

The `hyperref-generic` package

A generic driver for `hyperref`

The L^AT_EX3 Project*

Released XXXX-XX-XX

This package generates a generic driver for `hyperref` meant to be used with the new LaTeX PDF management code. It is loaded automatically if the PDF management code is active. The name of the driver will probably change.

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} %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 rationals 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 too. This will take some time. Until then it is recommended to follow the following rules

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

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

- If the package `color` is used it is currently recommended to define colors after `hyperref`.
- 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 for two reasons:

- External package like `hyperxmp` want access to them too.
- `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 store in a Metadata container, and can be retrieved with `\GetDocumentMetadata`.

```
\edef\my@pdfauthor{\GetDocumentMetadata{hyp/pdfauthor}}
```

If the key hasn't be set, the result is empty.

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

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

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

4 Dates

XMP: yyyy-mm-ddThh:mm:ss+tt:tt, PDF: D:YYYYMMDDhhmmss+01'00'

5 PDF page size (mediabox)

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

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

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

The problem is to know which value to use (with the memoir class e.g. `\stockwidth` should be used instead of `\paperwidth`), and detecting this not a `hyperref` task. Instead the packages which change these values should also set the PDF page size. Also there are too many actors here: `color/graphicsx`, `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 `graphicsx` 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 `cmymk`. `rgb` is the default. It does *not* change the model of text colors. Be aware that while the PDF format allows `cmymk` (4 numbers) in the `/C` key of an annotation, this is often ignored by pdf viewers and the colors can be wrong.
- The boolean keys `url`, `link`, `run`, `menu`, `file` allow to deactivate locally the link types.

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_colorruncolor_bool,  
\l_hyp_annot_colormenu_bool,  
\l_hyp_annot_ocgcolorlink_bool,  
\l_hyp_annot_ocgcolorurl_bool,  
\l_hyp_annot_ocgcolorfile_bool,  
\l_hyp_annot_ocgcolorruncolor_bool,  
\l_hyp_annot_ocgcolormenu_bool,
```

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

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

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

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

The border colors aren't saved in color names currently, but if the need would arise it would be 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 reenable the border if wanted. Internally `colorlinks` & friends will no longer define/undefine `\Hy@colorlink`, but instead use the hooks provided by the `l3pdfannot` package.

Color keys accept the following input syntax:

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

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

frenchlinks The option `frenchlinks` does nothing at all.

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

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

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

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

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

- `FitR`. If no argument (separated by spaces) follows then `pdfview` will use with `pdf`-`tex` and `luatex` the automatic calculation of the coordinates from the encompassing

box. With dvips and (x)dvipdfmx it will fall back to `Fit`. `pdfstartview` and `pdfremotestartview` will fallback to `Fit` too.

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

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

13 Assorted key descriptions

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

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

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

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

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

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

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

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

will create

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

<code>nesting</code> (<i>setup key</i>)	Typically PDF viewer can handle only GoToR annotations pointing to a PDF. So normally the default value <code>pdf</code> 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.		
<code>pdfborder</code> (<i>setup key</i>)	This key set accept as value three numbers or three numbers and an array describing		
<code>linkborder</code> (<i>setup key</i>)	a dash pattern, examples are <code>0 0 1</code> or <code>0 0 1 [3 2]</code> . The first two numbers should		
<code>urlborder</code> (<i>setup key</i>)	according to the reference set round corners, but PDF viewer seem to ignore it. The		
<code>runborder</code> (<i>setup key</i>)	third number is the line width of the border. Settings done with <code>pdfborderstyle</code> should		
<code>menuborder</code> (<i>setup key</i>)	take precedence.		
<code>pdfborderstyle</code> (<i>setup key</i>)	The value of this key is the content of the BS dictionary. As an example		
<code>linkborderstyle</code> (<i>setup key</i>)	<code>/Type/Border /W 1 /S/U /D[3 2]</code>		
<code>urlborderstyle</code> (<i>setup key</i>)	Key	Values	comment / example
<code>fileborderstyle</code> (<i>setup key</i>)	<code>/Type</code>	<code>/Border</code>	optional
<code>runborderstyle</code> (<i>setup key</i>)	<code>/W</code>	<code><number></code>	Width of border line
<code>menuborderstyle</code> (<i>setup key</i>)	<code>/S</code>	<code>/S</code>	solid (default)
		<code>/D</code>	dash pattern can be set with <code>/D</code>
		<code>/B</code>	beveled
		<code>/I</code>	inset
		<code>/U</code>	underline
	<code>/D</code>	<code><array of numbers></code>	dash pattern (lines/gaps) (default [3])
<code>pdfcreationdate</code> (<i>setup key</i>)	Setting these keys is normally not needed. If they are used the values are stored di-		
<code>pdfmoddate</code> (<i>setup key</i>)	rectly in the Info dictionary for <code>/Creationdate</code> and <code>/ModDate</code> . The values are converted		
	to strings but not processed further, so they should have the correct PDF format without		
	the enclosing parentheses, e.g. <code>D:20200202111111+01'00'</code> .		
<code>pdflinkmargin</code> (<i>setup key</i>)	As described in the <code>hyperref</code> manual the behaviour differs between the backends:		
	with dvips it is possible to change links locally, pdf _l atex and lua _l atex work by page, with		
	dvipdfmx the setting is global (and has to be done in the preamble).		
<code>pdflang</code> (<i>setup key</i>)	The key will work, but it is recommended to the set the language in <code>\DeclareDocumentMetadata</code>		
	instead.		

File I

hyperref-generic driver implementation

```

1 <*package>
2 <@@=hyp>
3 \ProvidesFile{hgeneric-experimental.def}
4 [2020/12/26 v0.71 %
5 Hyperref driver for luaTeX]
6
7 \RequirePackage{etoolbox} %why?
8 \ExplSyntaxOn
9 \cs_gset_eq:NN \@expl@color@set@nnn \color_set:nnn
10 \file_input:n {hyperref-colorschemes.def}
11 \ExplSyntaxOff

```

1 messages

Redirect the message name:

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

At first a message for the testing of the resource management

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

The pdfversion should be set in \DeclareDocumentMetadata

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

A generic message for ignored keys.

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

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

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

pdf/A message for not allowed Named actions

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

A message if the destination name is empty.

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

A message if the destination check fails

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

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

```
72 \msg_new:nnn
73 { hyp }
74 { ignore-deprecated-or-unknown-option-in-pdf-version }
75 {
76   Option~'~#1'~is~unknown~or~deprecated~in\\
77   pdf~version~#2.~Ignored.
78 }
79 \msg_new:nnn
80 { hyp }
81 { ignore-deprecated-or-unknown-value-in-pdf-version }
82 {
83   Value~'~#1'~is~unknown~or~deprecated~in\\
84   pdf~version~#2.~Ignored.
85 }
86 \msg_new:nnn
87 { hyp }
88 { replace-deprecated-or-unknown-value-in-pdf-version }
89 {
90   Value~'~#1'~is~unknown~or~deprecated~in\\
91   pdf~version~#2. Value~'~#3'~is used instead.
92 }
```

During development not all standard hyperref keys are known and the Hyp-handler needs to process some new keys unknown to him. This issues warnings for now:

```
93 \msg_new:nnn
94 { hyp }
95 { unknown-key }
96 {
97   unknown~key~#2~of~module~'~#1'~set~to~'~#3'.
98 }
```

```

99 \msg_new:nnn
100 { hyp }
101 { unknown-key-to-Hyp }
102 {
103   ignored-in-family-Hyp-unknown-key-#1.
104 }

```

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

```

105 \cs_new:Npn \__hyp_clist_display:n #1 {*-#1\\}
106 \msg_new:nnn
107 { hyp }
108 { unknown-choice }
109 {
110   Value-#3'-is-invalid-for-key-#1'.\\
111   The-key-accepts-only-the-choices\\
112   \clist_map_function:nN { #2 }\__hyp_clist_display:n
113 }
114
115 \msg_new:nnn
116 { hyp }
117 { unknown-choice+empty }
118 {
119   Value-#3'-is-invalid-for-key-#1'.\\
120   The-key-accepts-only-the-choices\\
121   \clist_map_function:nN { #2 }\__hyp_clist_display:n
122   An-empty-value-removes-the-setting.
123 }
124
125 \msg_new:nnn
126 { hyp }
127 { no-bool }
128 {
129   Value-#2'-is-invalid-for-key-#1'.\\
130   The-key-accepts-only-the-choices\\
131   *-true\\
132   *-false \\
133   *-and-an-empty-value-which-removes-the-setting.\\
134   No-value-is-equivalent-to-using-true'.
135 }

```

2 Variants

```

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

```

3 Overwriting/providing commands from hyperref

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

```

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

```

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

```
140 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n
```

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

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

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

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

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

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

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

`\Acrobatmenu` should use the new internal link command

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

`\hypersetup` should set the new keys. We can't also execute `\kvsetkeys{Hyp}` as this errors for example with colors. This means the driver has to provide new code for every key!

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

4 Compability commands

4.1 Metadata

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

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

4.2 citecolor

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

```
183 \hook_new:n{hyp/link/cite}
184 \color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
185 \color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
186 \keys_define:nn { hyp / setup }
187 {
188   ,citecolor .code:n = {\__hyp_color_set:nn {hyp/color/cite}{#1}}
189   ,citebordercolor
190     .code:n = {\__hyp_color_set:nn {hyp/color/citeborder}{#1}}
191 }
192 \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
193 {
194   \keys_set:nn { hyp / setup }
195   {
196     ,linkbordercolor= hyp/color/citeborder
197     ,linkcolor      = hyp/color/cite
198   }
199 }
```

5 Checks

The driver can not work properly if the pdfmanagement is not active, as keys need to write to the catalog and to info. But annotations and outlines should work. So should this be a fatal error? Should there be a difference between missing and inactive management? TODO

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

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

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



```

208 {
209   \PassOptionsToPackage{draft}{bookmark}
210 }

```

6 Reference and label commands

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

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

```

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

```

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

```

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

```

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

```

221 \cs_new_protected:Npn \__hyp_ref_check:nn #1 #2 %label/attribute
222 {
223   \__hyp_ref_if_exist:nnF {#1}{#2}
224   {
225     \protect\G@refundefinedtrue
226     \@latex@warning
227     {
228       Reference~'\tl_to_str:n {#1}'~with~attribute~'\tl_to_str:n {#2}'~
229       on~page~\thepage~\space undefined
230     }
231   }
232 }
233 \cs_generate_variant:Nn \__hyp_ref_check:nn {en}

```

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

```

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

```

(End definition for __hyp_ref_label:nn and others.)

7 Variables

7.1 Private temporary variables

At first a few generic tmp variables

```

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

```

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

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

```

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

```

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.

```
250 \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 251 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
252 {url,link,file,menu,run}
253 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
254 {
255     URI    = url,
256     GoTo   = link,
257     GoToR  = file,
258     Named  = menu,
259     Launch = run
260 }
261 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
262 {
263     url    = URI,
264     link   = GoTo,
265     file   = GoToR,
266     menu   = Named,
267     run    = Launch
268 }
269

```

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

7.3 Variables

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

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

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

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

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

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

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

```
286 \tl_new:N \l__hyp_dest_pdfview_tl
```

(End definition for `\l__hyp_dest_pdfview_tl`.)

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

`hyp/annot/url` (color name)

`hyp/annot/file` (color name)

`\g__hyp_bordercolormodel_str` This holds the export model for border color etc. It is currently either `space-sep-cmyk`

`hyp/annot/run` (color name) or `space-sep-rgb`. The default is the second. It can be change by the key

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

```
287 \str_new:N \g__hyp_bordercolormodel_str
```

(End definition for `\g__hyp_bordercolormodel_str`.)

7.4 Booleans

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

```
\l_hyp_annot_colorlink_bool
\l_hyp_annot_colorurl_bool
\l_hyp_annot_colorfile_bool
\l_hyp_annot_colorrund_bool
\l_hyp_annot_colormenu_bool
288 \seq_map_inline:Nn \c__hyp_annot_types_seq
289 {
290   \bool_new:c {l_hyp_annot_color#1_bool}
291 }
```

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

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

```
\l_hyp_annot_ocgcolorlink_bool
\l_hyp_annot_ocgcolorurl_bool
\l_hyp_annot_ocgcolorfile_bool
\l_hyp_annot_ocgcolorrund_bool
\l_hyp_annot_ocgcolormenu_bool
292 \seq_map_inline:Nn \c__hyp_annot_types_seq
293 {
294   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
295 }
```

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

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

```
not_Named_bool_1111\l_hyp_annot_Launch_bool
296 \seq_map_inline:Nn \c_pdfannot_link_types_seq
297 {
298   \bool_new:c {l__hyp_annot_#1_bool}
299   \bool_set_true:c {l__hyp_annot_#1_bool}
300 }
```

(End definition for `\l__hyp_annot_GoTo_bool` `\l_hyp_annot_URI_bool` `\l_hyp_annot_GoToR_bool` `\l__hyp_annot_Named_bool` `\l_hyp_annot_Launch_bool`.)

7.5 Boxes

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

```
301 \box_new:N \l__hyp_dest_box
```

(End definition for `\l__hyp_dest_box`.)

7.6 Regex

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

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

(End definition for `\c__hyp_dest_startview_regex`.)

7.7 PDF dictionaries

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

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

(End definition for `l__hyp_page/Trans`.)

8 PDF string conversion

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

`hyp/text/pdfstring`

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

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

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

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

(End definition for `__hyp_text_purify:nN`.)

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

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

(End definition for `__hyp_text_cleanup:N`.)

The last step converts the string to a PDF encoding. As we have at least two targets (hex and literal) there is an argument. The conversion assumes utf8 input, it is based on `cspdf_text_convert:nnN` in `l3pdftools`.

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

utf8/string-print	(lit) (utf8/string)
utf8/string	lit (utf8/string)
utf8/URIpdf-print	(percent encoded url)
utf8/URIpdf	percent encoded url
utf16/hex-print	<HEX> (utf16/hex)
utf16/hex	HEX (utf16/hex)
utf16/string-print	(lit) (utf16/string)
utf16/string-lit	lit (utf16/string)

`__hyp_text_convert:nN`

```

326 \cs_new_protected:Npn \__hyp_text_convert:nN #1 #2
327 {
328   \pdf_text_convert:nVN { #1 } #2 #2
329 }

```

(End definition for __hyp_text_convert:nN.)

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

`__hyp_text_pdfstring:nnN`

```

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

```

(End definition for __hyp_text_pdfstring:nnN.)

!!! temporary until all instances are gone

```

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

```

This is a special version for info keys:

`__hyp_text_pdfstring_info:nN`

```

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

```

(End definition for __hyp_text_pdfstring_info:nN.)

9 Pagelabels

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

The current implementation in hyperref/hyperref drivers:

xetex: hxtex.def, line 80-110

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

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

dvipdfm: identical to xetex

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

luatex identical to pdftex

The code in `hyperref` inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-23 is correctly split. If the counter can not be identified `hyperref` generates only `/P` entries with the whole content.

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

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

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

```
\__hyp_PageLabels_gpush:
  \Hy@PutCatalog
  \HyPL@StorePageLabel
349 \cs_new_protected:Npn \__hyp_PageLabels_gpush:
350   {
351     \pdfmanagement_add:nnx {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
352   }
353
354 \def\Hy@PutCatalog #1 {}
355
356
357 \legacy_if:nT { Hy@pdfpagelabels }
358   {
359     \cs_set_protected:Npn \HyPL@StorePageLabel #1
360       {
361         \tl_gput_right:Nx \HyPL@Labels { \the\Hy@abspage<<#1>> }
362         \__hyp_PageLabels_gpush:
363       }
364   }
```

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

10 Core Hyperref Commands

Every `hyperref` has to define eight core command:

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

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

10.1 Anchors / destinations

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

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

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

```
\__hyp_destination:nn

365 \Hy@WrapperDef \__hyp_destination:nn #1 #2
366 {
367   \mode_if_horizontal:T { \@savs\spacefactor }
368   \Hy@SaveLastskip      %defined in hyperref
369   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
370   \__hyp_text_pdfstring:xoN
371   { \HyperDestNameFilter{#1} }
372   { \l__hyp_text_enc_dest_tl }
373   \l__hyp_tmpa_tl
374   \str_if_eq:nnTF {#2} {fitrbx}
375   {
376     \exp_args:NV
377     \pdf_destination:nnnn \l__hyp_tmpa_tl
378     { \box_wd:N \l__hyp_dest_box }
379     { \box_ht:N \l__hyp_dest_box }
380     { \box_dp:N \l__hyp_dest_box }
381   }
382   {
383     \exp_args:NV
384     \pdf_destination:nf
385     { \l__hyp_tmpa_tl }
386     { #2 }
387   }
388   \Hy@RestoreLastskip %defined in hyperref
389   \mode_if_horizontal:T { \spacefactor\@savs }
390 }
```

(End definition for `__hyp_destination:nn`.)

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

```
\hyper@anchor
\hyper@anchorstart
\hyper@anchorend

391 \cs_new_protected:Npn \hyper@anchor #1
392 {
393   \exp_args:NnV
394   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
395 }
```



```

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

```

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

10.2 GoTo Links

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

At first the internal link commands:

```

408 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
409 {
410   \mode_leave_vertical:
411   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
412   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
413   {
414     \msg_warning:nnx
415     { hyp }
416     { empty-destination-name }
417     { \c__hyp_dest_undefined_tl }
418     \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
419   }
420   {
421     \__hyp_text_pdfstring:xoN
422     { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
423     { \l__hyp_text_enc_dest_tl }
424     \l__hyp_dest_name_tmpa_tl
425   }
426   \exp_args:No
427   \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
428 }
429
430 \cs_new_protected:Npn \__hyp_link_goto_end:
431 {
432   \pdfannot_link_goto_end:
433 }

```

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

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

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

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

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

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

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

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

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

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

```
434
435 \cs_new_protected:Npn \hyper@link #1 #2 #3 {%#1 context, #2=destination name, #3 content
436 {
437   \bool_if:NTF \l__hyp_annot_GoTo_bool
438   {
439     \Hy@VerboseLinkStart{#1}{#2}
440     \group_begin:
441     \hook_use:n {hyp/link/#1}
442     \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
443     \__hyp_link_goto_end:
444     \group_end:
445     \Hy@VerboseLinkStop
446   }{#3}
447 }
448
449 \cs_new_protected:Npn \hyper@linkstart #1 #2 {%#1 context, #2=destination name
450 {
451   \bool_if:NT \l__hyp_annot_GoTo_bool
452   {
453     \Hy@VerboseLinkStart{#1}{#2}% only for debug
454     \group_begin:
455     \hook_use:n {hyp/link/#1}
456     \__hyp_link_goto_begin:nw {#2}
457   }
458 }
459
460 \cs_new_protected:Npn \hyper@linkend
461 {
462   \bool_if:NT \l__hyp_annot_GoTo_bool
463   {
464     \__hyp_link_goto_end:
465     \group_end:
466     \Hy@VerboseLinkStop
```

```

467     }
468 }

```

10.3 URI links

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

```

469 \pdfdict_new:n {l_hyp/annot/A/URI}
470 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
471 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
472
473 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
474 {
475     \bool_if:NTF \l__hyp_annot_URI_bool
476     {
477         \group_begin:
478         \__hyp_text_pdfstring:xoN
479         { #2}
480         { \l__hyp_text_enc_uri_print_tl }
481         \l__hyp_uri_tmpa_tl
482         \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
483         \ifHy@href@ismap
484             \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
485         \fi
486         \tl_if_empty:NF \Hy@href@nextactionraw
487         {
488             \str_remove_once:Nn \Hy@href@nextactionraw {/Next}
489             \pdfdict_put:nno{l_hyp/annot/A/Next}{\Hy@href@nextactionraw}
490         }
491         \cs_set_eq:NN \# \c_hash_str
492         \cs_set_eq:NN \% \c_percent_str
493         \Hy@safe@activetrue
494         \mode_leave_vertical:
495         \pdfannot_link:nxn { URI }
496         {
497             /A
498             <<
499             \pdfdict_use:n {l_hyp/annot/A/URI}
500             >>
501         }
502         {
503             #1
504             \Hy@xspace@end
505             \Hy@VerboseLinkStop %where is the start??
506         }
507         \group_end:
508     }
509     {#1}
510 }
511

```

10.4 GoToR Links files

```

512 \pdfdict_new:n {l_hyp/annot/A/GoToR}

```

```

513 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
514 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
515
516 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
517 {
518   \bool_if:NTF \l__hyp_annot_GoToR_bool
519   {
520     \group_begin:
521     \tl_set:Nn \l__hyp_filename_tmpa_tl { #2 }
522     \Hy@CleanupFile \l__hyp_filename_tmpa_tl
523     \__hyp_text_pdfstring:ooN
524     { \l__hyp_filename_tmpa_tl }
525     { \l__hyp_text_enc_file_print_tl }
526     \l__hyp_filename_tmpa_tl
527     \pdfdict_put:nno {l_hyp/annot/A/GoToR}{F}{\l__hyp_filename_tmpa_tl}
528     \__hyp_text_pdfstring:nnN
529     { #3 }
530     { \l__hyp_text_enc_dest_print_tl }
531     \l__hyp_dest_name_tmpa_tl
532     \Hy@MakeRemoteAction
533     \tl_if_blank:eTF {#3}
534     {
535       \pdfdict_put:nnx {l_hyp/annot/A/GoToR}{D}
536       {[ \Hy@href@page/\l__hyp_dest_pdfremotestartview_tl]}
537     }
538     {
539       \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
540     }
541     \tl_if_empty:NF \Hy@href@nextactionraw
542     {
543       \str_remove_once:Nn \Hy@href@nextactionraw {/Next}
544       \pdfdict_put:nno{l_hyp/annot/A/Next}{Next}{\Hy@href@nextactionraw}
545     }
546     \mode_leave_vertical:
547     \pdfannot_link:nxn %expansion??
548     { GoToR }
549     {
550       /A<<
551       \pdfdict_use:n {l_hyp/annot/A/GoToR}
552       >>
553     }
554     {
555       #1\Hy@xspace@end
556       \Hy@VerboseLinkStop %where is the start??
557     }
558     \group_end:
559   }
560   {#1}
561 }

```

10.5 Launch links

We define \hyper@linklaunch for naming consistency

```

562 \pdfdict_new:n {l_hyp/annot/A/Launch}

```

```

563 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
564 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
565
566 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
567 {
568   \bool_if:NTF \l__hyp_annot_Launch_bool
569   {
570     \group_begin:
571     \__hyp_text_pdfstring:nnN
572       { #1 }
573       { \l__hyp_text_enc_file_print_tl }
574       \l__hyp_filename_tmpa_tl
575     \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
576     \__hyp_text_pdfstring:noN
577       { #3 }
578       { \l__hyp_text_enc_para_print_tl }
579       \l__hyp_para_tmpa_tl
580     \bool_if:nTF
581     {
582       \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
583       ||
584       \pdf_version_compare_p:Nn > {1.9}
585     }
586     {
587       \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
588     }
589     {
590       \pdfdict_put:nnx {l_hyp/annot/A/Launch}{Win}
591       {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
592     }
593     \tl_if_empty:NF \Hy@href@nextactionraw
594     {
595       \str_remove_once:Nn \Hy@href@nextactionraw {/Next}
596       \pdfdict_put:nno{l_hyp/annot/A/Next}{Next}{\Hy@href@nextactionraw}
597     }
598     \mode_leave_vertical:
599     \pdfannot_link:nxn
600       { Launch }
601       {
602         /A
603         <<
604         \pdfdict_use:n {l_hyp/annot/A/Launch}
605         >>
606       }
607       {
608         #2\Hy@xspace@end
609         \Hy@VerboseLinkStop %where is the start??
610       }
611     \group_end:
612   }
613   {#2}
614 }

```

The actually command used by hyperref is `\@hyper@launch` which uses a delimited ar-

gument, because of the color the definition is a bit convoluted.

```

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

```

10.6 Named links (menu)

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

```

622 \pdfdict_new:n {l_hyp/annot/A/Named}
623 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
624 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
625
626 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
627 {
628   \bool_if:NTF \l__hyp_annot_Named_bool
629   {
630     \group_begin:
631     \bool_set_true:N\l_tmpa_bool
632     \prop_get:NnNT \g_pdfmeta_standard_prop {named_actions} \l__hyp_tmpa_tl
633     {
634       \tl_if_in:NnF \l__hyp_tmpa_tl {#1}
635       {
636         \bool_set_false:N\l_tmpa_bool
637       }
638     }
639     \bool_if:NTF \l_tmpa_bool
640     {
641       \mode_leave_vertical:
642       \pdfdict_put:nnx {l_hyp/annot/A/Named}{N}
643       {\pdf_text_convert:nn{name-print}{#1}}
644       \tl_if_empty:NF \Hy@href@nextactionraw
645       {
646         \str_remove_once:Nn \Hy@href@nextactionraw {/Next}
647         \pdfdict_put:nno{l_hyp/annot/A/Next}{Next}{\Hy@href@nextactionraw}
648       }
649       \pdfannot_link:nxn { Named }
650       {
651         /A
652         <<
653         \pdfdict_use:n { l_hyp/annot/A/Named }
654         >>
655       }
656       {
657         #2
658         \Hy@xspace@end
659         \Hy@VerboseLinkStop
660       }
661     }
662   }

```

```

663         \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
664         #2
665     }
666     \group_end:
667 }
668 {#2}
669 }
670

```

11 Link decorations

11.1 Functions to export and select colors

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

```

\__hyp_color_export:nnN \__hyp_color_export:nnN {<color>} {<export format>} <tlvar>
\__hyp_colormodel_export:nnnN \__hyp_colormodel_export:nnnN {<color model>}{<value>} {<export format>}
<tlvar>

```

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).

```

671 \cs_new_protected:Npn \__hyp_color_export:nnN #1 #2 #3
672 {
673     \tl_if_head_eq_charcode:nTF {#1}[ %]
674     {
675         \__hyp_colormodel_export:wnnN #1 {#2} #3
676     }
677     {
678         \color_export:nnN {#1} {#2} #3
679     }
680 }
681
682 \cs_new_protected:Npn \__hyp_colormodel_export:wnnN [#1] #2 #3 #4
683 {
684     \color_export:nnnN {#1}{#2}{#3}#4
685 }
686
687 \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`.

```

688 \cs_new_protected:Npn \_hyp_color_select:n #1
689 {
690   \tl_if_head_eq_charcode:nNTF {#1}[ %]
691   {
692     \_hyp_colormodel_select_aux:wn #1
693   }
694   {
695     \color_select:n {#1}
696   }
697 }
698
699 \cs_new_protected:Npn \_hyp_colormodel_select_aux:wn [#1] #2
700 {
701   \color_select:nn {#1}{#2}
702 }
703
704 \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`.

```

705 \cs_new_protected:Npn \_hyp_color_set:nn #1 #2
706 {
707   \tl_if_head_eq_charcode:nNTF {#2}[ %]
708   {
709     \_hyp_colormodel_set_aux:nwn { #1 } #2
710   }
711   {
712     \color_set:nn {#1} {#2}
713   }
714 }
715
716 \cs_new_protected:Npn \_hyp_colormodel_set_aux:nwn #1 [#2] #3
717 {
718   \color_set:nnn {#1}{#2}{#3}
719 }
720
721 \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.


```

722 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
723 {
724   \hook_gput_code:nnn
725     {pdfannot/link/#2/begin}
726     {hyp/color}
727     {
728       \bool_if:cT { l_hyp_annot_color#1_bool }
729       {
730         \group_begin:
731         \color_select:n { hyp/color/#1}
732       }
733     }
734   \hook_gput_code:nnn
735     {pdfannot/link/#2/end}
736     {hyp/color}
737     {
738       \bool_if:cT { l_hyp_annot_color#1_bool }
739       {
740         \group_end:
741       }
742     }
743 }

```

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

```

744 \keys_define:nn { hyp / setup }
745 {
746   ,colorlinks .meta:n =
747     {
748       ,pdfborder={0~0~0}
749       ,pdfborderstyle=
750       ,colorurl =#1
751       ,colorlink =#1
752       ,colorrun =#1
753       ,colormenu =#1
754       ,colorfile =#1
755     }
756   ,colorlinks .default:n = {true}
757 }

```

`colorurl` (*setup key*)

`colorlink` (*setup key*)

`colorrun` (*setup key*)

`colormenu` (*setup key*)

`colorfile` (*setup key*)

`urlcolor` (*setup key*)

`linkcolor` (*setup key*)

`runcolor` (*setup key*)

`menucolor` (*setup key*)

`filecolor` (*setup key*)

`allcolor` (*setup key*)

```

758 \seq_map_inline:Nn \c__hyp_annot_types_seq
759 {
760   \keys_define:nn { hyp / setup }
761   {
762     ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
763     ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
764   }
765 }
766
767 \keys_define:nn { hyp / setup }
768 {
769   ,allcolors .meta:n =
770   {
771     ,urlcolor=#1

```

```

772         ,linkcolor=#1
773         ,runcolor=#1
774         ,filecolor=#1
775         ,menucolor=#1
776     }
777     ,allcolors .value_required:n = true
778 }

```

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*)

```

779 \keys_define:nn { hyp / setup }
780 {
781     ,bordercolormodel .choices:nn =
782     {rgb,cmyk}
783     { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
784     ,bordercolormodel .initial:n = {rgb}
785 }
786 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
787 {
788     \keys_define:nn { hyp / setup }
789     {
790         #1bordercolor .code:n =
791         {
792             \tl_if_empty:nTF { ##1 }
793             {
794                 \pdfannot_dict_remove:nn
795                 {link/#2}
796                 { C }
797             }
798             {
799                 \__hyp_color_export:xVN {##1}\g__hyp_bordercolormodel_str \l__hyp_tmpa_tl
800                 \pdfannot_dict_put:nnx
801                 {link/#2}
802                 { C }
803                 { [\l__hyp_tmpa_tl] }
804             }
805         }
806     }
807 }
808
809 \keys_define:nn { hyp / setup }
810 {
811     ,allbordercolors .meta:n =
812     {
813         ,linkbordercolor=#1
814         ,urlbordercolor =#1
815         ,filebordercolor=#1
816         ,menubordercolor=#1

```

```

817         ,runbordercolor =#1
818     }
819     ,allbordercolors .value_required:n = true
820 }
821

```

11.3.2 Borderwidth and -arc

```

822 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
823 {
824     \keys_define:nn { hyp / setup }
825     {
826         #1border .code:n =
827         {
828             \tl_if_empty:nTF { ##1 }
829             {
830                 \pdfannot_dict_remove:nn
831                 {link/#2}
832                 { Border }
833             }
834             {
835                 \pdfannot_dict_put:nnn
836                 {link/#2}
837                 { Border }
838                 { [##1] }
839             }
840         }
841     }
842 }
843 \keys_define:nn { hyp / setup }
844 {
845     ,pdfborder .code:n =
846     {
847         \tl_if_empty:nTF { #1 }
848         {
849             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
850             {
851                 \pdfannot_dict_remove:nn
852                 {link/##2}
853                 { Border }
854             }
855         }
856         {
857             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
858             {
859                 \pdfannot_dict_put:nnn
860                 {link/##2}
861                 { Border }
862                 { [#1] }
863             }
864         }
865     }
866     ,pdfborder .initial:n = {0~0~1},
867 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 868 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 869 {
runborderstyle (setup key) 870   \keys_define:nn { hyp / setup }
fileborderstyle (setup key) 871   {
menuborderstyle (setup key) 872     #1borderstyle .code:n =
873     {
874       \tl_if_empty:nTF { ##1 }
875       {
876         \pdfannot_dict_remove:nn
877         {link/##2}
878         { BS }
879       }
880       {
881         \pdfannot_dict_put:nnn
882         {link/##2}
883         { BS }
884         { <<##1>> }
885       }
886     }
887   }
888 }
889 \keys_define:nn { hyp / setup }
890 {
891   ,pdfborderstyle .code:n =
892   {
893     \tl_if_empty:nTF { #1 }
894     {
895       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
896       {
897         \pdfannot_dict_remove:nn
898         {link/##2}
899         { BS }
900       }
901     }
902     {
903       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
904       {
905         \pdfannot_dict_put:nnn
906         {link/##2}
907         { BS }
908         { <<##1>> }
909       }
910     }
911   }
912   ,pdfborderstyle .initial:n = {},
913 }

```

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.

```
914 \cs_new_protected:Npn \__hyp_ocg_init:
915 {
916   \pdf_object_new:nn { l__hyp_ocg_view_dict_obj } { dict }
917   \pdf_object_new:nn { l__hyp_ocg_print_dict_obj } { dict }
918   \pdf_object_new:nn { l__hyp_ocg_config_dict_obj } { dict }
919   \pdf_object_new:nn { l__hyp_ocg_ref_array_obj } { array }
920   \pdf_object_write:nx { l__hyp_ocg_ref_array_obj }
921   {
922     \pdf_object_ref:n { l__hyp_ocg_view_dict_obj }
923     \c_space_tl
924     \pdf_object_ref:n { l__hyp_ocg_print_dict_obj }
925   }
926   \pdf_object_write:nn { l__hyp_ocg_view_dict_obj }
927   {
928     /Type/OCG
929     /Name(View)
930     /Usage
931     <<
932       /Print <</PrintState/OFF>>~
933       /View <</ViewState/ON >>~
934     >>
935   }
936   \pdf_object_write:nn { l__hyp_ocg_print_dict_obj }
937   {
938     /Type/OCG
939     /Name(Print)
940     /Usage
941     <<
942       /Print <</PrintState/ON>>~
943       /View <</ViewState/OFF>>~
944     >>
945   }
946   \pdfmanagement_add:nxx { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {l__hyp_o
947   \pdfmanagement_add:nxx { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {l__hyp_o
948   \pdf_object_write:nx { l__hyp_ocg_config_dict_obj }
949   {
950     /OFF[\pdf_object_ref:n { l__hyp_ocg_print_dict_obj }]
951     /AS[
952       <<
953         /Event/View
954         /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
955         /Category[/View]
956       >>
957       <<
958         /Event/Print
959         /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
960         /Category[/Print]
```

```

961         >>
962         <<
963         /Event/Export
964         /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
965         /Category[/Print]
966         >>
967         ]
968     }
969     \pdfmanagement_add:nnx { Catalog / OCPProperties }{ D }{ \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
970     \cs_gset:Npn \__hyp_ocg_init: {}
971 }

```

(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.

```

972 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
973 {
974     \hook_gput_code:nnn
975     {pdfannot/link/#2/begin}
976     {hyp/ocg}
977     {
978         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
979         {
980             \__hyp_ocg_init:
981             \group_begin:
982             \hbox_set:Nw \l__hyp_tmpa_box
983         }
984     }
985     \hook_gput_code:nnn
986     {pdfannot/link/#2/end}
987     {hyp/ocg}
988     {
989         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
990         {
991             \hbox_set_end:
992             \mbox
993             {
994                 \pdf_bdc_obj:nn {OC}{l__hyp_ocg_print_dict_obj}
995                 \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
996                 \pdf_emc:
997                 \pdf_bdc_obj:nn {OC}{l__hyp_ocg_view_dict_obj}
998                 \group_begin:
999                 \color_select:n { hyp/color/#1 }
1000                 \box_use_drop:N \l__hyp_tmpa_box
1001                 \group_end:
1002                 \pdf_emc:
1003             }
1004             \group_end:
1005         }
1006     }
1007 }

```

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

ocgcolorlink (setup key) 1008 \pdf_version_compare:NnTF < {1.5}

ocgcolorurl (setup key)

ocgcolorfile (setup key)

ocgcolormenu (setup key)

ocgcolorrun (setup key)

```

1009 {
1010   \keys_define:nn { hyp / setup }
1011   {
1012     ,_ocgcolorlinks .code:n =
1013     {
1014       \msg_warning:nnxx
1015       { hyp }
1016       { ignore-deprecated-or-unknown-option-in-pdf-version }
1017       { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1018     }
1019   }
1020 }
1021 {
1022   \keys_define:nn { hyp / setup }
1023   {
1024     ,_ocgcolorlinks .meta:n =
1025     {
1026       ocgcolorlink=#1,
1027       ocgcolorurl=#1,
1028       ocgcolorfile=#1,
1029       ocgcolorrun=#1,
1030       ocgcolormenu=#1
1031     }
1032     ,_ocgcolorlinks .default:n = true
1033   }
1034 }
1035
1036 \keys_define:nn { hyp / setup }
1037 {
1038   ,ocgcolorlinks .choice:
1039   ,ocgcolorlinks / true .meta:n =
1040   {
1041     pdfborder      = {0~0~0},
1042     pdfborderstyle = {},
1043     colorlinks     = false,
1044     _ocgcolorlinks = true
1045   }
1046   ,ocgcolorlinks / false .meta:n =
1047   {
1048     _ocgcolorlinks = false
1049   }
1050   ,ocgcolorlinks .default:n = {true}
1051 }
1052
1053 \seq_map_inline:Nn \c__hyp_annot_types_seq
1054 {
1055   \pdf_version_compare:NnTF < {1.5}
1056   {
1057     \keys_define:nn { hyp / setup }
1058     {
1059       ,ocgcolor#1 .code:n=
1060       {
1061         \msg_warning:nnxx
1062         { hyp }

```

```

1063         { ignore-deprecated-or-unknown-option-in-pdf-version }
1064         { ocgcolor#1 }
1065         { \pdf_version_major:\pdf_version_minor: }
1066     }
1067 }
1068 }
1069 {
1070     \keys_define:nn { hyp / setup }
1071     {
1072         ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1073     }
1074 }
1075 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1076 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1077 {
1078     \keys_define:nn { hyp / setup }
1079     {
1080         ,#1highlight .choices:nn =
1081         { /I, /N, /O, /P }
1082         {
1083             \pdfannot_dict_put:nnn
1084             {link/#2}
1085             { H }
1086             { ##1 }
1087         }
1088         ,#1highlight / .code:n =
1089         {
1090             \pdfannot_dict_remove:nn
1091             {link/#2}
1092             { H }
1093         }
1094     }
1095     ,#1highlight / unknown .code:n =
1096     {
1097         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1098         { pdfhighlight }
1099         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1100         { \exp_not:n {##1} }
1101     }
1102 }
1103 }
1104 }
1105
1106
1107 \keys_define:nn { hyp / setup }
1108 {
1109     ,pdfhighlight .choices:nn =
1110     { /I, /N, /O, /P }
1111     {
1112         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop

```



```

1113         {
1114             \pdfannot_dict_put:nnn
1115             {link/####2}
1116             { H }
1117             { #1 }
1118         }
1119     }
1120     ,pdfhighlight / .code:n =
1121     {
1122         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1123         {
1124             \pdfannot_dict_remove:nn
1125             {link/##2}
1126             { H }
1127         }
1128     }
1129     ,pdfhighlight .initial:n = {/I},
1130     ,pdfhighlight / unknown .code:n =
1131     {
1132         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1133         { pdfhighlight }
1134         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1135         { \exp_not:n {#1} }
1136     }
1137 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1138 \keys_define:nn { hyp / setup }
hideurl (setup key) 1139 {
hidefile (setup key) 1140     hidelinks .meta:n =
hiderun (setup key) 1141     {
hidemenu (setup key) 1142         ,colorlinks      = false
1143         ,ocgcolorlinks = false
1144         ,pdfborder      = { 0~0~0 }
1145         ,pdfborderstyle=
1146     }
1147 }
1148
1149 \seq_map_inline:Nn \c__hyp_annot_types_seq
1150 {
1151     \keys_define:nn { hyp / setup }
1152     {
1153         hide#1 .meta:n =
1154         {
1155             ,color#1      = false
1156             ,ocgcolor#1   = false
1157             ,#1border      = { 0~0~0 }
1158             ,#1borderstyle =
1159         }
1160     }

```

```
1161 }
```

11.7 color schemes and settings

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

`colorscheme (setup key)`

```
1162 \keys_define:nn { hyp / setup }
1163 {
1164   colorscheme .code:n =
1165   {
1166     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1167     {
1168       \keys_set:nn { hyp / setup }
1169       {
1170         ##1 = ##2
1171       }
1172     }
1173   }
1174 }
1175 \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)`

```
1176 \keys_define:nn { hyp / setup }
1177 {
1178   ,unicode .code:n = {}
1179   ,pdfencoding .code:n = {}
1180   ,pdfversion .code:n =
1181   {
1182     \msg_warning:nn { hyp }{ pdfversion-disabled }
1183   }
1184 }
1185 %
```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

`verbose (setup key)`

`debug (setup key)`

`draft (setup key)`

`final (setup key)`

```
1186 \keys_define:nn { hyp / setup }
1187 {
1188   ,verbose .choice:
1189   ,verbose / true .code:n = { \Hy@verbosetrue}
1190   ,verbose / false .code:n = { \Hy@verbosefalse}
1191   ,verbose .default:n = {true}
1192   ,debug .meta:n = {verbose=#1}
1193   ,debug .default:n = {true}
```

```

1194 }
1195 \keys_define:nn { hyp / setup }
1196 {
1197   ,draft .code:n =
1198   {
1199     \Hy@drafttrue
1200     \PassOptionsToPackage{draft}{bookmark}
1201   }
1202   ,final .code:n =
1203   {
1204     \Hy@finaltrue
1205     \PassOptionsToPackage{final}{bookmark}
1206   }
1207 }

extension (setup key)
hypertextnames (setup key) 1208 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1209 {
pageanchor (setup key) 1210   ,extension .tl_set:N = \XR@ext
linktoc (setup key) 1211   ,extension .initial:n= pdf
linktocpage (setup key) 1212   ,hypertextnames .choice:
plainpages (setup key) 1213   ,hypertextnames / true .code:n = { \Hy@hypertextnamestrue}
localanchorname (setup key) 1214   ,hypertextnames / false .code:n = { \Hy@hypertextnamesfalse}
linkfileprefix (setup key) 1215   ,hypertextnames .default:n = {true}
1216   ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1217   ,localanchorname .choice:
1218   ,localanchorname / true .code:n = { \Hy@localanchornamestrue }
1219   ,localanchorname / false .code:n = { \Hy@localanchornamefalse }
1220   ,localanchorname .default:n = {true}
1221   ,naturalnames .choice:
1222   ,naturalnames / true .code:n = { \Hy@naturalnamestrue}
1223   ,naturalnames / false .code:n = { \Hy@naturalnamesfalse}
1224   ,naturalnames .default:n = {true}
1225   ,pageanchor .choice:
1226   ,pageanchor / true .code:n = { \Hy@pageanchortrue}
1227   ,pageanchor / false .code:n = { \Hy@pageanchorfalse}
1228   ,pageanchor .default:n = {true}
1229   ,plainpages .choice:
1230   ,plainpages / true .code:n = { \Hy@plainpagestrue}
1231   ,plainpages / false .code:n = { \Hy@plainpagesfalse}
1232   ,plainpages .default:n = {true}
1233 }
1234
1235 \keys_define:nn { hyp / setup }
1236 {
1237   ,linktoc .choices:nn = { none, section, all, page }
1238   {
1239     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1240   }
1241   ,linktoc / unknown .code:n =
1242   {
1243     \msg_warning:nnxxx { hyp } { unknown-choice }
1244     { linktoc }
1245     { none, section, all, page }

```

```

1246         { \exp_not:n {#1} }
1247     }
1248     ,linktocpage .choice:
1249     ,linktocpage / true .meta:n = {linktoc=page}
1250     ,linktocpage / false .meta:n = {linktoc=section}
1251     ,linktocpage .default:n = true
1252 }
1253
link (setup key) This booleans allow to disable the link types.
url (setup key) 1254 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1255 {
menu (setup key) 1256     \keys_define:nn { hyp / setup }
run (setup key) 1257     {
1258         ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1259     }
1260 }

1261 \keys_define:nn { hyp / setup }
1262 {
1263     ,baseurl .code:n =
1264     {
1265         \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1266         \tl_if_empty:NTF \l__hyp_tmpa_tl
1267         {
1268             \pdfmanagement_remove:nn {Catalog} { URI }
1269         }
1270         {
1271             \pdfmanagement_add:nnx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1272         }
1273     }
1274     %only false does something ...
1275     ,bookmarks .choice:
1276     ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1277     ,bookmarks / true .code:n = {}
1278     ,bookmarks .default:n = {true}
1279     ,bookmarksnumbered .choice:
1280     ,bookmarksnumbered / false .code:n = { \Hy@bookmarksnumberedfalse }
1281     ,bookmarksnumbered / true .code:n = { \Hy@bookmarksnumberedtrue }
1282     ,bookmarksnumbered .default:n = {true}
1283     ,bookmarksopen .choice:
1284     ,bookmarksopen / false .code:n = { \Hy@bookmarksopenfalse }
1285     ,bookmarksopen / true .code:n = { \Hy@bookmarksopentruetrue }
1286     ,bookmarksopen .default:n = {true}
1287     ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1288     ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1289     ,pdfcenterwindow .choice:
1290     ,pdfcenterwindow / false .code:n =
1291     {
1292         \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1293     }
1294     ,pdfcenterwindow / true .code:n =
1295     {
1296         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }

```

```

1297     }
1298 ,pdfcenterwindow / .code:n =
1299     {
1300         \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1301     }
1302 ,pdfcenterwindow / unknown .code:n =
1303     {
1304         \msg_warning:nnxx { hyp } { no-bool }
1305         { pdfcenterwindow }
1306         { \exp_not:n {#1} }
1307     }
1308 ,pdfcenterwindow .default:n = true
1309 ,pdfdirection .choice:
1310 ,pdfdirection / L2R .code:n =
1311     {
1312         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1313     }
1314 ,pdfdirection / R2L .code:n =
1315     {
1316         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1317     }
1318 ,pdfdirection / .code:n =
1319     {
1320         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1321     }
1322 ,pdfdirection / unknown .code:n =
1323     {
1324         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1325         { pdfdirection }
1326         { L2R , R2L }
1327         { \exp_not:n {#1} }
1328     }
1329 ,pdfdisplaydoctitle .choice:
1330 ,pdfdisplaydoctitle / false .code:n =
1331     {
1332         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1333     }
1334 ,pdfdisplaydoctitle / true .code:n =
1335     {
1336         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1337     }
1338 ,pdfdisplaydoctitle .default:n = true
1339 ,pdfduplex .choices:nn =
1340 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1341 {
1342     \pdf_version_compare:NnTF > {1.6}
1343     {
1344         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1345         { PrintDuplex } { /#1 }
1346     }
1347     {
1348         \msg_warning:nnxx
1349         {hyp}
1350         {ignore-deprecated-or-unknown-option-in-pdf-version}

```

```

1351         {pdfduplex}
1352         {\pdf_version:}
1353     }
1354 }%
1355 ,pdfduplex / .code:n =
1356 {
1357     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1358 }
1359 ,pdfduplex / unknown .code:n =
1360 {
1361     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1362     { pdfduplex }
1363     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1364     { \exp_not:n {#1} }
1365 }
1366 ,pdffitwindow .choice:
1367 ,pdffitwindow / false .code:n =
1368 {
1369     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1370 }
1371 ,pdffitwindow / true .code:n =
1372 {
1373     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1374 }
1375 ,pdffitwindow / .code:n =
1376 {
1377     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1378 }
1379 ,pdffitwindow .default:n = true
1380 ,pdffitwindow / unknown .code:n =
1381 {
1382     \msg_warning:nnxx { hyp } { no-bool }
1383     { pdffitwindow }
1384     { \exp_not:n {#1} }
1385 }
1386 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1387 ,pdflinkmargin .initial:n = {1pt}
1388 ,pdfmenubar .choice:
1389 ,pdfmenubar / true .code:n =
1390 {
1391     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1392 }
1393 ,pdfmenubar / false .code:n =
1394 {
1395     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1396     { HideMenubar } { true }
1397 }
1398 ,pdfmenubar / .code:n =
1399 {
1400     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1401 }
1402 ,pdfmenubar .default:n = true
1403 ,pdfmenubar / unknown .code:n =
1404 {

```

```

1405     \msg_warning:nnxx { hyp } { no-bool }
1406     { pdfmenubar }
1407     { \exp_not:n {#1} }
1408 }
1409 ,pdfnewwindow .choice:
1410 ,pdfnewwindow / true .code:n =
1411 {
1412     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1413     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1414 }
1415 ,pdfnewwindow / false .code:n =
1416 {
1417     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1418     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1419 }
1420 ,pdfnewwindow / .code:n =
1421 {
1422     \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1423     \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1424 }
1425 ,pdfnonfullscreenpagemode .choices:nn =
1426 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1427 {
1428     \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1429     { NonFullScreenPageMode } {/#1}
1430 }
1431 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1432 {
1433     \pdf_version_compare:NnTF < {1.6}
1434     {
1435         %message
1436     }
1437     {
1438         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1439         {NonFullScreenPageMode}{/UseAttachments}
1440     }
1441 }
1442 ,pdfnonfullscreenpagemode / .code:n =
1443 {
1444     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1445 }
1446 ,pdfnonfullscreenpagemode / unknown .code:n =
1447 {
1448     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1449     { pdfnonfullscreenpagemode }
1450     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1451     { \exp_not:n {#1} }
1452 }
1453 ,pdfnumcopies .code:n =
1454 {
1455     \pdf_version_compare:NnTF > {1.6}
1456     {
1457         \tl_if_empty:nTF {#1}
1458         {

```

```

1459         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1460     }
1461     {
1462         \pdfmanagement_add:nxx {Catalog / ViewerPreferences }
1463         {NumCopies}{#1}
1464     }
1465 }
1466 {
1467     \msg_warning:nxxx
1468     {hyp}
1469     {ignore-deprecated-or-unknown-option-in-pdf-version}
1470     {pdfnumcopies}
1471     {\pdf_version:}
1472 }
1473 }
1474 ,pdfpagelayout .choices:nn =
1475 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1476 { \pdfmanagement_add:nxx {Catalog} { PageLayout }{ /#1 } }
1477 ,pdfpagelayout / .code:n =
1478 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1479 ,pdfpagelayout / unknown .code:n =
1480 {
1481     \msg_warning:nxxx { hyp } { unknown-choice+empty }
1482     { pdfpagelayout }
1483     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1484     { \exp_not:n {#1} }
1485 }
1486 ,pdfpagemode .choices:nn =
1487 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1488 { \pdfmanagement_add:nxx {Catalog} { PageMode }{ /#1 } }
1489 ,pdfpagemode / UseAttachments .code:n =
1490 {
1491     \pdf_version_compare:NnTF > {1.5}
1492     {
1493         \pdfmanagement_add:nxx {Catalog} { PageMode }{ /UseAttachments }
1494     }
1495     {
1496         \msg_warning:nxxx
1497         {hyp}
1498         {ignore-deprecated-or-unknown-value-in-pdf-version}
1499         {UseAttachments}
1500         {\pdf_version:}
1501     }
1502 }
1503 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1504 ,pdfpagemode / unknown .code:n =
1505 {
1506     \msg_warning:nxxx { hyp } { unknown-choice+empty }
1507     { pdfpagemode }
1508     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1509     { \exp_not:n {#1} }
1510 }
1511 ,pdfpagescrop .code:n =
1512 {

```



```

1513     \tl_if_empty:nTF %or blank?
1514     {
1515         \pdfmanagement_remove:nn {Pages} { CropBox }
1516     }
1517     {
1518         \pdfmanagement_add:nxx {Pages} { CropBox } { [#1] }
1519     }
1520 }
1521 ,pdfpicktraybypdfsize .choice:
1522 ,pdfpicktraybypdfsize / true .code:n =
1523 {
1524     \pdf_version_compare:NnTF > {1.6}
1525     {
1526         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1527         { PickTrayByPDFSize } { true }
1528     }
1529     {
1530         \msg_warning:nxxx
1531         {hyp}
1532         {ignore-deprecated-or-unknown-option-in-pdf-version}
1533         {pdfpicktraybypdfsize}
1534         {\pdf_version:}
1535     }
1536 }
1537 ,pdfpicktraybypdfsize / false .code:n =
1538 {
1539     \pdf_version_compare:NnTF > {1.6}
1540     {
1541         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1542         { PickTrayByPDFSize } { false }
1543     }
1544     {
1545         \msg_warning:nxxx
1546         {hyp}
1547         {ignore-deprecated-or-unknown-option-in-pdf-version}
1548         {pdfpicktraybypdfsize}
1549         {\pdf_version:}
1550     }
1551 }
1552 ,pdfpicktraybypdfsize / .code:n =
1553 {
1554     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1555 }
1556 ,pdfpicktraybypdfsize / unknown .code:n =
1557 {
1558     \msg_warning:nxxx { hyp } { no-bool }
1559     { picktraybypdfsize }
1560     { \exp_not:n {#1} }
1561 }
1562 ,pdfprintarea .choices:nn =
1563 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1564 {
1565     \pdf_version_compare:NnTF < {2.0}
1566     {

```

```

1567         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1568         { PrintArea } { /#1 }
1569     }
1570     {
1571         \msg_warning:nnxx
1572         {hyp}
1573         {ignore-deprecated-or-unknown-option-in-pdf-version}
1574         {pdfprintarea}
1575         {\pdf_version:}
1576     }
1577 }%
1578 ,pdfprintarea / .code:n =
1579 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1580 ,pdfprintarea / unknown .code:n =
1581 {
1582     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1583     { pdfprintarea }
1584     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1585     { \exp_not:n {#1} }
1586 }
1587 ,pdfprintclip .choices:nn =
1588 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1589 {
1590     \pdf_version_compare:NnTF < {2.0}
1591     {
1592         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1593         { PrintClip } { /#1 }
1594     }
1595     {
1596         \msg_warning:nnxx
1597         {hyp}
1598         {ignore-deprecated-or-unknown-option-in-pdf-version}
1599         {pdfprintclip}
1600         {\pdf_version:}
1601     }
1602 }%
1603 ,pdfprintclip / .code:n =
1604 {
1605     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1606 }
1607 ,pdfprintclip / unknown .code:n =
1608 {
1609     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1610     { pdfprintclip }
1611     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1612     { \exp_not:n {#1} }
1613 }
1614 ,pdfprintpagerange .code:n =
1615 {
1616     \pdf_version_compare:NnTF > {1.6}
1617     {
1618         \tl_if_empty:nTF { #1}
1619         {
1620             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }

```

```

1621         { PrintPageRange }
1622     }
1623     {
1624         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1625         {PrintPageRange}{[#1]}
1626     }
1627 }
1628 {
1629     \msg_warning:nnxx
1630     {hyp}
1631     {ignore-deprecated-or-unknown-option-in-pdf-version}
1632     {pdfprintpagerange}
1633     {\pdf_version:}
1634 }
1635 }
1636 ,pdfprintscaling .choices:nn =
1637 { None, AppDefault }
1638 {
1639     \pdf_version_compare:NnTF > {1.5}
1640     {
1641         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1642         { PrintScaling } { /#1 }
1643     }
1644     {
1645         \msg_warning:nnxx
1646         {hyp}
1647         {ignore-deprecated-or-unknown-option-in-pdf-version}
1648         {pdfprintscaling}
1649         {\pdf_version:}
1650     }
1651 }%
1652 ,pdfprintscaling / .code:n =
1653 {
1654     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1655 }
1656 ,pdfprintscaling / unknown .code:n =
1657 {
1658     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1659     { pdfprintarea }
1660     { None, AppDefault }
1661     { \exp_not:n {#1} }
1662 }
1663 ,pdfremotestartview .code:n =
1664 {
1665     \tl_set:Nx \l__hyp_tmpa_tl {#1-null-null-null~}
1666     \exp_args:NNV
1667     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1668     {
1669         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1670     }
1671     {
1672         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1673         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1674     }

```

```

1675     }
1676 ,pdfremotestartview .initial:n = {Fit}
1677 % pdfstartpage is special as it shares code with pdfstartview
1678 ,pdfstartpage .code:n =
1679 {
1680     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1681     \bool_if:nTF
1682     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1683     {
1684         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1685     }
1686     {
1687         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1688         {
1689             [\pdf_object_pageref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1690         }
1691     }
1692 }
1693 ,pdfstartpage .initial:n =1
1694 ,pdfstartview .code:n =
1695 {
1696     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1697     \exp_args:NNV
1698     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1699     {
1700         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1701     }
1702     {
1703         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1704         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1705     }
1706     \bool_if:nTF
1707     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1708     {
1709         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1710     }
1711     {
1712         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1713         {
1714             [\pdf_object_pageref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1715         }
1716     }
1717 }
1718 ,pdfstartview .initial:n = Fit
1719 ,pdftoolbar .choice:
1720 ,pdftoolbar / true .code:n =
1721 {
1722     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1723 }
1724 ,pdftoolbar / false .code:n =
1725 {
1726     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
1727     { HideToolbar } { true }
1728 }

```

```

1729 ,pdftoolbar / true .code:n =
1730 {
1731     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1732 }
1733 ,pdftoolbar .default:n = true
1734 ,pdftoolbar / unknown .code:n =
1735 {
1736     \msg_warning:nnxx { hyp } { no-bool }
1737     { pdftoolbar }
1738     { \exp_not:n {#1} }
1739 }
1740 % pdfview see below.
1741 ,pdfviewarea .choices:nn =
1742 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1743 {
1744     \pdf_version_compare:NnTF < {2.0}
1745     {
1746         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1747         { ViewArea } { /#1 }
1748     }
1749     {
1750         \msg_warning:nnxx
1751         {hyp}
1752         {ignore-deprecated-or-unknown-option-in-pdf-version}
1753         {pdfviewarea}
1754         {\pdf_version:}
1755     }
1756 }%
1757 ,pdfviewarea / .code:n =
1758 {
1759     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1760 }
1761 ,pdfviewarea / unknown .code:n =
1762 {
1763     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1764     { pdfviewarea }
1765     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1766     { \exp_not:n {#1} }
1767 }
1768 ,pdfviewclip .choices:nn =
1769 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1770 {
1771     \pdf_version_compare:NnTF < {2.0}
1772     {
1773         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1774         { ViewClip } { /#1 }
1775     }
1776     {
1777         \msg_warning:nnxx
1778         {hyp}
1779         {ignore-deprecated-or-unknown-option-in-pdf-version}
1780         {pdfviewclip}
1781         {\pdf_version:}
1782     }

```

```

1783     }%
1784 ,pdfviewclip / .code:n =
1785     {
1786         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
1787     }
1788 ,pdfviewclip / unknown .code:n =
1789     {
1790         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1791         { pdfviewclip }
1792         { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1793         { \exp_not:n {#1} }
1794     }
1795 ,pdfwindowui .choice:
1796 ,pdfwindowui / true .code:n =
1797     {
1798         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
1799     }
1800 ,pdfwindowui / false .code:n =
1801     {
1802         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1803         { HideWindowUI } { true }
1804     }
1805 ,pdfwindowui / .code:n =
1806     {
1807         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
1808     }
1809 ,pdfwindowui / unknown .code:n =
1810     {
1811         \msg_warning:nnxx { hyp } { no-bool }
1812         { pdfwindowui }
1813         { \exp_not:n {#1} }
1814     }
1815 ,pdfwindowui .default:n = true
1816 }

```

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

```

1817 \keys_define:nn { hyp / setup }
1818 {
1819     ,pdfview .code:n =
1820     {
1821         \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
1822         \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } } {
1823             {
1824                 { xyz }
1825                 {
1826                     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
1827                     {
1828                         \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
1829                         \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
1830                         {
1831                             \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1832                         }
1833                     }
1834                     \tl_set:Nx \l__hyp_dest_pdfview_tl

```

```

1835         {
1836             \fp_eval:n { \l__hyp_tmpa_tl * 100 }
1837         }
1838     }
1839 }
1840 {
1841     \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1842 }
1843 }
1844 { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
1845 { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
1846 { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
1847 { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
1848 { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
1849 { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
1850 { fitr }
1851 {
1852     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
1853     {
1854         \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
1855     }
1856     {
1857         %ensure 4 values ...
1858         \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
1859         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1860         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1861         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1862         \hbox_set_to_wd:Nnn \l__hyp_dest_box
1863         {
1864             \fp_eval:n
1865             {
1866                 round
1867                 (
1868                     abs
1869                     (
1870                         \seq_item:Nn\l__hyp_tmpa_seq{4}
1871                         -
1872                         (\seq_item:Nn\l__hyp_tmpa_seq{2})
1873                     ),
1874                     3
1875                 )
1876             }bp
1877         }{ }
1878         \box_set_dp:Nn \l__hyp_dest_box
1879         {
1880             \fp_eval:n
1881             {
1882                 round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
1883             }bp
1884         }
1885         \box_set_ht:Nn \l__hyp_dest_box
1886         {
1887             \seq_item:Nn\l__hyp_tmpa_seq{5}bp
1888         }

```

```

1889         }
1890     }
1891 }
1892 {
1893     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
1894     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
1895 }
1896 }
1897 ,pdfview .initial:n = {xyz}
1898 }

```

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.

```

1899 \keys_define:nn { hyp / setup }
1900 {
1901     ,pdflang .code:n =
1902     {
1903         \tl_if_empty:nTF { #1 }
1904         {
1905             \pdfmanagement_remove:nn {Catalog} { Lang }
1906         }
1907         {
1908             \pdfmanagement_add:nnx {Catalog} { Lang } { (#1) }
1909         }
1910         \__hyp_store_metadata:nn {pdflang}{#1}
1911     }
1912 }

```

12.3.1 “info keys”

`pdfauthor` (*setup key*) The keys store their value also in the metadata container, so that hyperxmp can use them.

```

pdfcreator (setup key) 1913 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
pdfsubject (setup key) 1914 {
pdfproducer (setup key) 1915     \keys_define:nn { hyp / setup }
pdfkeywords (setup key) 1916     {
1917         pdf#1 .code:n =
1918         {
1919             \tl_if_blank:nTF {##1}
1920             {
1921                 \pdfmanagement_remove:nn {Info}{#2}
1922             }
1923             {
1924                 \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
1925                 \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
1926                 {
1927                     \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
1928                 }
1929             }

```



```

1930         \__hyp_store_metadata:nn {pdf#1}{##1}
1931     }
1932 }
1933 \keys_define:nn { hyp / info }
1934 {
1935     #2 .code:n =
1936     {
1937         \tl_if_blank:nTF {##1}
1938         {
1939             \pdfmanagement_remove:nn {Info}{#2}
1940         }
1941         {
1942             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
1943             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
1944             {
1945                 \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
1946             }
1947         }
1948         \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
1949     }
1950 ,unknown .code:n =
1951 {
1952     \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
1953     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
1954     {
1955         \exp_args:Nnx
1956         \pdfmanagement_add:nnx {Info}
1957         {\pdf_text_convert:nV {name } \l_keys_key_str}{\l__hyp_tmpa_str}
1958     }
1959 }
1960 }
1961 }
1962 \__hyp_setup_info_key:nn {author} {Author}
1963 \__hyp_setup_info_key:nn {title} {Title}
1964 \__hyp_setup_info_key:nn {producer} {Producer}
1965 \__hyp_setup_info_key:nn {creator} {Creator}
1966 % ignored key: addtopdfcreator
1967 \__hyp_setup_info_key:nn {subject} {Subject}
1968 \__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.

```

1969 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
1970 {
1971     \keys_define:nn { hyp / setup }
1972     {
1973         pdf#1 .code:n =
1974         {
1975             \tl_if_blank:nTF {##1}
1976             {
1977                 \pdfmanagement_remove:nn {Info}{#2}
1978             }
1979             {
1980                 \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}

```

```

1981     }
1982     \__hyp_store_metadata:nn {pdf#1}{##1}
1983   }
1984 }
1985 \keys_define:nn { hyp / info }
1986 {
1987   #2 .code:n =
1988   {
1989     \tl_if_blank:nTF {##1}
1990     {
1991       \pdfmanagement_remove:nn {Info}{#2}
1992     }
1993     {
1994       \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
1995     }
1996     \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
1997   }
1998 }
1999 }
2000
2001 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2002 \__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 ...

```

2003 \keys_define:nn { hyp / setup }
2004 {
2005   ,pdftrapped .code:n =
2006   {
2007     \exp_args:Nne
2008     \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2009   }
2010   ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2011   {
2012     \pdfmanagement_add:nnx {Info}{Trapped}
2013     {/
2014       \str_uppercase:f { \str_head:n { #1 } }
2015       \str_lowercase:f { \str_tail:n { #1 } }
2016     }
2017     \__hyp_store_metadata:nx {pdftrapped}
2018     {
2019       \str_uppercase:f { \str_head:n { #1 } }
2020       \str_lowercase:f { \str_tail:n { #1 } }
2021     }
2022   }
2023   ,_pdftrapped / unknown .code:n =
2024   {
2025     \msg_warning:nnxxx { hyp } { unknown-choice }
2026     { pdftrapped }
2027     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2028     { \exp_not:n {#1} }
2029   }
2030 }

```

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

```

2031 \keys_define:nn { hyp / setup }
2032 {
2033     pdfinfo .code:n =
2034     {
2035         \keys_set:nn { hyp / info } { #1 }
2036     }
2037 }

```

Now we set some default values

```

2038 \keys_set:nn { hyp / setup} {pdfcreator = LaTeX-with-hyperref}
2039 \keys_set:nn { hyp / setup} {pdfauthor  = }
2040 \keys_set:nn { hyp / setup} {pdftitle   = }
2041 \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:

```

2042 \clist_map_inline:nn
2043 {
2044     ,pdfcopyright
2045     ,pdftype
2046     ,pdflicenseurl
2047     ,pdfauthortitle
2048     ,pdfcaptionwriter
2049     ,pdfmetalang
2050     ,pdfapart
2051     ,pdfaconformance
2052     ,pdfuapart
2053     ,pdfxstandard
2054     ,pdfsource
2055     ,pdfdocumentid
2056     ,pdfinstanceid
2057     ,pdfversionid
2058     ,pdfrendition
2059     ,pdfpublication
2060     ,pdfpubtype
2061     ,pdfbytes
2062     ,pdfnumpages
2063     ,pdfissn
2064     ,pdfeissn
2065     ,pdfisbn
2066     ,pdfbookedition
2067     ,pdfpublisher
2068     ,pdfvolumenum
2069     ,pdfissuenum
2070     ,pdfpagerange
2071     ,pdfdoi
2072     ,pdfurl
2073     ,pdfidentifier
2074     ,pdfsubtitle
2075     ,pdfpubstatus
2076     ,pdfcontactaddress

```

```

2077 ,pdfcontactcity
2078 ,pdfcontactregion
2079 ,pdfcontactpostcode
2080 ,pdfcontactcountry
2081 ,pdfcontactphone
2082 ,pdfcontactemail
2083 ,pdfcontacturl
2084 ,pdfdate
2085 }
2086 {
2087   \keys_define:nn { hyp / setup }
2088   {
2089     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2090   }
2091 }

```

12.5 Transitions

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

```

2092 \keys_define:nn { hyp / setup }
2093 {
2094   pdfpageduration .code:n =
2095   {
2096     \tl_if_blank:nTF { #1 }
2097     {
2098       \pdfmanagement_remove:nn {Page}{Dur}
2099     }
2100     {
2101       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2102     }
2103   }
2104 }

```

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

The keys allowed in the argument of `pdfpagetransition` are

```

2105 \keys_define:nn { hyp / setup }
2106 {
2107   pdfpagetransition .code:n =
2108   {
2109     \tl_if_blank:nTF {#1}
2110     {
2111       \pdfmanagement_remove:nn {Page}{Trans}
2112     }
2113     {
2114       \group_begin:
2115       \keys_set:nn { hyp / trans }{style=R,#1}
2116       \pdf_object_now:nx { dict }
2117       {
2118         \pdfdict_use:n {l__hyp_page/Trans}
2119       }
2120       \pdfmanagement_add:nnx {Page}{Trans}{\pdf_object_last:}
2121       \group_end:
2122     }
2123   }
2124 }
2125 \keys_define:nn { hyp / trans }
2126 {
2127   ,style .choices:nn =
2128   {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2129   { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2130   ,style / unknown .code:n =
2131   {
2132     \msg_warning:nnxxx { hyp } { unknown-choice }
2133     { trans / style }
2134     { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2135     { \exp_not:n {#1} }
2136   }
2137   ,duration .code:n =
2138   {
2139     \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}

```

```

2140     }
2141     ,direction .choices:nn =
2142     {H,V}
2143     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2144     ,direction .choices:nn =
2145     {0,90,180,270,315}
2146     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ #1 } }
2147     ,direction / None .code:n =
2148     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ /None } }
2149     ,direction / unknown .code:n =
2150     {
2151         \msg_warning:nnxxx { hyp } { unknown-choice }
2152         { trans / direction }
2153         {
2154             H~(horizontal,~only~Split,~Blinds),
2155             V~(vertical,~only~Split,~Blinds),
2156             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2157             90~(bottom~to~top,~only~Wipe),
2158             180~(right~to~left,~only~Wipe),
2159             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2160             315~(top~left~to~bottom,~only~Glitter),
2161             None~(only~Fly)
2162         }
2163         { \exp_not:n {#1} }
2164     }
2165     ,motion .choices:nn =
2166     {I,0}
2167     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2168     ,motion / unknown .code:n =
2169     {
2170         \msg_warning:nnxxx { hyp } { unknown-choice }
2171         { trans / motion }
2172         { I~(inwards) , 0~(outwards) }
2173         { \exp_not:n {#1} }
2174     }
2175     ,scale .code:n =
2176     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2177     ,opaque .choices:nn = {true,false}
2178     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2179     ,opaque / unknown .code:n =
2180     {
2181         \msg_warning:nnxxx { hyp } { unknown-choice }
2182         { trans / B }
2183         { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2184         { \exp_not:n {#1} }
2185     }
2186     % try to set unknown keys as style
2187     ,unknown .code:n =
2188     {
2189         % warning ...
2190         \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2191     }
2192 }

```

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

```

2193 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}
    Unfinished Form field code
2194 \NewDocumentCommand \MakeFieldObject { m m }
2195 {
2196   \pdf_xform_new:nnn { #2 }{} { #1 }
2197 }
2198
2199
2200 \prop_new:N    \g__hyp_AcroForm_CoFields_prop
2201 \prop_new:N    \g__hyp_AcroForm_Fields_prop
2202
2203 \let\HyField@afields\ltx@empty
2204 \let\HyField@cofields\ltx@empty
2205 %% UF test for old pdftex removed
2206 %\let\HyField@AuxAddToFields\ltx@gobble
2207 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2208 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2209
2210 % the value doesn't matter, but with a prop we avoid duplicates and it is
2211 % clearly faster than removing them from a sequence
2212 \def\HyField@AuxAddToFields#1
2213 {
2214   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2215 }%
2216
2217 %fields with empty key get a value too -- lets hope that
2218 %this give the expected behaviour
2219 \def\HyField@AuxAddToCoFields #1 #2
2220 {
2221   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2222 }
2223
2224 \Hy@AtBeginDocument
2225 {
2226   \if@filesw
2227     \immediate\write\@mainaux{%
2228       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2229     }%
2230     \immediate\write\@mainaux{%
2231       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2232     }%
2233   \fi
2234   \let\HyField@AfterAuxOpen\@firstofone
2235 }%
2236
2237 \def\HyField@AddToFields
2238 {
2239   \exp_args:Nx\HyField__hypAddToFields
2240   {
2241     %\pdf_link_last:
2242     \pdfannot_box_last:
2243   }

```

```

2244 \ifx\Fld@calculate@code\ltx@empty
2245 \else
2246 \begingroup
2247 \Hy@safe@activetrue
2248 \edef\Hy@temp{%
2249 \endgroup
2250 \if@filesw
2251 \write\@mainaux
2252 {
2253 \string\HyField@AuxAddToCoFields
2254 {
2255 \Fld@calculate@sortkey
2256 }
2257 {
2258 % \pdf_link_last:
2259 \pdfannot_box_last:
2260 }
2261 }
2262 \fi
2263 }%
2264 \Hy@temp
2265 \fi
2266 }%
2267
2268 \def\HyField__hypAddToFields#1{
2269 \HyField@AfterAuxOpen{%
2270 \if@filesw
2271 \write\@mainaux{%
2272 \string\HyField@AuxAddToFields{#1}%
2273 }%
2274 \fi
2275 }%
2276 }%
2277
2278 \ExplSyntaxOff
2279 \ExplSyntaxOn
2280 \tl_new:N \l__hyp_CheckmarkYes_tl
2281 \tl_set:Nn \l__hyp_CheckmarkYes_tl { __hyp_xform_CheckMarkYes }
2282 \tl_new:N \l__hyp_CheckmarkOff_tl
2283 \tl_set:Nn \l__hyp_CheckmarkOff_tl { __hyp_xform_CheckMarkOff }
2284
2285 \def\@Form[#1]
2286 {
2287 \@ifundefined{textcolor}{\let\textcolor\@gobble}{ }
2288 \kvsetkeys{Form}{#1}
2289 \pdf@ifdraftmode{
2290 {
2291 \Hy@FormObjects
2292 \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2293 {
2294 \pdfmanagement_add:nx { Catalog / AcroForm } { Fields }{##1}
2295 %\pdfmanagement_show:n { Catalog / AcroForm }
2296 }
2297 \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop

```



```

2298     {
2299         \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2300         {
2301             \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2302         }
2303         \seq_sort:Nn \l__hyp_tmpa_seq
2304         {
2305             \int_compare:nNtF { \pdf@strcmp{##1}{##2} } > { 0 }
2306             { \sort_return_swapped: }
2307             { \sort_return_same: }
2308         }
2309         \seq_map_inline:Nn \l__hyp_tmpa_seq
2310         {
2311             \pdfmanagement_add:nnx { Catalog / AcroForm }
2312             { CO }
2313             {
2314                 \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2315             }
2316         }
2317     }
2318     \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2319     {ZaDb} {\pdf_object_ref:n {l__hyp_font_zapdingbats_obj} }
2320     \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2321     {Helv} {\pdf_object_ref:n {l__hyp_font_helvetica_obj} }
2322     \pdfmanagement_add:nnx {Catalog /AcroForm}
2323     {DA}{(/Helv~10~Tf~0~g)}
2324     \prop_if_in:NnTF\g_pdfmeta_standard_prop {form_no_NeedAppearances}
2325     {
2326         \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2327     }
2328     {
2329         \legacy_if:nT { HyField@NeedAppearances }
2330         {
2331             \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2332         }
2333     }
2334 }
2335 \MakeFieldObject
2336 {
2337     \group_begin:
2338     \fontfamily{pzd}
2339     \fontencoding{U}
2340     \fontseries{m}
2341     \fontshape{n}
2342     \selectfont
2343     \char123
2344     \group_end:
2345 }
2346 {__hyp_xform_Ding}
2347 \MakeFieldObject
2348 {
2349     \group_begin:
2350     \fontfamily{pzd}
2351     \fontencoding{U}

```

```

2352         \fontseries{m}
2353         \fontshape{n}
2354         \selectfont
2355         \phantom{\char123}
2356         \group_end:
2357     }
2358     {__hyp_xform_DingOff}
2359 \MakeFieldObject
2360 {
2361     \group_begin:
2362     \fontfamily{pzd}
2363     \fontencoding{U}
2364     \fontseries{m}
2365     \fontshape{n}
2366     \selectfont
2367     \char51
2368     \group_end:
2369 }
2370 {__hyp_xform_CheckMarkYes}
2371 \MakeFieldObject
2372 {
2373     \group_begin:
2374     \fontfamily{pzd}
2375     \fontencoding{U}
2376     \fontseries{m}
2377     \fontshape{n}
2378     \selectfont
2379     \phantom{\char51} %perhaps xetex needs some small glyph ..
2380     \group_end:
2381 }
2382 {__hyp_xform_CheckMarkOff}
2383 \MakeFieldObject
2384 {
2385     \fbox{\textcolor{yellow}{\textsf{Submit}}}% %color?
2386 }
2387 {__hyp_xform_Submit}
2388 \MakeFieldObject
2389 {
2390     \fbox{\textcolor{yellow}{\textsf{SubmitP}}}% %color?
2391 }
2392 {__hyp_xform_SubmitP}
2393 }
2394 \ExplSyntaxOff
2395 \let\@endForm\ltx@empty
2396 \let\HyAnn@AbsPageLabel\ltx@empty
2397 \let\Fld@pageobjref\ltx@empty
2398
2399 \ExplSyntaxOn
2400 \newcount\HyAnn@Count
2401 \HyAnn@Count=\ltx@zero
2402 \def\HyAnn@AbsPageLabel
2403 {
2404     \global\advance\HyAnn@Count by\ltx@one
2405     %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%

```

```

2406     %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2407     %\zref@refused{HyAnn@\the\HyAnn@Count}%
2408     \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
2409     \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
2410 }%
2411 \def\Fld@pageobjref
2412 {
2413     \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
2414     {
2415         /P~\pdf_object_pageref:n
2416         {
2417             \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
2418         }
2419     }
2420 }
2421 \ExplSyntaxOff
2422 \ExplSyntaxOn
2423 %% check if the attr should be set through
2424 %% hooks.
2425 %% check if options are missing.
2426 \def\@TextField[#1]#2{% parameters, label
2427     \def\Fld@name{#2}%
2428     \let\Fld@default\ltx@empty
2429     \let\Fld@value\@empty
2430     \def\Fld@width{\DefaultWidthofText}%
2431     \def\Fld@height{%
2432         \ifFld@multiline
2433             \DefaultHeightofTextMultiline
2434         \else
2435             \DefaultHeightofText
2436         \fi
2437     }%
2438     \begingroup
2439     \expandafter\HyField@SetKeys\expandafter{%
2440         \DefaultOptionsofText,#1%
2441     }%
2442     \PDFForm@Name
2443     \HyField@FlagsText
2444     \ifFld@hidden\def\Fld@width{1sp}\fi
2445     \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2446     \LayoutTextField{#2}{%
2447         \leavevmode
2448         \HyAnn@AbsPageLabel
2449         \Hy@escapeform\PDFForm@Text
2450         \pdfannot_box:nnnn
2451         {\Fld@width}
2452         {\Fld@height}
2453         {0pt} %is this correct?
2454         {\PDFForm@Text}
2455         \MakeTextField{\Fld@width}{\Fld@height}
2456         \HyField@AddToFields
2457     }%
2458 \endgroup
2459 }

```

```

2460 \providecommand\@curropt{}
2461 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2462   \def\Fld@name{#2}
2463   \let\Fld@default\relax
2464   \let\Fld@value\relax
2465   \def\Fld@width{\DefaultWidthofChoiceMenu}
2466   \def\Fld@height{\DefaultHeightofChoiceMenu}
2467   \beginingroup
2468     \Fld@menulength=0 %
2469     \@tempdima\z@
2470     \clist_map_variable:nNn { #3 } \@curropt
2471     %\@for\@curropt:=#3\do
2472     {%
2473       \expandafter\Fld@checkequals\@curropt==\%
2474       \Hy@StepCount\Fld@menulength
2475       \settowidth{\@tempdimb}{\@currDisplay}%
2476       \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2477     }%
2478     \advance\@tempdima by~15\p@
2479     \beginingroup
2480       \HyField@SetKeys{#1}
2481     \edef\x{\endgroup
2482       \noexpand\expandafter
2483       \noexpand\HyField@SetKeys
2484       \noexpand\expandafter{%
2485         \expandafter\noexpand\csname DefaultOptionsof%
2486           \ifFld@radio
2487             Radio%
2488           \else
2489             \ifFld@combo
2490               \ifFld@popdown
2491                 PopdownBox%
2492               \else
2493                 ComboBox%
2494               \fi
2495             \else
2496               ListBox%
2497             \fi
2498           \fi
2499         \endcsname
2500       }%
2501     }\x
2502     \HyField@SetKeys{#1}%
2503     \PDFForm@Name
2504     \ifFld@hidden\def\Fld@width{1sp}\fi
2505     \ifx\Fld@value\relax
2506       \let\Fld@value\Fld@default
2507     \fi
2508     \LayoutChoiceField{#2}{%
2509       \ifFld@radio
2510         \HyField@FlagsRadioButton
2511         \__hypRadio{#3}%
2512       \else
2513         \beginingroup

```

```

2514         \HyField@FlagsChoice
2515         \ifdim\Fld@width<\@tempdima
2516             \ifdim\@tempdima<1cm\@tempdima1cm\fi
2517             \edef\Fld@width{\the\@tempdima}%
2518         \fi
2519         \ifFld@combo
2520         \else
2521             \@tempdima=\the\Fld@menulength\Fld@charsize
2522             \advance\@tempdima by~\Fld@borderwidth bp %
2523             \advance\@tempdima by~\Fld@borderwidth bp %
2524             \edef\Fld@height{\the\@tempdima}%
2525         \fi
2526         \__hypListbox{#3}%
2527     \endgroup
2528 \fi
2529 }%
2530 \endgroup
2531 }
2532 \tl_new:N \l__hyp_RadioYes_tl
2533 \tl_set:Nn \l__hyp_RadioYes_tl { __hyp_xform_Ding }
2534 \def\__hypRadio#1{%
2535     \Fld@listcount=0-%
2536     %\show\Fld@default
2537     \EdefEscapeName\Fld@default{\Fld@default}%
2538     \clist_map_variable:nNn { #1 } \@curropt
2539     %\@for\@curropt:=#1\do
2540     {%
2541         \expandafter\Fld@checkequals\@curropt==\%
2542         \EdefEscapeName\@currValue{\@currValue}%
2543         \Hy@StepCount\Fld@listcount
2544         \@currDisplay\space
2545         \leavevmode
2546         \HyAnn@AbsPageLabel
2547         \Hy@escapeform\PDFForm@Radio
2548         \pdfannot_box:nnnn
2549             {\Fld@width}
2550             {\Fld@height}
2551             {Opt} %is this correct?
2552         {
2553             \PDFForm@Radio
2554             /AP
2555             <<
2556             /N
2557             <<
2558             /\@currValue\c_space_tl \pdf_xform_ref:o {__hyp_xform_Ding}
2559             %/Off \c_space_tl \pdf_xform_ref:n {__hyp_xform_DingOff} %hm
2560             >>
2561             >>
2562         }
2563         {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2564         \int_compare:nNt { \Fld@listcount } = { 1 }
2565         { \HyField@AddToFields }
2566         \c_space_tl % deliberate space between radio buttons
2567         % to do: --> should be configurable

```

```

2568 }%
2569 }
2570
2571 \newcount\Fld@listcount
2572 \def\__hypListbox#1
2573 {
2574   \HyField@PDFChoices{#1}
2575   \mode_leave_vertical:
2576   \HyAnn@AbsPageLabel
2577   \Hy@escapeform\PDFForm@List
2578   \pdf_link_user:nnn
2579   {widget} %perhaps we need more types??
2580   {\PDFForm@List}
2581   {\MakeChoiceField{\Fld@width}{\Fld@height}}
2582   \HyField@AddToFields
2583 }
2584
2585
2586 \def\@PushButton[#1]#2{% parameters, label
2587   \def\Fld@name{#2}%
2588   \group_begin:
2589     \exp_args:No\HyField@SetKeys
2590     {
2591       \DefaultOptionsofPushButton,#1
2592     }
2593     \PDFForm@Name
2594     \prop_if_in:NnTF \g_pdfmeta_standard_prop {annot_action_no_A}
2595     {
2596       \msg_error:nn { hyp }{ pdfa-no-push-button }
2597       \LayoutPushButtonField
2598       {
2599         \mode_leave_vertical:
2600         \MakeButtonField{#2}
2601       }
2602     }
2603     {
2604       \HyField@FlagsPushButton
2605       \legacy_if:nT {Fld@hidden}
2606       {
2607         \def\Fld@width{1sp}
2608       }
2609       \LayoutPushButtonField
2610       {
2611         \mode_leave_vertical:
2612         \HyAnn@AbsPageLabel
2613         \Hy@escapeform\PDFForm@Push
2614         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2615         \pdfannot_box:nnnn
2616         {\box_wd:N\l_tmpa_box}
2617         {\box_ht:N\l_tmpa_box}
2618         {\box_dp:N\l_tmpa_box} %is this correct?
2619         {\PDFForm@Push}
2620         {\box_use:N\l_tmpa_box}
2621         \HyField@AddToFields

```

```

2622     }
2623 }
2624 \group_end:
2625 }
2626
2627 \def\@Submit[#1]#2
2628 {
2629   \def\Fld@width {\DefaultWidthofSubmit}
2630   \def\Fld@height{\DefaultHeightofSubmit}
2631   \group_begin:
2632     \exp_args:No\HyField@SetKeys
2633     {
2634       \DefaultOptionsofSubmit,#1
2635     }
2636     \HyField@FlagsPushButton
2637     \HyField@FlagsSubmit
2638     \legacy_if:nT { Fld@hidden }
2639     {
2640       \def\Fld@width{1sp}
2641     }
2642     \mode_leave_vertical:
2643     \HyAnn@AbsPageLabel
2644     \Hy@escapeform\PDFForm@Submit
2645     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2646     \pdfannot_box:nnnn
2647     {\box_wd:N\l_tmpa_box}
2648     {\box_ht:N\l_tmpa_box}
2649     {\box_dp:N\l_tmpa_box} %is this correct?
2650     {
2651       \PDFForm@Submit
2652       /AP<<
2653         /N~\pdf_xform_ref:n {__hyp_xform_Submit}~
2654         /D~\pdf_xform_ref:n {__hyp_xform_SubmitP}
2655       >>
2656     }
2657     \HyField@AddToFields
2658     \box_use:N\l_tmpa_box
2659
2660   \group_end:
2661 }
2662
2663 \def\@Reset[#1]#2
2664 {
2665   \def\Fld@width {\DefaultWidthofReset}
2666   \def\Fld@height{\DefaultHeightofReset}
2667   \group_begin:
2668     \exp_args:No\HyField@SetKeys
2669     {
2670       \DefaultOptionsofReset,#1
2671     }
2672     \mode_leave_vertical:
2673     \prop_if_in:NnTF \g_pdfmeta_standard_prop {annot_action_no_A}
2674     {
2675       \msg_error:nn { hyp }{ pdfa-no-reset-button }

```

```

2676     \MakeButtonField{#2}
2677   }
2678   {
2679     \HyField@FlagsPushButton
2680     \legacy_if:nT { Fld@hidden }
2681     { \def\Fld@width{1sp} }
2682     \HyAnn@AbsPageLabel
2683     \Hy@escapeform\PDFForm@Reset
2684     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2685     \pdfannot_box:nnnn
2686     {\box_wd:N\l_tmpa_box}
2687     {\box_ht:N\l_tmpa_box}
2688     {\box_dp:N\l_tmpa_box} %is this correct?
2689     { \PDFForm@Reset }
2690     \HyField@AddToFields
2691     \box_use:N \l_tmpa_box
2692   }
2693   \group_end:
2694 }
2695
2696 \def\@CheckBox[#1]#2
2697 {% parameters, label
2698   \def\Fld@name{#2}
2699   \def\Fld@default{0}
2700   \group_begin:
2701     \def\Fld@width {\DefaultWidthofCheckBox}
2702     \def\Fld@height{\DefaultHeightofCheckBox}
2703     \exp_args:No\HyField@SetKeys
2704     {
2705       \DefaultOptionsofCheckBox,#1
2706     }
2707     \PDFForm@Name
2708     \HyField@FlagsCheckBox
2709     \legacy_if:nT { Fld@hidden }
2710     {
2711       \def\Fld@width{1sp}
2712     }
2713     \LayoutCheckField{#2}
2714     {
2715       \mode_leave_vertical:
2716       \HyAnn@AbsPageLabel
2717       \Hy@escapeform\PDFForm@Check
2718       \pdfannot_box:nnnn
2719       {\Fld@width}
2720       {\Fld@height}
2721       {Opt} %is this correct?
2722       {\PDFForm@Check}
2723       \HyField@AddToFields %check if this works with xelatex ...
2724     }
2725     \group_end:
2726   }
2727 \ExplSyntaxOff
2728
2729 %hm. Should a luatex driver use type1 fonts in fields????

```



```

2730 \ExplSyntaxOn
2731 \def\Hy@FormObjects
2732 {
2733   \pdf_object_new:nn {l__hyp_encoding_pdfdoc_obj } { dict }
2734   \pdf_object_new:nn {l__hyp_font_zapfdingbats_obj } { dict }
2735   \pdf_object_new:nn {l__hyp_font_helvetica_obj } { dict }
2736   \pdf_object_write:nx {l__hyp_encoding_pdfdoc_obj }
2737   {
2738     /Type/Encoding
2739     /Differences[
2740       24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2741       /ring/tilde
2742       \c_space_tl
2743       39/quotesingle
2744       \c_space_tl
2745       96/grave %
2746       \iow_newline:
2747       128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2748       /fraction/guilsinglleft/guilsinglright/minus/perthousand
2749       /quotedblbase/quotedblleft/quotedblright/quoteleft
2750       /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2751       /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2752       \iow_newline:
2753       164/currency
2754       \c_space_tl
2755       166/brokenbar
2756       \c_space_tl
2757       168/dieresis/copyright/ordfeminine
2758       \c_space_tl
2759       172/logicalnot/.notdef/registered/macron/degree/plusminus
2760       /twosuperior/threesuperior/acute/mu
2761       \c_space_tl
2762       183/periodcentered/cedilla/onesuperior/ordmasculine
2763       \c_space_tl
2764       188/onequarter/onehalf/threequarters
2765       \iow_newline:
2766       192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
2767       /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
2768       /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
2769       /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
2770       /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
2771       /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
2772       /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
2773       /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
2774       /ocircumflex/otilde/odieresis/divide/oslash/ugrave
2775       /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
2776     ]
2777   }
2778   \pdf_object_write:nn {l__hyp_font_zapfdingbats_obj }
2779   {
2780     /Type/Font
2781     /Subtype/Type1
2782     /Name/ZaDb
2783     /BaseFont/ZapfDingbats

```

```

2784     }
2785     \pdf_object_write:nx {l__hyp_font_helvetica_obj }
2786     {
2787         /Type/Font
2788         /Subtype/Type1
2789         /Name/Helv
2790         /BaseFont/Helvetica
2791         /Encoding~\pdf_object_ref:n { l__hyp_encoding_pdfdoc_obj }
2792     }
2793     \global\let\Hy@FormObjects\relax
2794 }
2795 \ExplSyntaxOff
2796 \providecommand*{\Fld@pageobjref}{ }
2797 \ifcsname pdf@escapestring\endcsname
2798     \def\Hy@escapeform#1{%
2799         \ifHy@pdfescapeform
2800             \let\Hy@escapestring\pdfescapestring
2801         \else
2802             \let\Hy@escapestring\@firstofone
2803         \fi
2804     }%
2805     \Hy@escapeform{}%
2806 \else
2807     \let\Hy@escapestring\@firstofone
2808     \def\Hy@escapeform#1{%
2809         \ifHy@pdfescapeform
2810             \def\Hy@escapestring##1{%
2811                 \noexpand\Hy@escapestring{\noexpand##1}%
2812             }%
2813             \edef\Hy@temp{#1}%
2814             \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{}\@nil
2815             \def\Hy@escapestring##1{%
2816                 \@ifundefined{Hy@esc@\string##1}{%
2817                     ##1%
2818                     \ThisShouldNotHappen
2819                 }{%
2820                     \csname Hy@esc@\string##1\endcsname
2821                 }%
2822             }%
2823         \else
2824             \let\Hy@escapestring\@firstofone
2825         \fi
2826     }%
2827     \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
2828         \ifx\#3\%
2829         \else
2830             \expandafter
2831             \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
2832             \ltx@ReturnAfterFi{%
2833                 \Hy__hypescapeform#3\@nil
2834             }%
2835         \fi
2836     }%
2837 \fi

```

```

2838 \def\PDFForm@Name{%
2839 \PDFForm__hypName\Fld@name
2840 \ifx\Fld@altname\relax
2841 \else
2842 \PDFForm__hypName\Fld@altname
2843 \fi
2844 \ifx\Fld@mappingname\relax
2845 \else
2846 \PDFForm__hypName\Fld@mappingname
2847 \fi
2848 }
2849 \def\PDFForm__hypName#1{%
2850 \begingroup
2851 \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
2852 \ifHy@unicode
2853 \Hy@unicodedefalse
2854 \fi
2855 \fi
2856 \pdfstringdef\Hy@gtemp#1%
2857 \endgroup
2858 \let#1\Hy@gtemp
2859 }
2860 \def\Fld@X@additionalactions{%
2861 \ifx\Fld@keystroke@code\@empty
2862 \else
2863 /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
2864 \fi
2865 \ifx\Fld@format@code\@empty
2866 \else
2867 /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
2868 \fi
2869 \ifx\Fld@validate@code\@empty
2870 \else
2871 /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
2872 \fi
2873 \ifx\Fld@calculate@code\@empty
2874 \else
2875 /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
2876 \fi
2877 \ifx\Fld@onfocus@code\@empty
2878 \else
2879 /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
2880 \fi
2881 \ifx\Fld@onblur@code\@empty
2882 \else
2883 /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
2884 \fi
2885 \ifx\Fld@onmousedown@code\@empty
2886 \else
2887 /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
2888 \fi
2889 \ifx\Fld@onmouseup@code\@empty
2890 \else
2891 /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%

```

```

2892 \fi
2893 \ifx\Fld@onenter@code\@empty
2894 \else
2895   /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
2896 \fi
2897 \ifx\Fld@onexit@code\@empty
2898 \else
2899   /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
2900 \fi
2901 }
2902 \ExplSyntaxOn
2903 \def\Fld@additionalactions
2904 {%
2905   \exp_args:N\str_if_eq:nnF {\Fld@X@additionalactions}{}
2906   {
2907     \prop_if_in:NnF \g_pdfmeta_standard_prop {annot_widget_no_AA}
2908     {/AA<<\Fld@X@additionalactions>>}}
2909   }
2910 }
2911 \ExplSyntaxOff
2912 \def\Fld@annotnames{%
2913   /T(\Fld@name)%
2914   \ifx\Fld@altname\relax
2915   \else
2916     /TU(\Fld@altname)%
2917   \fi
2918   \ifx\Fld@mappingname\relax
2919   \else
2920     /TM(\Fld@mappingname)%
2921   \fi
2922 }
2923 \ExplSyntaxOn
2924 \def\PDFForm@Check
2925 {
2926   /Subtype/Widget
2927   ~\Fld@annotflags
2928   ~\Fld@pageobjref
2929   ~\Fld@annotnames
2930   /FT/Btn
2931   \Fld@flags
2932   /Q~\Fld@align
2933   /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
2934   /AP
2935   <<
2936     /N
2937     <<
2938       /Yes~\pdf_xform_ref:o{\l_hyp_CheckmarkYes_tl}
2939       /Off~\pdf_xform_ref:o{\l_hyp_CheckmarkOff_tl}
2940     >>
2941   >>
2942   /MK<<
2943     \int_compare:nNnF {\Fld@rotation}={0}
2944     {
2945       /R~\Fld@rotation

```

```

2946     }
2947     \tl_if_empty:NF\Fld@bordercolor
2948     {
2949         /BC[\Fld@bordercolor]
2950     }
2951     \tl_if_empty:NF\Fld@bcolor
2952     {
2953         /BG[\Fld@bcolor]
2954     }
2955     /CA(\Hy@escapestring{\Fld@cbsymbol})%
2956 >>
2957 /DA
2958 (
2959     /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
2960     \tl_if_empty:NF \Fld@color
2961     {
2962         \c_space_tl \Fld@color
2963     }
2964 )
2965 /H/P
2966 \legacy_if:nTF {Fld@checked}
2967 {
2968     /V/Yes /AS/Yes
2969 }
2970 {
2971     /V/Off /AS/Off
2972 }
2973 \Fld@additionalactions
2974 }
2975 \ExplSyntaxOff
2976 \ExplSyntaxOn
2977 \prop_if_in:NnF \g_pdfmeta_standard_prop {annot_action_no_A}
2978 {
2979     \def\PDFForm@Push
2980     {
2981         /Subtype/Widget
2982         ~\Fld@annotflags
2983         ~\Fld@pageobjref
2984         ~\Fld@annotnames
2985         /FT/Btn
2986         ~\Fld@flags
2987         /H/P
2988         /BS<</W~\Fld@borderwidth/S\Fld@borderstyle>>
2989         \bool_if:nT
2990         {
2991             !\int_compare_p:nNn {\Fld@rotation} = {0}
2992             ||
2993             \tl_if_exist_p:N \Fld@bordercolor
2994         }
2995         {
2996             /MK
2997             <<
2998             \int_compare:nNnF {\Fld@rotation} = {0}
2999             {

```

```

3000         /R~\Fld@rotation
3001     }
3002     \tl_if_exist:NT \Fld@bordercolor
3003     {
3004         /BC[\Fld@bordercolor]
3005     }
3006     >>
3007 }
3008 /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3009 \Fld@additionalactions
3010 }
3011 }
3012 \ExplSyntaxOff
3013 \def\PDFForm@List{%
3014     /Subtype/Widget%
3015     \Fld@annotflags
3016     \Fld@pageobjref
3017     \Fld@annotnames
3018     /FT/Ch%
3019     \Fld@flags
3020     /Q \Fld@align
3021     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3022     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3023         \ifx\Fld@bordercolor\relax\else 1\fi
3024         \ifx\Fld@bcolor\relax \else 1\fi
3025         \space
3026     \else
3027         /MK<<%
3028         \ifnum\Fld@rotation=\z@
3029         \else
3030             /R \Fld@rotation
3031         \fi
3032         \ifx\Fld@bordercolor\relax
3033         \else
3034             /BC[\Fld@bordercolor]%
3035         \fi
3036         \ifx\Fld@bcolor\relax
3037         \else
3038             /BG[\Fld@bcolor]%
3039         \fi
3040         >>%
3041     \fi
3042     /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3043         \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3044     \Fld@choices
3045     \Fld@additionalactions
3046 }
3047 \ExplSyntaxOn
3048 \def\PDFForm@Radio
3049 {
3050     /Subtype/Widget
3051     ~\Fld@annotflags
3052     ~\Fld@pageobjref
3053     ~\Fld@annotnames

```

```

3054     /FT/Btn
3055     \Fld@flags
3056     /H/P
3057     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3058     /MK<<
3059     \ifnum\Fld@rotation=\z@
3060     \else
3061     /R~\Fld@rotation
3062     \fi
3063     \ifx\Fld@bordercolor\relax
3064     \else
3065     /BC[\Fld@bordercolor]%
3066     \fi
3067     \ifx\Fld@bcolor\relax
3068     \else
3069     /BG[\Fld@bcolor]%
3070     \fi
3071     /CA(\Hy\escapestring{\Fld@radiosymbol})%
3072     >>
3073     /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3074     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3075     \ifx\Fld@default\@empty
3076     /V/Off%
3077     /DV/Off%
3078     \else
3079     /V/\Fld@default
3080     /DV/\Fld@default
3081     \fi
3082     \Fld@additionalactions
3083   }
3084   \ExplSyntaxOff
3085   \ExplSyntaxOn
3086   % Does an appeareance dict make sense here?
3087   \def\PDFForm@Text
3088   {
3089     /Subtype/Widget
3090     ~\Fld@annotflags
3091     ~\Fld@pageobjref
3092     ~\Fld@annotnames
3093     /FT/Tx
3094     ~\Fld@flags
3095     /Q~\Fld@align
3096     /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3097     \bool_if:nT
3098     {
3099       !\int_compare_p:nNn {\Fld@rotation} = {0}
3100       ||
3101       \tl_if_exist_p:N \Fld@bordercolor
3102       ||
3103       \tl_if_exist_p:N \Fld@bcolor
3104     }
3105     {
3106       /MK
3107       <<

```

```

3108         \int_compare:nNnF {\Fld@rotation} = {0}
3109         {
3110             /R~\Fld@rotation
3111         }
3112         \tl_if_exist:NT \Fld@bordercolor
3113         {
3114             /BC[\Fld@bordercolor]
3115         }
3116         \tl_if_exist:NT \Fld@bcolor
3117         {
3118             /BG[\Fld@bcolor]
3119         }
3120     >>
3121 }
3122 /DA
3123 (
3124     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3125     \tl_if_empty:NF {\c_space_tl\Fld@color}
3126 )
3127 /DV(\Hy@escapestring{\Fld@default})
3128 /V(\Hy@escapestring{\Fld@value})
3129 ~\Fld@additionalactions
3130 \int_compare:nNnT { \Fld@maxlen}>{0}
3131 {
3132     /MaxLen~\Fld@maxlen
3133 }
3134 }
3135 \ExplSyntaxOff
3136
3137 \def\PDFForm@Submit{%
3138     /Subtype/Widget%
3139     \Fld@annotflags
3140     \Fld@pageobjref
3141     \Fld@annotnames
3142     /FT/Btn%
3143     \Fld@flags
3144     /H/P%
3145     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3146     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3147         \ifx\Fld@bordercolor\relax\else 1\fi
3148         \space
3149     \else
3150         /MK<<%
3151         \ifnum\Fld@rotation=\z@
3152         \else
3153             /R \Fld@rotation
3154         \fi
3155         \ifx\Fld@bordercolor\relax
3156         \else
3157             /BC[\Fld@bordercolor]%
3158         \fi
3159     >>%
3160     \fi
3161     /A<<%

```



```

3162     /S/SubmitForm%
3163     /F<<%
3164     /FS/URL%
3165     /F(\Hy@escapestring{\Form@action})%
3166     >>%
3167     \Fld@submitflags
3168     >>%
3169     \Fld@additionalactions
3170 }
3171 \ExplSyntaxOn
3172 \prop_if_in:NnF \g_pdfmeta_standard_prop {annot_action_no_A}
3173 {
3174     \def\PDFForm@Reset{%
3175         /Subtype/Widget%
3176         \Fld@annotflags
3177         \Fld@pageobjref
3178         \Fld@annotnames
3179         /FT/Btn%
3180         \Fld@flags
3181         /H/P%
3182         /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3183         \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3184             \ifx\Fld@bordercolor\relax\else 1\fi
3185             \space
3186         \else
3187             /MK<<%
3188             \ifnum\Fld@rotation=\z@
3189             \else
3190                 /R~\Fld@rotation
3191             \fi
3192             \ifx\Fld@bordercolor\relax
3193             \else
3194                 /BC[\Fld@bordercolor]%
3195             \fi
3196             >>%
3197             \fi
3198             /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3199             /A<</S/ResetForm>>%
3200             \Fld@additionalactions
3201         }%
3202     }
3203
3204     %these patterns are used in hyperref checks.
3205     %it is unclear if they are really useful and if a backend support is
3206     %needed.
3207     \str_case:VnF \c_sys_backend_str
3208     {
3209         { pdfmode }
3210         {
3211             \def\HyPat@ObjRef
3212             {
3213                 [0-9]*[1-9][0-9]*~0~R
3214             }
3215         }

```

```

3216 { dvipdfmx }
3217 {
3218   \def\HyPat@ObjRef
3219   {
3220     @[\~]+
3221   }
3222 }
3223 { xdvipdfmx }
3224 {
3225   \def\HyPat@ObjRef
3226   {
3227     @[\~]+
3228   }
3229 }
3230 }
3231 { %also set in hyperref sty, so probably not needed.
3232   \def\HyPat@ObjRef/{.+}
3233 }
3234
3235
3236 \ExplSyntaxOff
3237 % UF: removed Hy@writebookmark
3238 %   \Hy@currentbookmarklevel{0}
3239 %   \Hy@numberline
3240 %   \__hypwritetorep
3241 %   counter{bookmark@seq@number}
3242 % removed \HyPsd@SanitizeForOutFile, not needed
3243 % removed \currentpdfbookmark, defined by bookmark,
3244 % should use \newcommand there
3245 % removed \subpdfbookmark, defined by bookmark,
3246 % should use \newcommand there
3247 % removed \belowpdfbookmark, defined by bookmark,
3248 % should use \newcommand there
3249 % removed \pdfbookmark, defined by bookmark,
3250 % \BOOKMARK
3251 % \@BOOKMARK
3252 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3253 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3254 %% removed \ReadBookmarks / unneeded with bookmark.
3255 %% removed \Hy@OutlineName
3256 %% removed \check@bm@number
3257 %% removed \calc@bm@number
3258
3259 \ifHy@implicit
3260 \else
3261   \expandafter\endinput
3262 \fi
3263 \newlength\Hy@SectionHShift
3264 \def\Hy@SectionAnchorHref#1{%
3265   \ifx\protect\@typeset@protect
3266     \Hy__hypSectionAnchor{#1}%
3267   \fi
3268 }
3269 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%

```

```

3270 \leavevmode
3271 \hbox to 0pt{%
3272   \kern-\Hy@SectionHShift
3273   \Hy@raisedlink{%
3274     \hyper@anchorstart{#1}\hyper@anchorend
3275   }%
3276   \hss
3277 }%
3278 }
3279 \let\H@old@ssect\@ssect
3280 \def\@ssect#1#2#3#4#5{%
3281   \Hy@MakeCurrentHrefAuto{section*}%
3282   \setlength{\Hy@SectionHShift}{#1}%
3283   \begingroup
3284     \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3285     \toks\tw@ \expandafter{%
3286       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3287       #5%
3288     }%
3289   \edef\x{\endgroup
3290     \the\toks@{\the\toks\tw@}%
3291   }\x
3292 }
3293 \let\H@old@schapter\@schapter
3294 \def\@schapter#1{%
3295   \begingroup
3296     \let\@mkboth\@gobbletwo
3297     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3298     \Hy@raisedlink{%
3299       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3300     }%
3301   \endgroup
3302   \H@old@schapter{#1}%
3303 }
3304 \ltx@ifundefined{chapter}{-}{%
3305   \let\Hy@org@chapter\@chapter
3306   \def\@chapter{%
3307     \def\Hy@next{%
3308       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3309       \Hy@raisedlink{%
3310         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3311       }%
3312     }%
3313     \ifnum\c@secnumdepth>\m@ne
3314       \ltx@ifundefined{if@mainmatter}%
3315       \iftrue{\csname if@mainmatter\endcsname}%
3316       \let\Hy@next\relax
3317     \fi
3318     \fi
3319     \Hy@next
3320     \Hy@org@chapter
3321   }%
3322 }
3323 \let\H@old@part\@part

```

```

3324 \begingroup\expandafter\expandafter\expandafter\endgroup
3325 \expandafter\ifx\csname chapter\endcsname\relax
3326   \let\Hy@secnum@part\z@
3327 \else
3328   \let\Hy@secnum@part\m@ne
3329 \fi
3330 \def\@part{%
3331   \ifnum\Hy@secnum@part>\c@secnumdepth
3332     \phantomsection
3333   \fi
3334   \H@old@part
3335 }
3336 \let\H@old@part\@spart
3337 \def\@spart#1{%
3338   \Hy@MakeCurrentHrefAuto{part*}%
3339   \Hy@raisedlink{%
3340     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3341   }%
3342   \H@old@part{#1}%
3343 }
3344 \let\H@old@sect\@sect
3345 \def\@sect#1#2#3#4#5#6[#7]#8{%
3346   \ifnum #2>\c@secnumdepth
3347     \expandafter\@firstoftwo
3348   \else
3349     \expandafter\@secondoftwo
3350   \fi
3351   {%
3352     \Hy@MakeCurrentHrefAuto{section*}%
3353     \setlength{\Hy@SectionHShift}{#3}%
3354     \begingroup
3355       \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[\@currentHref]}%
3356       \toks\tw@\expandafter{%
3357         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3358         #8%
3359       }%
3360     \edef\x{\endgroup
3361       \the\toks@{\the\toks\tw@}%
3362     }\x
3363   }{%
3364     \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[\@currentHref]{#8}%
3365   }%
3366 }
3367 \expandafter\def\csname Parent-4\endcsname{}
3368 \expandafter\def\csname Parent-3\endcsname{}
3369 \expandafter\def\csname Parent-2\endcsname{}
3370 \expandafter\def\csname Parent-1\endcsname{}
3371 \expandafter\def\csname Parent0\endcsname{}
3372 \expandafter\def\csname Parent1\endcsname{}
3373 \expandafter\def\csname Parent2\endcsname{}
3374 \expandafter\def\csname Parent3\endcsname{}
3375 \expandafter\def\csname Parent4\endcsname{}
3376 %%
3377 %% End of file 'hgeneric-experimental.def'.

```

```

3378 \end{package}
3379 \begin{colorscheme}
3380 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3381 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3382 % linkcolor=
3383 % filecolor=
3384 % urlcolor=
3385 % menucolor=
3386 % runcolor=
3387 % linkbordercolor=
3388 % filebordercolor=
3389 % urlbordercolor=
3390 % menubordercolor=
3391 % runbordercolor=
3392
3393 \prop_const_from_keyval:Nn \c__hyp_colorscheme_hypori_prop
3394 {
3395   linkcolor      = [rgb]{1,0,0}, %red
3396   filecolor      = [rgb]{0,1,1}, %cyan
3397   urlcolor       = [rgb]{1,0,1}, %magenta
3398   menucolor      = [rgb]{1, 0, 0}, %red
3399   runcolor       = [rgb]{0,1,1}, %cyan
3400   %-----
3401   linkbordercolor = [rgb]{1, 0 ,0 },
3402   filebordercolor = [rgb]{0, .5, .5},
3403   urlbordercolor  = [rgb]{0, 1, 1},
3404   menubordercolor = [rgb]{1, 0, 0},
3405   runbordercolor  = [rgb]{0, .7, .7}
3406 }
3407
3408 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3409 {
3410   linkcolor      = [rgb]{0,0.2,0.6},
3411   filecolor      = [rgb]{0.8,0,0.8},
3412   urlcolor       = [rgb]{0.8,0,0.8},
3413   menucolor      = [rgb]{0,0.2,0.6},
3414   runcolor       = [rgb]{0.8,0,0.8},
3415   %----- %-----
3416   linkbordercolor = [rgb]{0,0.2,0.6},
3417   filebordercolor = [rgb]{0.8,0,0.8},
3418   urlbordercolor  = [rgb]{0.8,0,0.8},
3419   menubordercolor = [rgb]{0,0.2,0.6},
3420   runbordercolor  = [rgb]{0.8,0,0.8}
3421 }
3422
3423 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3424 { %two colors: intern/extern
3425   linkcolor      = [rgb]{0.79216, 0, 0.12549},
3426   filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3427   urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3428   menucolor      = [rgb]{0.79216, 0, 0.12549 },
3429   runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3430   %----- %-----
3431   linkbordercolor = [rgb]{0.79216, 0, 0.12549},

```

```

3432     filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3433     urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3434     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3435     runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3436 }
3437
3438 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3439 { %all darkgray
3440     linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3441     filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3442     urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3443     menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3444     runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3445     %----- %-----
3446     linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3447     filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3448     urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3449     menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3450     runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3451 }
3452
3453 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3454 { %dvipsnam.def
3455     linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3456     filecolor      = [rgb]{1, 0, 0}, %Red
3457     urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3458     menucolor      = [rgb]{1, 0, 0}, %Red
3459     runcolor       = [rgb]{1, 0, 0}, %Red
3460     %----- %-----
3461     linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3462     filebordercolor = [rgb]{1, 0, 0}, %Red
3463     urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3464     menubordercolor = [rgb]{1, 0, 0}, %Red
3465     runbordercolor  = [rgb]{1, 0, 0} %Red
3466 }
3467
3468 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3469 { %dvipsnam.def
3470     linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3471     filecolor      = [rgb]{0, 1, 0}, %Green
3472     urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3473     menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3474     runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3475     %----- %-----
3476     linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3477     filebordercolor = [rgb]{0, 1, 0}, %Green
3478     urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3479     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3480     runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3481 }
3482
3483
3484 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3485 {

```

```

3486 linkcolor      = [rgb]{0.50196, 0, 0.02353},
3487 filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3488 urlcolor       = [rgb]{0.54118, 0, 0.52941},
3489 menucolor      = [rgb]{0.44706, 0.45882, 0},
3490 runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3491 %----- %-----
3492 linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3493 filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3494 urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3495 menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3496 runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3497 }
3498
3499 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3500 {
3501   linkcolor      = [rgb]{0,0,0},
3502   filecolor      = [rgb]{0,0,0},
3503   urlcolor       = [rgb]{0,0,0},
3504   menucolor      = [rgb]{0,0,0},
3505   runcolor       = [rgb]{0,0,0},
3506   %----- %-----
3507   linkbordercolor = [rgb]{0,0,0},
3508   filebordercolor = [rgb]{0,0,0},
3509   urlbordercolor  = [rgb]{0,0,0},
3510   menubordercolor = [rgb]{0,0,0},
3511   runbordercolor  = [rgb]{0,0,0}
3512 }
3513 </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			
\#	491	\AddToHook 205
\%	492	\advance 2404, 2478, 2522, 2523
\.	306, 310, 312	allcolor (hypersetup key) <u>758</u>
@curropt commands:			\author 2
\@curropt:	2471, 2539	
			B
\\	19, 20, 21,	\b 308
		28, 32, 42, 62, 69, 76, 83, 90, 105,	\begingroup ... 2246, 2438, 2467, 2479,
		110, 111, 119, 120, 129, 130, 131,	2513, 2850, 3283, 3295, 3324, 3354
		132, 133, 139, 617, 2473, 2541, 2828	\belowpdfbookmark 3247
_	304, 306, 310, 312	\BOOKMARK 3250
			bookmarkstype (hypersetup key) 10
			bool commands:
A			\bool_if:NTF ... 437, 451, 462, 475,
\A	304	518, 568, 628, 639, 728, 738, 978, 989
\Acrobatmenu	15, 156	\bool_if:nTF 580, 1681, 1706, 2989, 3097
\addcontentsline	10	\bool_lazy_and:nnTF 200
\AddToDocumentMetadata	180	

<code>\documentclass</code>	22	2476, 2494, 2497, 2498, 2504, 2507,
<code>draft</code> (hypersetup key)	<u>1186</u>	2516, 2518, 2525, 2528, 2803, 2825,
		2835, 2837, 2843, 2847, 2854, 2855,
		2864, 2868, 2872, 2876, 2880, 2884,
		2888, 2892, 2896, 2900, 2917, 2921,
		3022, 3023, 3024, 3031, 3035, 3039,
		3041, 3043, 3062, 3066, 3070, 3074,
		3081, 3146, 3147, 3154, 3158, 3160,
		3183, 3184, 3191, 3195, 3197, 3262,
		3267, 3317, 3318, 3329, 3333, 3350
		<code>file</code> (hypersetup key)
		7, <u>1254</u>
		<code>file</code> commands:
		<code>\file_input:n</code>
		10
		<code>fileborderstyle</code> (hypersetup key) .
		11, <u>868</u>
		<code>filecolor</code> (hypersetup key)
		758
		<code>final</code> (hypersetup key)
		<u>1186</u>
		<code>\fontencoding</code>
		2339, 2351, 2363, 2375
		<code>\fontfamily</code>
		2338, 2350, 2362, 2374
		<code>\fontseries</code>
		2340, 2352, 2364, 2376
		<code>\fontshape</code>
		2341, 2353, 2365, 2377
		<code>fp</code> commands:
		<code>\fp_eval:n</code>
		1836, 1864, 1880
		G
		<code>\GetDocumentMetadata</code>
		2
		<code>\global</code>
		2404, 2793
		<code>group</code> commands:
		<code>\group_begin:</code>
		. 332, 440, 454, 477, 520, 570, 630,
		730, 981, 998, 2114, 2337, 2349,
		2361, 2373, 2588, 2631, 2667, 2700
		<code>\group_end:</code>
		. 339, 444, 465, 507, 558, 611, 666,
		740, 1001, 1004, 2121, 2344, 2356,
		2368, 2380, 2624, 2660, 2693, 2725
		H
		<code>\hbox</code>
		3271
		<code>hbox</code> commands:
		<code>\hbox_overlap_right:n</code>
		995
		<code>\hbox_set:Nn</code>
		2614, 2645, 2684
		<code>\hbox_set:Nw</code>
		982
		<code>\hbox_set_end:</code>
		991
		<code>\hbox_set_to_wd:Nnn</code>
		1862
		<code>hidefile</code> (hypersetup key)
		<u>1138</u>
		<code>hidelink</code> (hypersetup key)
		<u>1138</u>
		<code>hidelinks</code> (hypersetup key)
		<u>1138</u>
		<code>hidemenu</code> (hypersetup key)
		<u>1138</u>
		<code>hiderun</code> (hypersetup key)
		<u>1138</u>
		<code>hideurl</code> (hypersetup key)
		<u>1138</u>
		<code>hook</code> commands:
		<code>\hook_gput_code:nnn</code>
		192, 724, 734, 974, 985
		<code>\hook_new:n</code>
		183, 317
		</

\hook_use:n	334, 441, 455	__hyp_colormodel_set_aux:nwn	705, 709, 716
\hss	3276	\c__hyp_colorscheme_daleif_prop	3408
Hy internal commands:		\c__hyp_colorscheme_henryford_	3499
\Hy_hyescapeform	2814, 2827, 2833	prop	3393
\Hy_hypSectionAnchor	3266, 3269	\c__hyp_colorscheme_julian_prop	3423
HyField internal commands:		\c__hyp_colorscheme_phelype_prop	3484
\HyField_hypAddToFields	2239, 2268	\c__hyp_colorscheme_szabolcsA_	3453
hyp commands:		prop	3468
\l_hyp_annot_colorfile_bool	288	\c__hyp_colorscheme_tivv_prop	3438
\l_hyp_annot_colorlink_bool	288	\l_hyp_dest_box	24, 301, 378, 379, 380, 1862, 1878, 1885
\l_hyp_annot_colormenu_bool	288	\l_hyp_dest_name_tmpa_tl	244, 411, 412, 418, 422, 424, 427, 531, 539
\l_hyp_annot_colorrurl_bool	288	\l_hyp_dest_pdfremotestartview_	270, 536, 1669, 1673
\l_hyp_annot_colorurl_bool	288	tl	270, 1680, 1682, 1689, 1707, 1714
\l_hyp_annot_ocgcolorfile_bool	292	\g__hyp_dest_pdfstartpage_tl	270, 1682, 1689, 1700, 1704, 1707, 1714
\l_hyp_annot_ocgcolorlink_bool	292	\g__hyp_dest_pdfstartview_tl	270, 1682, 1689, 1700, 1704, 1707, 1714
\l_hyp_annot_ocgcolormenu_bool	292	\l_hyp_dest_pdfview_tl	286, 394, 401, 1831, 1834, 1841, 1844, 1845, 1846, 1847, 1848, 1849, 1854, 1858, 1894
\l_hyp_annot_ocgcolorurl_bool	292	\c__hyp_dest_startview_regex	302, 1667, 1698
hyp internal commands:		\c__hyp_dest_undefined_tl	250, 417, 418
\g__hyp_AcroForm_CoFields_prop	2200, 2221, 2297, 2299, 2314	__hyp_destination:nn	24, 365, 365, 394, 401
\g__hyp_AcroForm_Fields_prop	2201, 2214, 2292	\l_hyp_filename_tmpa_tl	244, 521, 522, 524, 526, 527, 574, 575, 591
\l_hyp_annot_GoTo_bool	437, 451, 462	__hyp_link_goto_begin:nw	408, 442, 456
\l_hyp_annot_GoTo_bool\l_hyp_annot_URI_bool\l_hyp_annot_GoToR_bool\l_hyp_annot_Named_bool	296	__hyp_link_goto_end:	430, 443, 464
hyp_annot_Launch_bool	518	\c__hyp_map_annot_hyp_prop	251
\l_hyp_annot_GoToR_bool	518	\c__hyp_map_hyp_annot_prop	251, 722, 786, 822, 849, 857, 868, 895, 903, 972, 1076, 1112, 1122, 1254
\l_hyp_annot_Launch_bool	568	__hyp_ocg_init:	914, 914, 970, 980
\l_hyp_annot_Named_bool	628	__hyp_PageLabels_gpush:	349, 349, 362
\c__hyp_annot_types_seq	251, 288, 292, 758, 1053, 1149	\l_hyp_para_tmpa_tl	247, 579, 582, 591
\l_hyp_annot_URI_bool	475	\l_hyp_para_tmpa_tl\l_hyp_	244
\g__hyp_bordercolormodel_str	287, 783, 799	text_tmpa_str\g__hyp_text_	244
\l_hyp_CheckmarkOff_tl	2282, 2283, 2939	tmpa_str	2532, 2533
\l_hyp_CheckmarkYes_tl	2280, 2281, 2938	__hyp_ref_check:nn	211, 221, 233, 2409
__hyp_clist_display:n	105, 112, 121	__hyp_ref_if_exist:nn	211, 219, 220
__hyp_color_export:nnN	31, 671, 671, 687, 799	__hyp_ref_if_exist:nnTF	223, 2413
__hyp_color_select:n	31, 688, 688, 704	__hyp_ref_label:nn	211, 212, 218, 2408
__hyp_color_select:nN	31		
__hyp_color_set:nN	32		
__hyp_color_set:nn	32, 188, 190, 705, 705, 721, 763		
__hyp_colormodel_export:nnnN	31, 671		
__hyp_colormodel_export:wnnN	675, 682		
__hyp_colormodel_select_aux:wn	688, 692, 699		

_hyp_ref_value:nn	211 , 234 , 238 , 2417	\hypersetup keys:	
_hyp_setup_info_date_key:nn	...	allcolor	758
...	1969 , 2001 , 2002	bookmarkstyle	10
_hyp_setup_info_key:nn	...	bordercolormodel	10 , 779
...	1962 , 1963 , 1964 , 1965 , 1967 , 1968	colorfile	758
_hyp_store_metadata:nn	...	colorlink	758
...	177 , 182 , 1910 , 1930 , 1948 , 1982 , 1996 , 2017 , 2089	colorlinks	744
_hyp_text_cleanup:N	322 , 322 , 336	colormenu	758
_hyp_text_convert:nN	326 , 326 , 337	colorruncolorscheme	1162
\l_hyp_text_enc_dest_print_tl	...	colorurl	758
...	273 , 530	debug	1186
\l_hyp_text_enc_dest_tl	...	destlabel	10
...	24 , 273 , 372 , 423	draft	1186
\l_hyp_text_enc_file_print_tl	...	extension	10 , 1208
...	273 , 525 , 573	file	7 , 1254
\l_hyp_text_enc_info_print_tl	...	fileborderstyle	11 , 868
...	273 , 347	filecolor	758
\l_hyp_text_enc_para_print_tl	...	final	1186
...	273 , 578	hidefile	1138
\l_hyp_text_enc_uri_print_tl	...	hidelink	1138
...	273 , 480 , 1265	hidelinks	1138
_hyp_text_pdfstring:nnN	...	hidemenu	1138
...	330 , 330 , 342 , 344 , 347 , 370 , 421 , 478 , 523 , 528 , 571 , 576 , 1265	hiderun	1138
_hyp_text_pdfstring_info:nN	...	hideurl	1138
...	345 , 345 , 1924 , 1942 , 1952	hypertextnames	1208
_hyp_text_purify:nN	318 , 318 , 335	link	7 , 1254
\g_hyp_text_tmpa_str	249 , 338 , 340	linkborder	11
\l_hyp_text_tmpa_str	...	linkborderstyle	11 , 868
...	248 , 335 , 336 , 337 , 338	linkcolor	758
\l_hyp_tmpa_box	239 , 982 , 995 , 1000	linkfileprefix	1208
\l_hyp_tmpa_int	239	linktoc	1208
\l_hyp_tmpa_seq	239 , 1667 , 1669 , 1698 , 1700 , 1821 , 1822 , 1826 , 1828 , 1852 , 1859 , 1860 , 1861 , 1870 , 1872 , 1882 , 1887 , 2301 , 2303 , 2309	linktocpage	1208
\l_hyp_tmpa_str	...	localanchorname	1208
...	239 , 1924 , 1925 , 1927 , 1942 , 1943 , 1945 , 1952 , 1953 , 1957	menu	7 , 1254
\l_hyp_tmpa_tl	...	menuborder	11
...	239 , 373 , 377 , 385 , 632 , 634 , 799 , 803 , 1265 , 1266 , 1271 , 1665 , 1667 , 1696 , 1698 , 1828 , 1829 , 1836	menuborderstyle	11 , 868
\l_hyp_uri_tmpa_tl	244 , 481 , 482	menucolor	758
hyp/annot/file (color name)	287	naturalnames	1208
hyp/annot/link (color name)	287	nesting	11
hyp/annot/menu (color name)	287	ocgcolorfile	1008
hyp/annot/run (color name)	287	ocgcolorlink	1008
hyp/annot/url (color name)	287	ocgcolorlinks	1008
hyp/text/pdfstring	317	ocgcolormenu	1008
\hypercalcbp	9 , 15 , 140	ocgcolorruncolorurl	1008
\HyperDestNameFilter	371 , 422	pageanchor	1208
\hypersetup	1 , 2 , 6 , 7 , 10 , 15 , 59 , 91 , 163	pdfauthor	1913
		pdfborder	11
		pdfborderstyle	11 , 868
		pdfcreationdate	11 , 1969
		pdfcreator	1913
		pdfencoding	1176
		pdfinfo	2031

mode commands:		
\mode_if_horizontal:TF	367, 389	
\mode_leave_vertical:		
.....	410, 494, 546, 598,	
.....	641, 2575, 2599, 2611, 2642, 2672, 2715	
msg commands:		
\msg_error:nn	202, 2596, 2675	
\msg_line_context:	70	
\g_msg_module_name_prop	13	
\msg_new:nnn	38, 45, 50, 54,	
.....	58, 65, 72, 79, 86, 93, 99, 106, 115, 125	
\msg_new:nnnn	14, 24	
\msg_warning:nn	153, 1182	
\msg_warning:nnn	162, 414, 663	
\msg_warning:nnnn		
.....	1014, 1061, 1304, 1348,	
.....	1382, 1405, 1467, 1496, 1530, 1545,	
.....	1558, 1571, 1596, 1629, 1645, 1672,	
.....	1703, 1736, 1750, 1777, 1811, 1893	
\msg_warning:nnnnn	173,	
.....	1098, 1132, 1243, 1324, 1361, 1448,	
.....	1481, 1506, 1582, 1609, 1658, 1763,	
.....	1790, 2025, 2132, 2151, 2170, 2181	
N		
naturalnames (hypersetup key)	1208	
nesting (hypersetup key)	11	
\newcommand	143, 3244, 3246, 3248	
\newcount	2400, 2571	
\NewDocumentCommand	2194	
\newlength	3263	
\noexpand	2482, 2483, 2484, 2485, 2811	
O		
ocgcolorfile (hypersetup key)	1008	
ocgcolorlink (hypersetup key)	1008	
ocgcolorlinks (hypersetup key)	1008	
ocgcolormenu (hypersetup key)	1008	
ocgcolorrun (hypersetup key)	1008	
ocgcolorurl (hypersetup key)	1008	
P		
page/Trans internal commands:		
l__hyp_page/Trans	315	
pageanchor (hypersetup key)	1208	
\paperwidth	3	
\PassOptionsToPackage	209, 1200, 1205	
pdf commands:		
\pdf_bdc_obj:nn	994, 997	
\pdf_destination:nn	19, 136, 384	
\pdf_destination:nnnn	377	
\pdf_emc:	996, 1002	
\pdf_link_last:	2241, 2258	
\pdf_link_user:nnn	2578	
\pdf_object_last:	2120	
\pdf_object_new:nn		
.....	916, 917, 918, 919, 2733, 2734, 2735	
\pdf_object_now:nn	2116	
\pdf_object_pageref:n	1689, 1714, 2415	
\pdf_object_ref:n		
.....	922, 924, 946, 947, 950,	
.....	954, 959, 964, 969, 2319, 2321, 2791	
\pdf_object_write:nn		
.....	920, 926, 936, 948, 2736, 2778, 2785	
\pdf_text_convert:nn	643, 1957	
\pdf_text_convert:nnN	328	
\pdf_version:		
.....	1352, 1471, 1500, 1534, 1549,	
.....	1575, 1600, 1633, 1649, 1754, 1781	
\pdf_version_compare:NnTF		
.....	1008, 1055,	
.....	1342, 1433, 1455, 1491, 1524, 1539,	
.....	1565, 1590, 1616, 1639, 1744, 1771	
\pdf_version_compare_p:Nn	584	
\pdf_version_major:	150, 1017, 1065	
\pdf_version_minor:	149, 1017, 1065	
\pdf_xform_new:nnn	2196	
\pdf_xform_ref:n		
.....	2558, 2559, 2653, 2654, 2938, 2939	
pdfannot commands:		
\pdfannot_box:nnnn		
.....	2450, 2548, 2615, 2646, 2685, 2718	
\pdfannot_box_last:	2242, 2259	
\pdfannot_dict_put:nnn		
.....	800, 835, 859, 881, 905, 1083, 1114	
\pdfannot_dict_remove:nn		
.....	794, 830, 851, 876, 897, 1091, 1124	
\pdfannot_link:nnn	495, 547, 599, 649	
\pdfannot_link_goto_begin:nw	427	
\pdfannot_link_goto_end:	432	
\pdfannot_link_margin:n	5, 1386	
\c_pdfannot_link_types_seq	296	
pdfauthor (hypersetup key)	1913	
\pdfbookmark	3249	
pdfborder (hypersetup key)	11	
pdfborderstyle (hypersetup key)	11, 868	
pdfcreationdate (hypersetup key)	11, 1969	
pdfcreator (hypersetup key)	1913	
\pdfdest	8	
pdfdict commands:		
\pdfdict_new:n	315, 469, 512, 562, 622	
\pdfdict_put:nnn		
.....	316, 470, 471, 482, 484, 489, 513,	
.....	514, 527, 535, 539, 544, 563, 564,	
.....	575, 590, 596, 623, 624, 642, 647,	
.....	1412, 1413, 1417, 1418, 2129, 2139,	
.....	2143, 2146, 2148, 2167, 2176, 2178	
\pdfdict_remove:nn	587, 1422, 1423	

\pdfdict_use:n	499, 551, 604, 653, 2118
pdfencoding (hypersetup key)	1176
\pdfescapestring	2800
PDFForm internal commands:	
\PDFForm__hypName	2839, 2842, 2846, 2849
pdfinfo (hypersetup key)	2031
pdfkeywords (hypersetup key)	1913
pdflang (hypersetup key)	11, 1899
pdflinkmargin (hypersetup key)	11
pdfmanagement commands:	
\pdfmanagement_add:nn	1395
\pdfmanagement_add:nnn	351, 946, 947, 969, 1271, 1296, 1312, 1316, 1336, 1344, 1373, 1428, 1438, 1462, 1476, 1488, 1493, 1518, 1526, 1541, 1567, 1592, 1624, 1641, 1687, 1712, 1726, 1746, 1773, 1802, 1908, 1927, 1945, 1956, 1980, 1994, 2012, 2101, 2120, 2294, 2311, 2318, 2320, 2322, 2331
\pdfmanagement_if_active:p	201
\pdfmanagement_remove:nn	1268, 1292, 1300, 1320, 1332, 1357, 1369, 1377, 1391, 1400, 1444, 1459, 1478, 1515, 1554, 1579, 1605, 1620, 1654, 1684, 1709, 1722, 1731, 1759, 1786, 1798, 1807, 1905, 1921, 1939, 1977, 1991, 2098, 2111, 2326
\pdfmanagement_show:n	2295
pdfmeta commands:	
\g_pdfmeta_standard_prop	632, 2324, 2594, 2673, 2907, 2977, 3172
pdfmoddate (hypersetup key)	11, 1969
pdfproducer (hypersetup key)	1913
pdfremotestartview (hypersetup key)	9
pdfstartview (hypersetup key)	9
\pdfstringdef	6, 7, 15, 21, 2856
pdfsubject (hypersetup key)	1913
pdftitle (hypersetup key)	1913
pdftrapped (hypersetup key)	2003
pdfversion (hypersetup key)	1176
pdfview (hypersetup key)	9, 1817
\phantom	2355, 2379
\phantomsection	3332
plainpages (hypersetup key)	1208
prg commands:	
\prg_generate_conditional_variant:Nnn	220
\prg_new_eq_conditional:NNn	219
prop commands:	
\prop_const_from_keyval:Nn	253, 261, 3393, 3408, 3423, 3438, 3453, 3468, 3484, 3499
\prop_get:NnNTF	632
\prop_gput:Nnn	13, 2214, 2221
\prop_if_empty:NNTF	2297
\prop_if_in:NnNTF	2324, 2594, 2673, 2907, 2977, 3172
\prop_item:Nn	2314
\prop_map_inline:Nn	722, 786, 822, 849, 857, 868, 895, 903, 972, 1076, 1112, 1122, 1166, 1254, 2292, 2299
\prop_new:N	2200, 2201
\protect	225, 3265
\providecommand	141, 142, 2228, 2231, 2460, 2796
\ProvidesFile	3
R	
\ReadBookmarks	3254
ref commands:	
\ref_if_exist:nn	219
\ref_label:nn	215
\ref_value:nn	17, 236
regex commands:	
\regex_const:Nn	302
\regex_extract_once:NnNTF	1667, 1698
\relax	2463, 2464, 2505, 2793, 2840, 2844, 2914, 2918, 3023, 3024, 3032, 3036, 3063, 3067, 3147, 3155, 3184, 3192, 3316, 3325
\RemoveFromHook	1276
\RenewDocumentCommand	156
\RequirePackage	7, 20, 205, 3252
run (hypersetup key)	7, 1254
runborder (hypersetup key)	11
runborderstyle (hypersetup key)	11, 868
runcolor (hypersetup key)	758
S	
\selectfont	2342, 2354, 2366, 2378
seq commands:	
\seq_const_from_clist:Nn	251
\seq_count:N	1826, 1852
\seq_get_right:NN	1828
\seq_item:Nn	1669, 1700, 1822, 1870, 1872, 1882, 1887
\seq_map_inline:Nn	288, 292, 296, 758, 1053, 1149, 2309
\seq_new:N	241
\seq_put_right:Nn	1859, 1860, 1861, 2301
\seq_set_split:Nnn	1821
\seq_sort:Nn	2303
\setlength	3282, 3353
\setpdflinkmargin	5
\settowidth	2475

<code>\show</code>	2536	<code>\empty</code>	2429, 2445, 2861, 2865, 2869, 2873, 2877, 2881, 2885, 2889, 2893, 2897, 3043, 3074, 3075
sort commands:		<code>\endForm</code>	2395
<code>\sort_return_same:</code>	2307	<code>\esphack</code>	216
<code>\sort_return_swapped:</code>	2306	<code>\expl@color@set@nnn</code>	9
<code>\space</code>	229, 2544, 3025, 3042, 3043, 3073, 3074, 3148, 3182, 3185	<code>\@firstofone</code> ..	2234, 2802, 2807, 2824
<code>\spacefactor</code>	367, 389	<code>\@firstoftwo</code>	3347
<code>\stockwidth</code>	3	<code>\@for</code>	2471, 2539
str commands:		<code>\@gobble</code>	2287
<code>\c_colon_str</code>	617	<code>\@gobbletwo</code>	3296
<code>\c_hash_str</code>	491	<code>\@hyper@launch</code>	29, 617
<code>\c_percent_str</code>	492	<code>\@ifundefined</code>	2287, 2816
<code>\str_case:nnTF</code>	3207	<code>\@latex@warning</code>	226
<code>\str_case_e:nnTF</code>	1822	<code>\@mainaux</code>	2227, 2230, 2251, 2271
<code>\str_gset:Nn</code>	783	<code>\@mkboth</code>	3296
<code>\str_gset_eq:NN</code>	338	<code>\@nil</code>	2814, 2827, 2833
<code>\str_head:n</code>	2014, 2019	<code>\@part</code>	3323, 3330
<code>\str_if_eq:nnTF</code>		<code>\@pdfauthor</code>	16
.....	374, 1925, 1943, 1953, 2905	<code>\@pdfborder</code>	141
<code>\str_if_eq_p:nn</code>	582	<code>\@pdfborderstyle</code>	142
<code>\str_lowercase:n</code>		<code>\@savs</code>	367, 389
.....	1822, 1948, 1996, 2015, 2020	<code>\@schapter</code>	3293, 3294
<code>\str_new:N</code>	243, 248, 249, 287	<code>\@secondoftwo</code>	3349
<code>\str_remove_once:Nn</code>	488, 543, 595, 646	<code>\@sect</code>	3344, 3345
<code>\str_set:Nn</code>	320	<code>\@spart</code>	3336, 3337
<code>\str_set_convert:Nnnn</code>	6	<code>\@ssect</code>	3279, 3280
<code>\str_set_eq:NN</code>	340	<code>\@tempdima</code> ..	2469, 2476, 2478, 2515, 2516, 2517, 2521, 2522, 2523, 2524
<code>\str_tail:n</code>	2015, 2020	<code>\@tempdimb</code>	2475, 2476
<code>\str_uppercase:n</code>	2008, 2014, 2019	<code>\@typeset@protect</code>	3265
<code>\string</code>	2228, 2231, 2253, 2272, 2816, 2820, 2831	<code>\c@secnumdepth</code>	3313, 3331, 3346
<code>\subpdfbookmark</code>	3245	<code>\calc@bm@number</code>	3257
sys commands:		<code>\check@bm@number</code>	3256
<code>\c_sys_backend_str</code>	3207	<code>\Fld@additionalactions</code>	2903, 2973, 3009, 3045, 3082, 3129, 3169, 3200
T			
\TeX and \LaTeX 2 _ε commands:		<code>\Fld@align</code>	2932, 3020, 3095
<code>\@BOOKMARK</code>	3251	<code>\Fld@altname</code> ..	2840, 2842, 2914, 2916
<code>\@CheckBox</code>	2696	<code>\Fld@annotflags</code>	2927, 2982, 3015, 3051, 3090, 3139, 3176
<code>\@ChoiceMenu</code>	2461	<code>\Fld@annotnames</code>	2912, 2929, 2984, 3017, 3053, 3092, 3141, 3178
<code>\@Form</code>	2285	<code>\Fld@bcolor</code>	2951, 2953, 3036, 3038, 3067, 3069, 3103, 3116, 3118
<code>\@PushButton</code>	2586	<code>\fld@bcolor</code>	3024
<code>\@Reset</code>	2663	<code>\Fld@bordercolor</code>	2947, 2949, 2993, 3002, 3004, 3023, 3032, 3034, 3063, 3065, 3101, 3112, 3114, 3147, 3155, 3157, 3184, 3192, 3194
<code>\@Submit</code>	2627	<code>\Fld@borderstyle</code>	2933, 2988, 3021, 3057, 3096, 3145, 3198
<code>\@TextField</code>	2426	<code>\Fld@borderwidth</code> ..	2522, 2523, 2933, 2988, 3021, 3057, 3096, 3145, 3198
<code>\@bookmarksopenlevel</code>	1287	<code>\Fld@calculate@code</code> ..	2244, 2873, 2875
<code>\@bsphack</code>	214		
<code>\@chapter</code>	3305, 3306		
<code>\@currDisplay</code>	2475, 2544		
<code>\@currValue</code>	2542, 2558		
<code>\@currentHref</code>			
.....	3286, 3299, 3310, 3340, 3357		
<code>\@curropt</code> ..	2460, 2470, 2473, 2538, 2541		

\Fld@calculate@sortkey	2255	\H@old@ssect	3279, 3284
\Fld@cbsymbol	2955	\Hy@abspage	361
\Fld@charsize		\Hy@activeanchorfalse	406
.. 2521, 2959, 3042, 3073, 3124, 3182		\Hy@activeanchortrue	399
\Fld@checkequals	2473, 2541	\Hy@AtBeginDocument	2208, 2224
\Fld@choices	3044	\Hy@bookmarksnumberedfalse ...	1280
\Fld@color 2960, 2962, 3043, 3074, 3125		\Hy@bookmarksnumberedtrue	1281
\Fld@default		\Hy@bookmarksopenfalse	1284
... 2428, 2445, 2463, 2506, 2536,		\Hy@bookmarksopentrue	1285
2537, 2699, 3075, 3079, 3080, 3127		\Hy@bookmarkstype	1288
\Fld@flags	2931,	\Hy@chapapp	3297, 3308
2986, 3019, 3055, 3094, 3143, 3180		\Hy@CleanupFile	522
\Fld@format@code	2865, 2867	\Hy@colorlink	26
\Fld@height		\Hy@currentbookmarklevel	3238
2431, 2452, 2455, 2466, 2524, 2550,		\Hy@DisableOption	148, 155
2563, 2581, 2630, 2666, 2702, 2720		\Hy@drafttrue	1199
\Fld@keystroke@code	2861, 2863	\Hy@escapeform	
\Fld@listcount 2535, 2543, 2564, 2571	 2449, 2547, 2577, 2613,	
\Fld@mappingname 2844, 2846, 2918, 2920		2644, 2683, 2717, 2798, 2805, 2808	
\Fld@maxlen	3130, 3132	\Hy@escapestring	2800,
\Fld@menulength 2468, 2474, 2521		2802, 2807, 2810, 2811, 2814, 2815,	
\Fld@name		2824, 2827, 2863, 2867, 2871, 2875,	
.. 2427, 2462, 2587, 2698, 2839, 2913		2879, 2883, 2887, 2891, 2895, 2899,	
\Fld@onblur@code	2881, 2883	2955, 3008, 3071, 3127, 3128, 3165	
\Fld@onclick@code	3008	\Hy@finaltrue	1204
\Fld@onenter@code	2893, 2895	\Hy@FormObjects 2291, 2731, 2793	
\Fld@onexit@code	2897, 2899	\Hy@gtmp	2856, 2858
\Fld@onfocus@code	2877, 2879	\Hy@href@nextactionraw	
\Fld@onmousedown@code ... 2885, 2887	 486, 488, 489, 541,	
\Fld@onmouseup@code	2889, 2891	543, 544, 593, 595, 596, 644, 646, 647	
\Fld@pageobjref		\Hy@href@page	536
..... 2397, 2411, 2796, 2928,		\Hy@hypertextnamesfalse	1214
2983, 3016, 3052, 3091, 3140, 3177		\Hy@hypertextnamestrue	1213
\Fld@radiosymbol	3071	\Hy@linkfileprefix	1216
\Fld@rotation	2943,	\Hy@linktoc	1239
2945, 2991, 2998, 3000, 3022, 3028,		\Hy@localanchornamefalse	1219
3030, 3059, 3061, 3099, 3108, 3110,		\Hy@localanchornamestrue	1218
3146, 3151, 3153, 3183, 3188, 3190		\Hy@MakeCurrentHrefAuto	
\Fld@submitflags	3167 3281, 3297, 3308, 3338, 3352	
\Fld@validate@code	2869, 2871	\Hy@MakeRemoteAction	532
\Fld@value		\Hy@naturalnamesfalse	1223
.. 2429, 2445, 2464, 2505, 2506, 3128		\Hy@naturalnamestrue	1222
\Fld@width	2430,	\Hy@next	3307, 3316, 3319
2444, 2451, 2455, 2465, 2504, 2515,		\Hy@numberline	144, 3239
2517, 2549, 2563, 2581, 2607, 2629,		\Hy@org@chapter	3305, 3320
2640, 2665, 2681, 2701, 2711, 2719		\Hy@OutlineName	3255
\Fld@X@additionalactions		\Hy@OutlineRerunCheck	3253
..... 2860, 2905, 2908		\Hy@pageanchorfalse	1227
\Form@action	3165	\Hy@pageanchortrue	1226
\G@refundefinedtrue	225	\Hy@pdfmajorversion	150
\H@old@part	3323, 3334	\Hy@pdfminorversion	149
\H@old@chapter	3293, 3302	\Hy@pdfstringtrue	22, 333
\H@old@ssect	3344, 3355, 3364	\Hy@pdfversion	2851
\H@old@spart	3336, 3342	\Hy@plainpagesfalse	1231

\Hy@plainpagetrue	1230	\hyper@link	26, 26, 435
\Hy@pstringdef	343, 2831	\hyper@linkend	25, 26, 460
\Hy@PutCatalog	349	\hyper@linkfile	516
\Hy@raisedlink	3273, 3298, 3309, 3339	\hyper@linklaunch	28, 566, 620
\Hy@RestoreLastskip	388	\hyper@linknamed	30, 158, 626
\Hy@safe@activetrue	493, 2247	\hyper@linkstart	25, 26, 449
\Hy@SaveLastskip	368	\hyper@linkurl	473
\Hy@secnum@part	3326, 3328, 3331	\HyPL@Labels	351, 361
\Hy@SectionAnchorHref	3264, 3286, 3357	\HyPL@StorePageLabel	349
\Hy@SectionHShift	3263, 3272, 3282, 3353	\HyPsd@LoadUnicode	145
\Hy@StepCount	2474, 2543	\HyPsd@pdfencoding	147
\Hy@temp	2248, 2264, 2813, 2814	\HyPsd@pdfencoding@unicode	147
\Hy@unicodetrue	2853	\HyPsd@SanitizeForOutFile	3242
\Hy@unicodetrue	146	\if@filesw	2226, 2250, 2270
\Hy@verboseanchor	369	\ifFld@combo	2489, 2519
\Hy@verbosefalse	1190	\ifFld@hidden	2444, 2504
\Hy@VerboseLinkStart	439, 453	\ifFld@multiline	2432
\Hy@VerboseLinkStop	445, 466, 505, 556, 609, 659	\ifFld@popdown	2490
\Hy@verbostrue	1189	\ifFld@radio	2486, 2509
\Hy@VersionChecked	137	\ifHy@href@ismap	483
\Hy@WrapperDef	365	\ifHy@implicit	3259
\Hy@xspace@end	442, 504, 555, 608, 658	\ifHy@pdfescapeform	2799, 2809
\HyAnn@AbsPageLabel	2396, 2402, 2448, 2546, 2576, 2612, 2643, 2682, 2716	\ifHy@unicode	2852
\HyAnn@Count	2400, 2401, 2404, 2405, 2406, 2407, 2408, 2409, 2413, 2417	\kv@set@family@handler	161
\HyField@AddToFields	2237, 2456, 2565, 2582, 2621, 2657, 2690, 2723	\ltx@empty	2203, 2204, 2244, 2395, 2396, 2397, 2428
\HyField@afields	2203	\ltx@gobble	2206
\HyField@AfterAuxOpen	2208, 2234, 2269	\ltx@gobbletwo	2207
\HyField@AuxAddToCoFields	2207, 2219, 2231, 2253	\ltx@ifUndefined	3304, 3314
\HyField@AuxAddToFields	2206, 2212, 2228, 2272	\ltx@one	2404
\HyField@cofields	2204	\ltx@ReturnAfterFi	2832
\HyField@FlagsCheckBox	2708	\ltx@zero	2401
\HyField@FlagsChoice	2514	\m@ne	3313, 3328
\HyField@FlagsPushButton	2604, 2636, 2679	\OBJ@OCG@view	143
\HyField@FlagsRadioButton	2510	\p@	2478
\HyField@FlagsSubmit	2637	\pdf@ifdraftmode	2289
\HyField@FlagsText	2443	\PDF@SetupDoc	138
\HyField@PDFChoices	2574	\pdf@strcmp	2305
\HyField@SetKeys	2439, 2480, 2483, 2502, 2589, 2632, 2668, 2703	\PDFForm@Check	2717, 2722, 2924
\HyPat@ObjRef	3211, 3218, 3225, 3232	\PDFForm@List	2577, 2580, 3013
\hyper@anchor	391	\PDFForm@Name	2442, 2503, 2593, 2707, 2838
\hyper@anchorend	391, 3274, 3299, 3310, 3340	\PDFForm@Push	2613, 2619, 2979
\hyper@anchorstart	391, 3274, 3299, 3310, 3340	\PDFForm@Radio	2547, 2553, 3048
		\PDFForm@Reset	2683, 2689, 3174
		\PDFForm@Submit	2644, 2651, 3137
		\PDFForm@Text	2449, 2454, 3087
		\protected@edef	411
		\strip@pt	2959, 3042, 3073, 3124, 3182
		\toks@	3284, 3290, 3355, 3361
		\tw@	3285, 3290, 3356, 3361
		\XR@ext	1210
		\z@	2469, 3022, 3028, 3059, 3146, 3151, 3183, 3188, 3326

<code>\zref@labelbylist</code>	2406	<code>\tl_if_in:NnTF</code>	634
<code>\zref@labelbyprops</code>	2405	<code>\tl_new:N</code>	240, 244, 245, 246,
<code>\zref@refused</code>	2407		247, 270, 271, 272, 273, 274, 275,
<code>\texorpdfstring</code>	22		276, 277, 278, 286, 2280, 2282, 2532
text commands:		<code>\tl_set:Nn</code>	179, 280,
<code>\text_purify:n</code>	6, 320		281, 282, 283, 284, 285, 521, 1665,
<code>\textcolor</code>	2287, 2385, 2390		1669, 1673, 1696, 1831, 1834, 1841,
<code>\textsf</code>	2385, 2390		1844, 1845, 1846, 1847, 1848, 1849,
<code>\the</code>	361,		1854, 1858, 1894, 2281, 2283, 2533
	2405, 2406, 2407, 2408, 2409, 2413,	<code>\tl_set_eq:NN</code>	418
	2417, 2517, 2521, 2524, 3290, 3361	<code>\tl_to_str:n</code>	
<code>\thepage</code>	229		20, 21, 22, 33, 228, 1980, 1994
<code>\ThisShouldNotHappen</code>	2818	token commands:	
<code>\title</code>	2	<code>\token_to_str:N</code>	29
tl commands:		<code>\toks</code>	3285, 3290, 3356, 3361
<code>\c_space_tl</code>	144, 923,		
	954, 959, 964, 2558, 2559, 2566,		
	2742, 2744, 2754, 2756, 2758, 2761,		
	2763, 2959, 2962, 3096, 3124, 3125		
<code>\tl_const:Nn</code>	250		
<code>\tl_gput_right:Nn</code>	361		
<code>\tl_gset:Nn</code>	1680, 1700, 1704		
<code>\tl_if_blank:nTF</code>			
	533, 1919, 1937, 1975, 1989, 2096, 2109		
<code>\tl_if_empty:NnTF</code> ...	412, 486, 541,		
	593, 644, 1266, 2947, 2951, 2960, 3125		
<code>\tl_if_empty:nTF</code>	792, 828,		
	847, 874, 893, 1457, 1513, 1618, 1903		
<code>\tl_if_empty_p:N</code>	1682, 1707		
<code>\tl_if_eq:NnTF</code>	1829		
<code>\tl_if_exist:NnTF</code> ...	3002, 3112, 3116		
<code>\tl_if_exist_p:N</code> ...	2993, 3101, 3103		
<code>\tl_if_head_eq_charcode:nNTF</code> ...			
	673, 690, 707		

U

unicode (hypersetup key)	1176
url (hypersetup key)	7, 1254
<code>\url</code>	4
urlborder (hypersetup key)	11
urlborderstyle (hypersetup key) ..	11, 868
urlcolor (hypersetup key)	758
<code>\Urlfont</code>	4
use commands:	
<code>\use:n</code>	615

V

verbose (hypersetup key)	1186
--------------------------------	------

W

<code>\write</code>	2227, 2230, 2251, 2271
---------------------------	------------------------

X

<code>\x</code>	2481, 2501, 3289, 3291, 3360, 3362
-----------------------	------------------------------------