname</code>	2837, 2839, 2911, 2913
<code>\@pdfborderstyle</code>	141	<code>\Fld@maxlen</code>	3121, 3123
<code>\@savsf</code>	366, 388	<code>\Fld@menulength</code>	2461, 2467, 2514
<code>\@schapter</code>	3282, 3283	<code>\Fld@name</code>
<code>\@secondoftwo</code>	3338	..	2420, 2455, 2580, 2691, 2832, 2906
<code>\@sect</code>	3333, 3334	<code>\Fld@onblur@code</code>	2874, 2876
<code>\@spart</code>	3325, 3326	<code>\Fld@onclick@code</code>	2999
<code>\@ssect</code>	3268, 3269	<code>\Fld@onenter@code</code>	2886, 2888
<code>\@tempdima</code>	2462, 2469, 2471, 2508,	<code>\Fld@onexit@code</code>	2890, 2892
	2509, 2510, 2514, 2515, 2516, 2517	<code>\Fld@onfocus@code</code>	2870, 2872
<code>\@tempdimb</code>	2468, 2469	<code>\Fld@onmousedown@code</code>	2878, 2880
<code>\@typeset@protect</code>	3254	<code>\Fld@onmouseup@code</code>	2882, 2884
<code>\c@secnumdepth</code>	3302, 3320, 3335	<code>\Fld@pageobjref</code>
<code>\calc@bm@number</code>	3246	2390, 2404, 2789, 2921,
<code>\check@bm@number</code>	3245		2974, 3007, 3043, 3082, 3131, 3166
<code>\Fld@additionalactions</code>	2896, 2966,	<code>\Fld@radiosymbol</code>	3062
	3000, 3036, 3073, 3120, 3160, 3189	<code>\Fld@rotation</code>
<code>\Fld@align</code>	2925, 3011, 3086		2936,
<code>\Fld@altname</code>	2833, 2835, 2907, 2909		2938, 2982, 2989, 2991, 3013, 3019,
<code>\Fld@annotflags</code>	2920,		3021, 3050, 3052, 3090, 3099, 3101,
	2973, 3006, 3042, 3081, 3130, 3165		3137, 3142, 3144, 3172, 3177, 3179
<code>\Fld@annotnames</code>	2905, 2922,	<code>\Fld@submitflags</code>	3158
	2975, 3008, 3044, 3083, 3132, 3167	<code>\Fld@validate@code</code>	2862, 2864
<code>\Fld@bcolor</code>	2944, 2946, 3027,	<code>\Fld@value</code>
	3029, 3058, 3060, 3094, 3107, 3109	..	2422, 2438, 2457, 2498, 2499, 3119
<code>\fld@bcolor</code>	3015	<code>\Fld@width</code>
<code>\Fld@bordercolor</code>	2940,		2423,
	2942, 2984, 2993, 2995, 3014, 3023,		2437, 2444, 2448, 2458, 2497, 2508,
	3025, 3054, 3056, 3092, 3103, 3105,		2510, 2542, 2556, 2574, 2592, 2622,
	3138, 3146, 3148, 3173, 3181, 3183		2633, 2658, 2670, 2694, 2704, 2712
<code>\Fld@borderstyle</code>	2926,	<code>\Fld@X@additionalactions</code>
	2979, 3012, 3048, 3087, 3136, 3187	2853, 2898, 2901
<code>\Fld@borderwidth</code>	2515, 2516, 2926,	<code>\Form@action</code>	3156
	2979, 3012, 3048, 3087, 3136, 3187	<code>\G@refundefinedtrue</code>	224
<code>\Fld@calculate@code</code>	2238, 2866, 2868	<code>\H@old@part</code>	3312, 3323
<code>\Fld@calculate@sortkey</code>	2249	<code>\H@old@schapter</code>	3282, 3291
<code>\Fld@cbsymbol</code>	2948	<code>\H@old@sect</code>	3333, 3344, 3353
<code>\Fld@charsize</code>	<code>\H@old@spart</code>	3325, 3331
..	2514, 2952, 3033, 3064, 3115, 3171	<code>\H@old@ssect</code>	3268, 3273
<code>\Fld@checkequals</code>	2466, 2534	<code>\Hy@abspage</code>	360
<code>\Fld@choices</code>	3035	<code>\Hy@activeanchorfalse</code>	405
<code>\Fld@color</code>	2953, 2955, 3034, 3065, 3116	<code>\Hy@activeanchortrue</code>	398
		<code>\Hy@AtBeginDocument</code>	2203, 2219
		<code>\Hy@bookmarksnumberedfalse</code>	1275
		<code>\Hy@bookmarksnumberedtrue</code>	1276

\Hy@bookmarksopenfalse	1279	\Hy@secnum@part	3315, 3317, 3320
\Hy@bookmarksopentru	1280	\Hy@SectionAnchorHref	3253, 3275, 3346
\Hy@bookmarkstype	1283	\Hy@SectionHShift	3252, 3261, 3271, 3342
\Hy@chapapp	3286, 3297	\Hy@StepCount	2467, 2536
\Hy@CleanupFile	522	\Hy@temp	2242, 2257, 2806, 2807
\Hy@colorlink	26	\Hy@unicodetrue	2846
\Hy@currentbookmarklevel	3227	\Hy@unicodetrue	145
\Hy@DisableOption	147, 154	\Hy@VerboseAnchor	368
\Hy@drafttrue	1194	\Hy@verbosefalse	1185
\Hy@escapeform	2442, 2540, 2570, 2598, 2637, 2672, 2710, 2791, 2798, 2801	\Hy@VerboseLinkStart	438, 452
\Hy@escapestring	2793, 2795, 2800, 2803, 2804, 2807, 2808, 2817, 2820, 2856, 2860, 2864, 2868, 2872, 2876, 2880, 2884, 2888, 2892, 2948, 2999, 3062, 3118, 3119, 3156	\Hy@VerboseLinkStop	444, 465, 505, 557, 611, 654
\Hy@finaltrue	1199	\Hy@verbosetrue	1184
\Hy@FormObjects	2284, 2724, 2786	\Hy@VersionChecked	136
\Hy@gtemp	2849, 2851	\Hy@WrapperDef	364
\Hy@href@nextactionraw	485, 487, 488, 541, 543, 544, 594, 596, 597, 638, 640, 641	\Hy@xspace@end	441, 504, 556, 610, 653
\Hy@href@page	536	\HyAnn@AbsPageLabel	2389, 2395, 2441, 2539, 2569, 2597, 2636, 2671, 2709
\Hy@hypertextnamesfalse	1209	\HyAnn@Count	2393, 2394, 2397, 2398, 2399, 2400, 2401, 2402, 2406, 2410
\Hy@hypertextnametrue	1208	\HyField@AddToFields	2232, 2449, 2558, 2575, 2606, 2650, 2679, 2716
\Hy@linkfileprefix	1211	\HyField@afields	2198
\Hy@linktoc	1234	\HyField@AfterAuxOpen	2203, 2229, 2262
\Hy@localanchornametrue	1213	\HyField@AuxAddToCoFields	2202, 2214, 2226, 2247
\Hy@MakeCurrentHrefAuto	3270, 3286, 3297, 3327, 3341	\HyField@AuxAddToFields	2201, 2207, 2223, 2265
\Hy@MakeRemoteAction	532	\HyField@cofields	2199
\Hy@naturalnamesfalse	1218	\HyField@FlagsCheckBox	2701
\Hy@naturalnametrue	1217	\HyField@FlagsChoice	2507
\Hy@next	3296, 3305, 3308	\HyField@FlagsPushButton	2589, 2629, 2668
\Hy@numberline	143, 3228	\HyField@FlagsRadioButton	2503
\Hy@org@chapter	3294, 3309	\HyField@FlagsSubmit	2630
\Hy@OutlineName	3244	\HyField@FlagsText	2436
\Hy@OutlineRerunCheck	3242	\HyField@PDFChoices	2567
\Hy@pageanchorfalse	1222	\HyField@SetKeys	2432, 2473, 2476, 2495, 2582, 2625, 2661, 2696
\Hy@pageanchortrue	1221	\HyPat@ObjRef	3200, 3207, 3214, 3221
\Hy@pdfmajorversion	149	\hyper@anchor	390
\Hy@pdfminorversion	148	\hyper@anchorend	390, 3263, 3288, 3299, 3329
\Hy@pdfstringtrue	22, 332	\hyper@anchorstart	390, 3263, 3288, 3299, 3329
\Hy@pdfversion	2844	\hyper@link	26, 26, 434
\Hy@plainpagesfalse	1226	\hyper@linkend	26, 26, 459
\Hy@plainpagetrue	1225	\hyper@linkfile	516
\Hy@pstringdef	342, 2824	\hyper@linklaunch	29, 567, 622
\Hy@PutCatalog	348	\hyper@linknamed	30, 157, 628
\Hy@raisedlink	3262, 3287, 3298, 3328	\hyper@linkstart	26, 26, 448
\Hy@RestoreLastskip	387	\hyper@linkurl	472
\Hy@safe@activetrue	492, 2241		
\Hy@SaveLastskip	367		

\HyPL@Labels	350, 360	\textsf	2378, 2383
\HyPL@StorePageLabel	348	\the	360,
\HyPsd@LoadUnicode	144		2398, 2399, 2400, 2401, 2402, 2406,
\HyPsd@pdfencoding	146		2410, 2510, 2514, 2517, 3279, 3350
\HyPsd@pdfencoding@unicode	146	\thepage	228
\HyPsd@SanitizeForOutFile	3231	\ThisShouldNotHappen	2811
\if@filesw	2221, 2244, 2263	\title	2
\ifFld@combo	2482, 2512	tl commands:	
\ifFld@hidden	2437, 2497	\c_space_tl	143, 918,
\ifFld@multiline	2425		949, 954, 959, 2551, 2552, 2559,
\ifFld@popdown	2483		2735, 2737, 2747, 2749, 2751, 2754,
\ifFld@radio	2479, 2502		2756, 2952, 2955, 3087, 3115, 3116
\ifHy@href@ismap	482	\tl_const:Nn	249
\ifHy@implicit	3248	\tl_gput_right:Nn	360
\ifHy@pdfescapeform	2792, 2802	\tl_gset:Nn	1675, 1695, 1699
\ifHy@unicode	2845	\tl_if_blank:NnTF	
\kv@set@family@handler	160		533, 1914, 1932, 1970, 1984, 2091, 2104
\ltx@empty	2198,	\tl_if_empty:NnTF	411, 485, 541,
	2199, 2238, 2388, 2389, 2390, 2421		594, 638, 1261, 2940, 2944, 2953, 3116
\ltx@gobble	2201	\tl_if_empty:NnTF	787, 823,
\ltx@gobbletwo	2202		842, 869, 888, 1452, 1508, 1613, 1898
\ltx@ifUndefined	3293, 3303	\tl_if_empty_p:N	1677, 1702
\ltx@one	2397	\tl_if_eq:NnTF	1824
\ltx@ReturnAfterFi	2825	\tl_if_exist:NnTF	2993, 3103, 3107
\ltx@zero	2394	\tl_if_exist_p:N	2984, 3092, 3094
\m@ne	3302, 3317	\tl_if_head_eq_charcode:NnTF	
\OBJ@OCG@view	142		668, 685, 702
\p@	2471	\tl_new:N	239, 243, 244, 245,
\pdf@ifdraftmode	2282		246, 269, 270, 271, 272, 273, 274,
\PDF@SetupDoc	137		275, 276, 277, 285, 2273, 2275, 2525
\pdf@strcmp	2298	\tl_set:Nn	178, 279,
\PDFForm@Check	2710, 2715, 2917		280, 281, 282, 283, 284, 521, 1660,
\PDFForm@List	2570, 2573, 3004		1664, 1668, 1691, 1826, 1829, 1836,
\PDFForm@Name			1839, 1840, 1841, 1842, 1843, 1844,
	2435, 2496, 2586, 2700, 2831		1849, 1853, 1889, 2274, 2276, 2526
\PDFForm@Push	2598, 2604, 2970	\tl_set_eq:NN	417
\PDFForm@Radio	2540, 2546, 3039	\tl_to_str:n	
\PDFForm@Reset	2672, 2678, 3163		19, 20, 21, 32, 227, 1975, 1989
\PDFForm@Submit	2637, 2644, 3128	token commands:	
\PDFForm@Text	2442, 2447, 3078	\token_to_str:N	28
\protected@edef	410	\toks	3274, 3279, 3345, 3350
\strip@pt	2952, 3033, 3064, 3115, 3171		
\toks@	3273, 3279, 3344, 3350	U	
\tw@	3274, 3279, 3345, 3350	unicode (hypersetup key)	1171
\XR@ext	1205	url (hypersetup key)	8, 1249
\z@	2462, 3013, 3019,	\url	4
	3050, 3137, 3142, 3172, 3177, 3315	urlborder (hypersetup key)	11
\zref@labelbylist	2399	urlborderstyle (hypersetup key)	11, 863
\zref@labelbyprops	2398	urlcolor (hypersetup key)	753
\zref@refused	2400	\Urlfont	4
\texorpdfstring	22	use commands:	
text commands:		\use:n	617
\text_purify:n	6, 319	V	
\textcolor	2280, 2378, 2383	verbose (hypersetup key)	1181

	W	X
\write	2222, 2225, 2245, 2264	\x 2474, 2494, 3278, 3280, 3349, 3351