

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
`url` (*setup key*) new driver offers also boolean keys `link`, `url`, `file`, `run` and `menu` which allow to locally
`file` (*setup key*) disable a link type. So e.g. `\hypersetup{link=false}\ref{abc}` will give a reference
`run` (*setup key*) without link (this is naturally also possible with `\ref*{abc}`). This disables also all
`menu` (*setup key*) hooks of the link type, so the link is for example no longer colored. It also removes the
implicit grouping of the content.

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

```

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

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

```

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

```

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.

```

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.

```
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.

```

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

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.

```

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

```

10.4 GoToR Links files

```

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

```

```

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

```

10.5 Launch links

We define \hyper@linklaunch for naming consistency

```

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

```

```

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

```

The actually command used by hyperref is `\@hyper@launch` which uses a delimited 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 \pdfmeta_if_named_action_allowed:nTF {#1}
632 {
633 \mode_leave_vertical:
634 \pdfdict_put:nnx {l_hyp/annot/A/Named}{N}
635 {\pdf_text_convert:nn{name-print}{#1}}
636 \tl_if_empty:NF \Hy@href@nextactionraw
637 {
638 \str_remove_once:Nn \Hy@href@nextactionraw {/Next}
639 \pdfdict_put:nno{l_hyp/annot/A/Next}{Next}{\Hy@href@nextactionraw}
640 }
641 \pdfannot_link:nxn { Named }
642 {
643 /A
644 <<
645 \pdfdict_use:n { l_hyp/annot/A/Named }
646 >>
647 }
648 {
649 #2
650 \Hy@xspace@end
651 \Hy@VerboseLinkStop
652 }
653 }
654 {
655 \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
656 #2
657 }
658 \group_end:
659 }
660 {#2}
661 }
662

```

11 Link decorations

11.1 Functions to export and select colors

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

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

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

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

```

663 \cs_new_protected:Npn \_hyp_color_export:nnN #1 #2 #3
664 {
665   \tl_if_head_eq_charcode:nNTF {#1}[ %]
666   {
667     \_hyp_colormodel_export:wnnN #1 {#2} #3
668   }
669   {
670     \color_export:nnN {#1} {#2} #3
671   }
672 }
673
674 \cs_new_protected:Npn \_hyp_colormodel_export:wnnN [#1] #2 #3 #4
675 {
676   \color_export:nnnnN {#1}{#2}{#3}#4
677 }
678
679 \cs_generate_variant:Nn \_hyp_color_export:nnN {xVN}

```

(End definition for `_hyp_color_export:nnN` and `_hyp_colormodel_export:nnnN`.)

<code>_hyp_color_select:n</code>	<code>_hyp_color_select:nN {<color>}</code>
-----------------------------------	--

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.

```

680 \cs_new_protected:Npn \_hyp_color_select:n #1
681 {
682   \tl_if_head_eq_charcode:nNTF {#1}[ %]
683   {
684     \_hyp_colormodel_select_aux:wn #1
685   }
686   {
687     \color_select:n {#1}
688   }

```

```

689 }
690
691 \cs_new_protected:Npn \__hyp_colormodel_select_aux:wn [#1] #2
692 {
693   \color_select:nn {#1}{#2}
694 }
695
696 \cs_generate_variant:Nn \__hyp_color_select:n {e}

```

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

```

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

```

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

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

`__hyp_colormodel_set_aux:nwn`

```

697 \cs_new_protected:Npn \__hyp_color_set:nn #1 #2
698 {
699   \tl_if_head_eq_charcode:nNTF {#2}[ %]
700   {
701     \__hyp_colormodel_set_aux:nwn { #1 } #2
702   }
703   {
704     \color_set:nn {#1} {#2}
705   }
706 }
707
708 \cs_new_protected:Npn \__hyp_colormodel_set_aux:nwn #1 [#2] #3
709 {
710   \color_set:nnn {#1}{#2}{#3}
711 }
712
713 \cs_generate_variant:Nn \__hyp_color_set:nn {ne}

```

(End definition for `__hyp_color_set:nn` and `__hyp_colormodel_set_aux:nwn`.)

11.2 Textcolor of links

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

```

714 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
715 {
716   \hook_gput_code:nnn
717     {pdfannot/link/#2/begin}
718     {hyp/color}
719     {
720       \bool_if:cT { l_hyp_annot_color#1_bool }
721       {
722         \group_begin:
723         \color_select:n { hyp/color/#1}
724       }

```



```

725     }
726     \hook_gput_code:nnn
727     {pdfannot/link/#2/end}
728     {hyp/color}
729     {
730       \bool_if:cT { l_hyp_annot_color#1_bool }
731       {
732         \group_end:
733       }
734     }
735   }

```

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

```

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

```

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*)

```

750 \seq_map_inline:Nn \c__hyp_annot_types_seq
751 {
752   \keys_define:nn { hyp / setup }
753   {
754     ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
755     ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
756   }
757 }
758
759 \keys_define:nn { hyp / setup }
760 {
761   ,allcolors .meta:n =
762   {
763     ,urlcolor=#1
764     ,linkcolor=#1
765     ,runcolor=#1
766     ,filecolor=#1
767     ,menucolor=#1
768   }
769   ,allcolors .value_required:n = true
770 }

```

11.3 Style and color of borders

11.3.1 Border color

The border color is set by link type. The color can be set as rgb (default) or cmyk (unusual). This can be set with the `bordercolormodel` key:

`bordercolormodel` (*setup key*)

```
771 \keys_define:nn { hyp / setup }
772 {
773   ,bordercolormodel .choices:nn =
774   {rgb,cmyk}
775   { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
776   ,bordercolormodel .initial:n = {rgb}
777 }

778 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
779 {
780   \keys_define:nn { hyp / setup }
781   {
782     #1bordercolor .code:n =
783     {
784       \tl_if_empty:nTF { ##1 }
785       {
786         \pdfannot_dict_remove:nn
787         {link/#2}
788         { C }
789       }
790       {
791         \__hyp_color_export:xVN {##1}\g__hyp_bordercolormodel_str \l__hyp_tmpa_tl
792         \pdfannot_dict_put:nnx
793         {link/#2}
794         { C }
795         { [\l__hyp_tmpa_tl] }
796       }
797     }
798   }
799 }

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

11.3.2 Borderwidth and -arc

```

814 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
815 {
816   \keys_define:nn { hyp / setup }
817   {
818     #1border .code:n =
819     {
820       \tl_if_empty:nTF { ##1 }
821       {
822         \pdfannot_dict_remove:nn
823         {link/#2}
824         { Border }
825       }
826       {
827         \pdfannot_dict_put:nnn
828         {link/#2}
829         { Border }
830         { [##1] }
831       }
832     }
833   }
834 }
835 \keys_define:nn { hyp / setup }
836 {
837   ,pdfborder .code:n =
838   {
839     \tl_if_empty:nTF { #1 }
840     {
841       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
842       {
843         \pdfannot_dict_remove:nn
844         {link/##2}
845         { Border }
846       }
847     }
848     {
849       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
850       {
851         \pdfannot_dict_put:nnn
852         {link/##2}
853         { Border }
854         { [#1] }
855       }
856     }
857   }
858   ,pdfborder .initial:n = {0~0~1},
859 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 860 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 861 {
runborderstyle (setup key) 862   \keys_define:nn { hyp / setup }
fileborderstyle (setup key)
menuborderstyle (setup key)

```

```

863 {
864     #1borderstyle .code:n =
865     {
866         \tl_if_empty:nTF { ##1 }
867         {
868             \pdfannot_dict_remove:nn
869             {link/#2}
870             { BS }
871         }
872         {
873             \pdfannot_dict_put:nnn
874             {link/#2}
875             { BS }
876             { <<##1>> }
877         }
878     }
879 }
880 }
881 \keys_define:nn { hyp / setup }
882 {
883     ,pdfborderstyle .code:n =
884     {
885         \tl_if_empty:nTF { #1 }
886         {
887             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
888             {
889                 \pdfannot_dict_remove:nn
890                 {link/##2}
891                 { BS }
892             }
893         }
894         {
895             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
896             {
897                 \pdfannot_dict_put:nnn
898                 {link/##2}
899                 { BS }
900                 { <<##1>> }
901             }
902         }
903     }
904     ,pdfborderstyle .initial:n = {},
905 }

```

11.4 ocgcolorlinks

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

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

```

906 \cs_new_protected:Npn \__hyp_ocg_init:
907 {
908     \pdf_object_new:nn { l__hyp_ocg_view_dict_obj } { dict }

```

```

909 \pdf_object_new:nn { l__hyp_ocg_print_dict_obj } { dict }
910 \pdf_object_new:nn { l__hyp_ocg_config_dict_obj } { dict }
911 \pdf_object_new:nn { l__hyp_ocg_ref_array_obj } { array }
912 \pdf_object_write:nx { l__hyp_ocg_ref_array_obj }
913 {
914   \pdf_object_ref:n { l__hyp_ocg_view_dict_obj }
915   \c_space_tl
916   \pdf_object_ref:n { l__hyp_ocg_print_dict_obj }
917 }
918 \pdf_object_write:nn { l__hyp_ocg_view_dict_obj }
919 {
920   /Type/OCG
921   /Name(View)
922   /Usage
923   <<
924     /Print <</PrintState/OFF>>~
925     /View <</ViewState/ON >>~
926   >>
927 }
928 \pdf_object_write:nn { l__hyp_ocg_print_dict_obj }
929 {
930   /Type/OCG
931   /Name(Print)
932   /Usage
933   <<
934     /Print <</PrintState/ON>>~
935     /View <</ViewState/OFF>>~
936   >>
937 }
938 \pdfmanagement_add:nxx { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {l__hyp_ocg_ref_array_obj} }
939 \pdfmanagement_add:nxx { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {l__hyp_ocg_print_dict_obj} }
940 \pdf_object_write:nx { l__hyp_ocg_config_dict_obj }
941 {
942   /OFF[\pdf_object_ref:n { l__hyp_ocg_print_dict_obj }]
943   /AS[
944     <<
945       /Event/View
946       /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
947       /Category[/View]
948     >>
949     <<
950       /Event/Print
951       /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
952       /Category[/Print]
953     >>
954     <<
955       /Event/Export
956       /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
957       /Category[/Print]
958     >>
959   ]
960 }
961 \pdfmanagement_add:nxx { Catalog / OCProperties }{ D }{ \pdf_object_ref:n { l__hyp_ocg_ref_array_obj } }
962 \cs_gset:Npn \__hyp_ocg_init: {}

```

```
963 }
```

(End definition for `_hyp_ocg_init:`.)

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

```
964 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
965 {
966   \hook_gput_code:nnn
967     {pdfannot/link/#2/begin}
968     {hyp/ocg}
969   {
970     \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
971     {
972       \_hyp_ocg_init:
973       \group_begin:
974       \hbox_set:Nw \l__hyp_tmpa_box
975     }
976   }
977   \hook_gput_code:nnn
978     {pdfannot/link/#2/end}
979     {hyp/ocg}
980   {
981     \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
982     {
983       \hbox_set_end:
984       \mbox
985       {
986         \pdf_bdc_obj:nn {0C}{l__hyp_ocg_print_dict_obj}
987         \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
988         \pdf_emc:
989         \pdf_bdc_obj:nn {0C}{l__hyp_ocg_view_dict_obj}
990         \group_begin:
991         \color_select:n { hyp/color/#1 }
992         \box_use_drop:N \l__hyp_tmpa_box
993         \group_end:
994         \pdf_emc:
995       }
996       \group_end:
997     }
998   }
999 }
```

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

```
ocgcolorlink (setup key) 1000 \pdf_version_compare:NnTF < {1.5}
ocgcolorurl (setup key) 1001 {
ocgcolorfile (setup key) 1002   \keys_define:nn { hyp / setup }
ocgcolormenu (setup key) 1003   {
ocgcolorrun (setup key) 1004     ,_ocgcolorlinks .code:n =
1005     {
1006       \msg_warning:nnxx
1007       { hyp }
1008       { ignore-deprecated-or-unknown-option-in-pdf-version }
1009       { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1010     }
```

```

1011     }
1012   }
1013   {
1014     \keys_define:nn { hyp / setup }
1015     {
1016       ,_ocgcolorlinks .meta:n =
1017       {
1018         ocgcolorlink=#1,
1019         ocgcolorurl=#1,
1020         ocgcolorfile=#1,
1021         ocgcolorrun=#1,
1022         ocgcolormenu=#1
1023       }
1024       ,_ocgcolorlinks .default:n = true
1025     }
1026   }
1027
1028   \keys_define:nn { hyp / setup }
1029   {
1030     ,ocgcolorlinks .choice:
1031     ,ocgcolorlinks / true .meta:n =
1032     {
1033       pdfborder      = {0~0~0},
1034       pdfborderstyle = {},
1035       colorlinks     = false,
1036       _ocgcolorlinks = true
1037     }
1038     ,ocgcolorlinks / false .meta:n =
1039     {
1040       _ocgcolorlinks = false
1041     }
1042     ,ocgcolorlinks .default:n = {true}
1043   }
1044
1045   \seq_map_inline:Nn \c__hyp_annot_types_seq
1046   {
1047     \pdf_version_compare:NnTF < {1.5}
1048     {
1049       \keys_define:nn { hyp / setup }
1050       {
1051         ,ocgcolor#1 .code:n=
1052         {
1053           \msg_warning:nnxx
1054           { hyp }
1055           { ignore-deprecated-or-unknown-option-in-pdf-version }
1056           { ocgcolor#1 }
1057           { \pdf_version_major:.\pdf_version_minor: }
1058         }
1059       }
1060     }
1061     {
1062       \keys_define:nn { hyp / setup }
1063       {
1064         ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }

```

```

1065     }
1066   }
1067 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1068 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1069 {
1070   \keys_define:nn { hyp / setup }
1071   {
1072     ,#1highlight .choices:nn =
1073     { /I, /N, /O, /P}
1074     {
1075       \pdfannot_dict_put:nnn
1076       {link/#2}
1077       { H }
1078       { ##1 }
1079     }
1080     ,#1highlight / .code:n =
1081     {
1082       \pdfannot_dict_remove:nn
1083       {link/#2}
1084       { H }
1085     }
1086     }
1087     ,#1highlight / unknown .code:n =
1088     {
1089       \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1090       { pdfhighlight }
1091       { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1092       { \exp_not:n {##1} }
1093     }
1094   }
1095 }
1096 }
1097
1098
1099 \keys_define:nn { hyp / setup }
1100 {
1101   ,pdfhighlight .choices:nn =
1102   { /I, /N, /O, /P}
1103   {
1104     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1105     {
1106       \pdfannot_dict_put:nnn
1107       {link/####2}
1108       { H }
1109       { #1 }
1110     }
1111   }
1112   ,pdfhighlight / .code:n =
1113   {
1114     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop

```



```

1115         {
1116             \pdfannot_dict_remove:nn
1117             {link/##2}
1118             { H }
1119         }
1120     }
1121     ,pdfhighlight .initial:n = {/I},
1122     ,pdfhighlight / unknown .code:n =
1123     {
1124         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1125         { pdfhighlight }
1126         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1127         { \exp_not:n {#1} }
1128     }
1129 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1130 \keys_define:nn { hyp / setup }
hideurl (setup key) 1131 {
hidefile (setup key) 1132     hidelinks .meta:n =
hiderun (setup key) 1133     {
hidemenu (setup key) 1134         ,colorlinks      = false
1135         ,ocgcolorlinks = false
1136         ,pdfborder      = { 0~0~0 }
1137         ,pdfborderstyle=
1138     }
1139 }
1140
1141 \seq_map_inline:Nn \c__hyp_annot_types_seq
1142 {
1143     \keys_define:nn { hyp / setup }
1144     {
1145         hide#1 .meta:n =
1146         {
1147             ,color#1      = false
1148             ,ocgcolor#1   = false
1149             ,#1border     = { 0~0~0 }
1150             ,#1borderstyle =
1151         }
1152     }
1153 }

```

11.7 color schemes and settings

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

```

colorscheme (setup key)
1154 \keys_define:nn { hyp / setup }
1155 {
1156     colorscheme .code:n =

```

```

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

```

12 Keys

12.1 Ignored keys

The following are ignored (with or without warnings)

```

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

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

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

```

```

1196         \Hy@finaltrue
1197         \PassOptionsToPackage{final}{bookmark}
1198     }
1199 }

extension (setup key)
hypertextnames (setup key) 1200 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1201 {
pageanchor (setup key) 1202 ,extension .tl_set:N = \XR@ext
linktoc (setup key) 1203 ,extension .initial:n= pdf
linktocpage (setup key) 1204 ,hypertextnames .choice:
plainpages (setup key) 1205 ,hypertextnames / true .code:n = { \Hy@hypertextnamestrue}
localanchorname (setup key) 1206 ,hypertextnames / false .code:n = { \Hy@hypertextnamesfalse}
linkfileprefix (setup key) 1207 ,hypertextnames .default:n = {true}
1208 ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1209 ,localanchorname .choice:
1210 ,localanchorname / true .code:n = { \Hy@localanchornamestrue }
1211 ,localanchorname / false .code:n = { \Hy@localanchornamefalse }
1212 ,localanchorname .default:n = {true}
1213 ,naturalnames .choice:
1214 ,naturalnames / true .code:n = { \Hy@naturalnamestrue}
1215 ,naturalnames / false .code:n = { \Hy@naturalnamesfalse}
1216 ,naturalnames .default:n = {true}
1217 ,pageanchor .choice:
1218 ,pageanchor / true .code:n = { \Hy@pageanchortrue}
1219 ,pageanchor / false .code:n = { \Hy@pageanchorfalse}
1220 ,pageanchor .default:n = {true}
1221 ,plainpages .choice:
1222 ,plainpages / true .code:n = { \Hy@plainpagestrue}
1223 ,plainpages / false .code:n = { \Hy@plainpagesfalse}
1224 ,plainpages .default:n = {true}
1225 }
1226
1227 \keys_define:nn { hyp / setup }
1228 {
1229 ,linktoc .choices:nn = { none, section, all, page }
1230 {
1231 \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1232 }
1233 ,linktoc / unknown .code:n =
1234 {
1235 \msg_warning:nnxxx { hyp } { unknown-choice }
1236 { linktoc }
1237 { none, section, all, page }
1238 { \exp_not:n {#1} }
1239 }
1240 ,linktocpage .choice:
1241 ,linktocpage / true .meta:n = {linktoc=page}
1242 ,linktocpage / false .meta:n = {linktoc=section}
1243 ,linktocpage .default:n = true
1244 }
1245

```

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

`url (setup key)`

`file (setup key)`

`menu (setup key)`

`run (setup key)`

```

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

1253 \keys_define:nn { hyp / setup }
1254 {
1255   ,baseurl .code:n =
1256   {
1257     \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1258     \tl_if_empty:NTF \l__hyp_tmpa_tl
1259     {
1260       \pdfmanagement_remove:nn {Catalog} { URI }
1261     }
1262     {
1263       \pdfmanagement_add:nnx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1264     }
1265   }
1266   %only false does something ...
1267   ,bookmarks .choice:
1268   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1269   ,bookmarks / true .code:n = {}
1270   ,bookmarks .default:n = {true}
1271   ,bookmarksnumbered .choice:
1272   ,bookmarksnumbered / false .code:n = { \Hy@bookmarksnumberedfalse }
1273   ,bookmarksnumbered / true .code:n = { \Hy@bookmarksnumberedtrue }
1274   ,bookmarksnumbered .default:n = {true}
1275   ,bookmarksopen .choice:
1276   ,bookmarksopen / false .code:n = { \Hy@bookmarksopenfalse }
1277   ,bookmarksopen / true .code:n = { \Hy@bookmarksopentruetrue }
1278   ,bookmarksopen .default:n = {true}
1279   ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1280   ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1281   ,pdfcenterwindow .choice:
1282   ,pdfcenterwindow / false .code:n =
1283   {
1284     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1285   }
1286   ,pdfcenterwindow / true .code:n =
1287   {
1288     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1289   }
1290   ,pdfcenterwindow / .code:n =
1291   {
1292     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1293   }
1294   ,pdfcenterwindow / unknown .code:n =
1295   {
1296     \msg_warning:nnxx { hyp } { no-bool }
1297     { pdfcenterwindow }
1298     { \exp_not:n {#1} }

```

```

1299     }
1300 ,pdfcenterwindow      .default:n = true
1301 ,pdfdirection         .choice:
1302 ,pdfdirection / L2R .code:n =
1303     {
1304         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1305     }
1306 ,pdfdirection / R2L .code:n =
1307     {
1308         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1309     }
1310 ,pdfdirection /      .code:n =
1311     {
1312         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1313     }
1314 ,pdfdirection / unknown .code:n =
1315     {
1316         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1317         { pdfdirection }
1318         { L2R , R2L }
1319         { \exp_not:n {#1} }
1320     }
1321 ,pdfdisplaydoctitle   .choice:
1322 ,pdfdisplaydoctitle / false .code:n =
1323     {
1324         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1325     }
1326 ,pdfdisplaydoctitle / true .code:n =
1327     {
1328         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1329     }
1330 ,pdfdisplaydoctitle   .default:n = true
1331 ,pdfduplex            .choices:nn =
1332 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1333 {
1334     \pdf_version_compare:NnTF > {1.6}
1335     {
1336         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1337         { PrintDuplex } { /#1 }
1338     }
1339     {
1340         \msg_warning:nnxx
1341         {hyp}
1342         {ignore-deprecated-or-unknown-option-in-pdf-version}
1343         {pdfduplex}
1344         {\pdf_version:}
1345     }
1346 }%
1347 ,pdfduplex /          .code:n =
1348     {
1349         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1350     }
1351 ,pdfduplex / unknown .code:n =
1352     {

```

```

1353     \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1354     { pdfduplex }
1355     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1356     { \exp_not:n {#1} }
1357 }
1358 ,pdffitwindow .choice:
1359 ,pdffitwindow / false .code:n =
1360 {
1361     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1362 }
1363 ,pdffitwindow / true .code:n =
1364 {
1365     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1366 }
1367 ,pdffitwindow / .code:n =
1368 {
1369     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1370 }
1371 ,pdffitwindow .default:n = true
1372 ,pdffitwindow / unknown .code:n =
1373 {
1374     \msg_warning:nxxx { hyp } { no-bool }
1375     { pdffitwindow }
1376     { \exp_not:n {#1} }
1377 }
1378 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1379 ,pdflinkmargin .initial:n = {1pt}
1380 ,pdfmenubar .choice:
1381 ,pdfmenubar / true .code:n =
1382 {
1383     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1384 }
1385 ,pdfmenubar / false .code:n =
1386 {
1387     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1388     { HideMenubar } { true }
1389 }
1390 ,pdfmenubar / .code:n =
1391 {
1392     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1393 }
1394 ,pdfmenubar .default:n = true
1395 ,pdfmenubar / unknown .code:n =
1396 {
1397     \msg_warning:nxxx { hyp } { no-bool }
1398     { pdfmenubar }
1399     { \exp_not:n {#1} }
1400 }
1401 ,pdfnewwindow .choice:
1402 ,pdfnewwindow / true .code:n =
1403 {
1404     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1405     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1406 }

```

```

1407 ,pdfnewwindow / false .code:n =
1408 {
1409     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1410     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1411 }
1412 ,pdfnewwindow / .code:n =
1413 {
1414     \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1415     \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1416 }
1417 ,pdfnonfullscreenpagemode .choices:nn =
1418 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1419 {
1420     \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1421     { NonFullScreenPageMode } {/#1}
1422 }
1423 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1424 {
1425     \pdf_version_compare:NnTF < {1.6}
1426     {
1427         %message
1428     }
1429     {
1430         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1431         {NonFullScreenPageMode}{/UseAttachments}
1432     }
1433 }
1434 ,pdfnonfullscreenpagemode / .code:n =
1435 {
1436     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1437 }
1438 ,pdfnonfullscreenpagemode / unknown .code:n =
1439 {
1440     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1441     { pdfnonfullscreenpagemode }
1442     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1443     { \exp_not:n {#1} }
1444 }
1445 ,pdfnumcopies .code:n =
1446 {
1447     \pdf_version_compare:NnTF > {1.6}
1448     {
1449         \tl_if_empty:NTF {#1}
1450         {
1451             \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1452         }
1453         {
1454             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1455             {NumCopies}{#1}
1456         }
1457     }
1458     {
1459         \msg_warning:nnxx
1460         {hyp}

```

```

1461         {ignore-deprecated-or-unknown-option-in-pdf-version}
1462         {pdfnumcopies}
1463         {\pdf_version:}
1464     }
1465 }
1466 ,pdfpagelayout .choices:nn =
1467 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1468 { \pdfmanagement_add:nnx {Catalog} { PageLayout }{ /#1 } }
1469 ,pdfpagelayout / .code:n =
1470 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1471 ,pdfpagelayout / unknown .code:n =
1472 {
1473     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1474     { pdfpagelayout }
1475     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1476     { \exp_not:n {#1} }
1477 }
1478 ,pdfpagemode .choices:nn =
1479 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1480 { \pdfmanagement_add:nnx {Catalog} { PageMode }{ /#1 } }
1481 ,pdfpagemode / UseAttachments .code:n =
1482 {
1483     \pdf_version_compare:NnTF > {1.5}
1484     {
1485         \pdfmanagement_add:nnx {Catalog} { PageMode }{ /UseAttachments }
1486     }
1487     {
1488         \msg_warning:nnxx
1489         {hyp}
1490         {ignore-deprecated-or-unknown-value-in-pdf-version}
1491         {UseAttachments}
1492         {\pdf_version:}
1493     }
1494 }
1495 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1496 ,pdfpagemode / unknown .code:n =
1497 {
1498     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1499     { pdfpagemode }
1500     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1501     { \exp_not:n {#1} }
1502 }
1503 ,pdfpagescrop .code:n =
1504 {
1505     \tl_if_empty:nTF %or blank?
1506     {
1507         \pdfmanagement_remove:nn {Pages} { CropBox }
1508     }
1509     {
1510         \pdfmanagement_add:nnx {Pages} { CropBox } { [#1] }
1511     }
1512 }
1513 ,pdfpicktraybypdfsize .choice:
1514 ,pdfpicktraybypdfsize / true .code:n =

```



```

1515 {
1516   \pdf_version_compare:NnTF > {1.6}
1517   {
1518     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1519     { PickTrayByPDFSize } { true }
1520   }
1521   {
1522     \msg_warning:nnxx
1523     {hyp}
1524     {ignore-deprecated-or-unknown-option-in-pdf-version}
1525     {pdfpicktraybypdfsize}
1526     {\pdf_version:}
1527   }
1528 }
1529 ,pdfpicktraybypdfsize / false .code:n =
1530 {
1531   \pdf_version_compare:NnTF > {1.6}
1532   {
1533     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1534     { PickTrayByPDFSize } { false }
1535   }
1536   {
1537     \msg_warning:nnxx
1538     {hyp}
1539     {ignore-deprecated-or-unknown-option-in-pdf-version}
1540     {pdfpicktraybypdfsize}
1541     {\pdf_version:}
1542   }
1543 }
1544 ,pdfpicktraybypdfsize / .code:n =
1545 {
1546   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1547 }
1548 ,pdfpicktraybypdfsize / unknown .code:n =
1549 {
1550   \msg_warning:nnxx { hyp } { no-bool }
1551   { picktraybypdfsize }
1552   { \exp_not:n {#1} }
1553 }
1554 ,pdfprintarea .choices:nn =
1555 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1556 {
1557   \pdf_version_compare:NnTF < {2.0}
1558   {
1559     \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1560     { PrintArea } { /#1 }
1561   }
1562   {
1563     \msg_warning:nnxx
1564     {hyp}
1565     {ignore-deprecated-or-unknown-option-in-pdf-version}
1566     {pdfprintarea}
1567     {\pdf_version:}
1568   }

```

```

1569     }%
1570 ,pdfprintarea / .code:n =
1571 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1572 ,pdfprintarea / unknown .code:n =
1573 {
1574     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1575     { pdfprintarea }
1576     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1577     { \exp_not:n {#1} }
1578 }
1579 ,pdfprintclip .choices:nn =
1580 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1581 {
1582     \pdf_version_compare:NnTF < {2.0}
1583     {
1584         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1585         { PrintClip } { /#1 }
1586     }
1587     {
1588         \msg_warning:nnxx
1589         {hyp}
1590         {ignore-deprecated-or-unknown-option-in-pdf-version}
1591         {pdfprintclip}
1592         {\pdf_version:}
1593     }
1594 }%
1595 ,pdfprintclip / .code:n =
1596 {
1597     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1598 }
1599 ,pdfprintclip / unknown .code:n =
1600 {
1601     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1602     { pdfprintclip }
1603     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1604     { \exp_not:n {#1} }
1605 }
1606 ,pdfprintpagerange .code:n =
1607 {
1608     \pdf_version_compare:NnTF > {1.6}
1609     {
1610         \tl_if_empty:nTF { #1}
1611         {
1612             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1613             { PrintPageRange }
1614         }
1615         {
1616             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1617             {PrintPageRange}{[#1]}
1618         }
1619     }
1620     {
1621         \msg_warning:nnxx
1622         {hyp}

```

```

1623         {ignore-deprecated-or-unknown-option-in-pdf-version}
1624         {pdfprintpagerange}
1625         {\pdf_version:}
1626     }
1627 }
1628 ,pdfprintscaling .choices:nn =
1629 { None, AppDefault }
1630 {
1631     \pdf_version_compare:NnTF > {1.5}
1632     {
1633         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1634         { PrintScaling } { /#1 }
1635     }
1636     {
1637         \msg_warning:nnxx
1638         {hyp}
1639         {ignore-deprecated-or-unknown-option-in-pdf-version}
1640         {pdfprintscaling}
1641         {\pdf_version:}
1642     }
1643 }%
1644 ,pdfprintscaling / .code:n =
1645 {
1646     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1647 }
1648 ,pdfprintscaling / unknown .code:n =
1649 {
1650     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1651     { pdfprintarea }
1652     { None, AppDefault }
1653     { \exp_not:n {#1} }
1654 }
1655 ,pdfremotestartview .code:n =
1656 {
1657     \tl_set:Nx \l__hyp_tmpa_tl {#1-null-null-null~}
1658     \exp_args:NNV
1659     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1660     {
1661         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1662     }
1663     {
1664         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1665         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1666     }
1667 }
1668 ,pdfremotestartview .initial:n = {Fit}
1669 % pdfstartpage is special as it shares code with pdfstartview
1670 ,pdfstartpage .code:n =
1671 {
1672     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1673     \bool_if:nTF
1674     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1675     {
1676         \pdfmanagement_remove:nn {Catalog} { OpenAction }

```

```

1677     }
1678     {
1679         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1680         {
1681             [\pdf_object_pageref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1682         }
1683     }
1684 }
1685 ,pdfstartpage .initial:n =1
1686 ,pdfstartview .code:n =
1687 {
1688     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~}
1689     \exp_args:NNV
1690     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa
1691     {
1692         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1693     }
1694     {
1695         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1696         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1697     }
1698     \bool_if:NTF
1699     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1700     {
1701         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1702     }
1703     {
1704         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1705         {
1706             [\pdf_object_pageref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1707         }
1708     }
1709 }
1710 ,pdfstartview .initial:n = Fit
1711 ,pdftoolbar .choice:
1712 ,pdftoolbar / true .code:n =
1713 {
1714     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1715 }
1716 ,pdftoolbar / false .code:n =
1717 {
1718     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
1719     { HideToolbar } { true }
1720 }
1721 ,pdftoolbar / true .code:n =
1722 {
1723     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1724 }
1725 ,pdftoolbar .default:n = true
1726 ,pdftoolbar / unknown .code:n =
1727 {
1728     \msg_warning:nnxx { hyp } { no-bool }
1729     { pdftoolbar }
1730     { \exp_not:n {#1} }

```

```

1731     }
1732 % pdfview see below.
1733 ,pdfviewarea .choices:nn =
1734 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1735 {
1736   \pdf_version_compare:NnTF < {2.0}
1737   {
1738     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1739     { ViewArea } { /#1 }
1740   }
1741   {
1742     \msg_warning:nnxx
1743     {hyp}
1744     {ignore-deprecated-or-unknown-option-in-pdf-version}
1745     {pdfviewarea}
1746     {\pdf_version:}
1747   }
1748 }%
1749 ,pdfviewarea / .code:n =
1750 {
1751   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1752 }
1753 ,pdfviewarea / unknown .code:n =
1754 {
1755   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1756   { pdfviewarea }
1757   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1758   { \exp_not:n {#1} }
1759 }
1760 ,pdfviewclip .choices:nn =
1761 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1762 {
1763   \pdf_version_compare:NnTF < {2.0}
1764   {
1765     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1766     { ViewClip } { /#1 }
1767   }
1768   {
1769     \msg_warning:nnxx
1770     {hyp}
1771     {ignore-deprecated-or-unknown-option-in-pdf-version}
1772     {pdfviewclip}
1773     {\pdf_version:}
1774   }
1775 }%
1776 ,pdfviewclip / .code:n =
1777 {
1778   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
1779 }
1780 ,pdfviewclip / unknown .code:n =
1781 {
1782   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1783   { pdfviewclip }
1784   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }

```

```

1785     { \exp_not:n {#1} }
1786   }
1787   ,pdfwindowui .choice:
1788   ,pdfwindowui / true .code:n =
1789   {
1790     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
1791   }
1792   ,pdfwindowui / false .code:n =
1793   {
1794     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1795     { HideWindowUI } { true }
1796   }
1797   ,pdfwindowui / .code:n =
1798   {
1799     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
1800   }
1801   ,pdfwindowui / unknown .code:n =
1802   {
1803     \msg_warning:nnxx { hyp } { no-bool }
1804     { pdfwindowui }
1805     { \exp_not:n {#1} }
1806   }
1807   ,pdfwindowui .default:n = true
1808 }

```

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

```

1809 \keys_define:nn { hyp / setup }
1810 {
1811   ,pdfview .code:n =
1812   {
1813     \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
1814     \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } } {
1815       {
1816         { xyz }
1817         {
1818           \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
1819           {
1820             \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
1821             \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
1822             {
1823               \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1824             }
1825             {
1826               \tl_set:Nx \l__hyp_dest_pdfview_tl
1827               {
1828                 \fp_eval:n { \l__hyp_tmpa_tl * 100 }
1829               }
1830             }
1831           }
1832           {
1833             \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1834           }
1835         }
1836         { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }

```

```

1837 { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
1838 { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh}}
1839 { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv}}
1840 { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
1841 { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
1842 { fitr }
1843 {
1844   \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
1845   {
1846     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
1847   }
1848   {
1849     %ensure 4 values ...
1850     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
1851     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1852     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1853     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1854     \hbox_set_to_wd:Nnn \l__hyp_dest_box
1855     {
1856       \fp_eval:n
1857       {
1858         round
1859         (
1860           abs
1861           (
1862             \seq_item:Nn\l__hyp_tmpa_seq{4}
1863             -
1864             (\seq_item:Nn\l__hyp_tmpa_seq{2})
1865           ),
1866           3
1867         )
1868       }bp
1869     }{}
1870     \box_set_dp:Nn \l__hyp_dest_box
1871     {
1872       \fp_eval:n
1873       {
1874         round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
1875       }bp
1876     }
1877     \box_set_ht:Nn \l__hyp_dest_box
1878     {
1879       \seq_item:Nn\l__hyp_tmpa_seq{5}bp
1880     }
1881   }
1882 }
1883 }
1884 {
1885   \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
1886   \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
1887 }
1888 }
1889 ,pdfview .initial:n = {xyz}
1890 }

```

12.3 “MetaData keys”

The following keys are relevant for the metadata: the info dictionary and the xmp-metadata.

`pdflang (setup key)` `pdflang` should be deprecated.

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

12.3.1 “info keys”

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

```
pdfcreator (setup key) 1905 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
pdfsubject (setup key) 1906 {
pdfproducer (setup key) 1907   \keys_define:nn { hyp / setup }
pdfkeywords (setup key) 1908   {
1909     pdf#1 .code:n =
1910     {
1911       \tl_if_blank:nTF {##1}
1912       {
1913         \pdfmanagement_remove:nn {Info}{#2}
1914       }
1915       {
1916         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
1917         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
1918         {
1919           \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
1920         }
1921       }
1922       \__hyp_store_metadata:nn {pdf#1}{##1}
1923     }
1924   }
1925   \keys_define:nn { hyp / info }
1926   {
1927     #2 .code:n =
1928     {
1929       \tl_if_blank:nTF {##1}
1930       {
1931         \pdfmanagement_remove:nn {Info}{#2}
1932       }
1933       {
```



```

1934         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
1935         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
1936         {
1937             \pdfmanagement_add:nxx {Info}{#2}{\l__hyp_tmpa_str}
1938         }
1939     }
1940     \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
1941 }
1942 ,unknown .code:n =
1943 {
1944     \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
1945     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
1946     {
1947         \exp_args:Nxx
1948         \pdfmanagement_add:nxx {Info}
1949         {\pdf_text_convert:nV {name } \l_keys_key_str}{\l__hyp_tmpa_str}
1950     }
1951 }
1952 }
1953 }
1954 \__hyp_setup_info_key:nn {author} {Author}
1955 \__hyp_setup_info_key:nn {title} {Title}
1956 \__hyp_setup_info_key:nn {producer} {Producer}
1957 \__hyp_setup_info_key:nn {creator} {Creator}
1958 % ignored key: addtopdfcreator
1959 \__hyp_setup_info_key:nn {subject} {Subject}
1960 \__hyp_setup_info_key:nn {keywords} {Keywords}

```

pdfcreationdate (setup key) These keys are not really needed. We store them too in the container. CreationDate and
pdfmoddate (setup key) ModDate should not use the hex encoding.

```

1961 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
1962 {
1963     \keys_define:nn { hyp / setup }
1964     {
1965         pdf#1 .code:n =
1966         {
1967             \tl_if_blank:nTF {##1}
1968             {
1969                 \pdfmanagement_remove:nn {Info}{#2}
1970             }
1971             {
1972                 \pdfmanagement_add:nxx {Info}{#2}{(\tl_to_str:n {##1})}
1973             }
1974             \__hyp_store_metadata:nn {pdf#1}{##1}
1975         }
1976     }
1977     \keys_define:nn { hyp / info }
1978     {
1979         #2 .code:n =
1980         {
1981             \tl_if_blank:nTF {##1}
1982             {
1983                 \pdfmanagement_remove:nn {Info}{#2}
1984             }

```

```

1985         {
1986         \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
1987         }
1988     \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
1989 }
1990 }
1991 }
1992
1993 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
1994 \__hyp_setup_info_date_key:nn {moddate} {ModDate}

```

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

```

1995 \keys_define:nn { hyp / setup }
1996 {
1997     ,pdftrapped .code:n =
1998     {
1999         \exp_args:Nne
2000         \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2001     }
2002     ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2003     {
2004         \pdfmanagement_add:nnx {Info}{Trapped}
2005         {/
2006         \str_uppercase:f { \str_head:n { #1 } }
2007         \str_lowercase:f { \str_tail:n { #1 } }
2008         }
2009         \__hyp_store_metadata:nx {pdftrapped}
2010         {
2011             \str_uppercase:f { \str_head:n { #1 } }
2012             \str_lowercase:f { \str_tail:n { #1 } }
2013         }
2014     }
2015     ,_pdftrapped / unknown .code:n =
2016     {
2017         \msg_warning:nnxxx { hyp } { unknown-choice }
2018         { pdftrapped }
2019         { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2020         { \exp_not:n {#1} }
2021     }
2022 }

```

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

```

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

```

Now we set some default values

```

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

```

12.4 hyperxmp keys

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

```
2034 \clist_map_inline:nn
2035 {
2036   ,pdfcopyright
2037   ,pdftype
2038   ,pdflicenseurl
2039   ,pdfauthortitle
2040   ,pdfcaptionwriter
2041   ,pdfmetalang
2042   ,pdfapart
2043   ,pdfaconformance
2044   ,pdfuapart
2045   ,pdfxstandard
2046   ,pdfsource
2047   ,pdfdocumentid
2048   ,pdfinstanceid
2049   ,pdfversionid
2050   ,pdfrendition
2051   ,pdfpublication
2052   ,pdfpubtype
2053   ,pdfbytes
2054   ,pdfnumpages
2055   ,pdfissn
2056   ,pdfeissn
2057   ,pdfisbn
2058   ,pdfbookedition
2059   ,pdfpublisher
2060   ,pdfvolumenum
2061   ,pdfissuenum
2062   ,pdfpagerange
2063   ,pdfdoi
2064   ,pdfurl
2065   ,pdfidentifier
2066   ,pdfsubtitle
2067   ,pdfpubstatus
2068   ,pdfcontactaddress
2069   ,pdfcontactcity
2070   ,pdfcontactregion
2071   ,pdfcontactpostcode
2072   ,pdfcontactcountry
2073   ,pdfcontactphone
2074   ,pdfcontactemail
2075   ,pdfcontacturl
2076   ,pdfdate
2077 }
2078 {
2079   \keys_define:nn { hyp / setup }
2080   {
2081     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2082   }
2083 }
```

12.5 Transitions

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

```

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

```

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

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

style	one of Split, Blinds, Box, Dissolve, Glitter, R, Fly, I, Uncover, Fade
duration	a number, describes the duration of the transition
direction	H (horizontal, only Split, Blinds, Box, I, Uncover, Fade) V (vertical, only Split, Blinds, Box, I, Uncover, Fade) 0 (left to right, only Wipe, Fly) 90 (bottom to top, only Wipe, Fly) 180 (right to left, only Wipe, Fly) 270 (top to bottom, only Wipe, Fly) 315 (top left to bottom right, only Wipe, Fly) 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

```

2097 \keys_define:nn { hyp / setup }
2098 {
2099   pdfpagetransition .code:n =
2100   {
2101     \tl_if_blank:nTF {#1}
2102     {
2103       \pdfmanagement_remove:nn {Page}{Trans}
2104     }
2105     {

```

```

2106         \group_begin:
2107         \keys_set:nn { hyp / trans }{style=R,#1}
2108         \pdf_object_now:nx { dict }
2109         {
2110             \pdfdict_use:n {l__hyp_page/Trans}
2111         }
2112         \pdfmanagement_add:nnx {Page}{Trans}{\pdf_object_last:}
2113         \group_end:
2114     }
2115 }
2116 }
2117 \keys_define:nn { hyp / trans }
2118 {
2119     ,style .choices:nn =
2120     {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2121     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2122     ,style / unknown .code:n =
2123     {
2124         \msg_warning:nnxxx { hyp } { unknown-choice }
2125         { trans / style }
2126         { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2127         { \exp_not:n {#1} }
2128     }
2129     ,duration .code:n =
2130     {
2131         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}
2132     }
2133     ,direction .choices:nn =
2134     {H,V}
2135     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2136     ,direction .choices:nn =
2137     {0,90,180,270,315}
2138     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ #1 } }
2139     ,direction / None .code:n =
2140     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ /None } }
2141     ,direction / unknown .code:n =
2142     {
2143         \msg_warning:nnxxx { hyp } { unknown-choice }
2144         { trans / direction }
2145         {
2146             H~(horizontal,~only~Split,~Blinds),
2147             V~(vertical,~only~Split,~Blinds),
2148             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2149             90~(bottom~to~top,~only~Wipe),
2150             180~(right~to~left,~only~Wipe),
2151             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2152             315~(top~left~to~bottom,~only~Glitter),
2153             None~(only~Fly)
2154         }
2155         { \exp_not:n {#1} }
2156     }
2157     ,motion .choices:nn =
2158     {I,0}
2159     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }

```

```

2160 ,motion / unknown .code:n =
2161 {
2162   \msg_warning:nnxxx { hyp } { unknown-choice }
2163   { trans / motion }
2164   { I~(inwards) , O~(outwards) }
2165   { \exp_not:n {#1} }
2166 }
2167 ,scale .code:n =
2168 { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2169 ,opaque .choices:nn = {true,false}
2170 { \pdfdict_put:nnn { l__hyp_page/Trans }{ B } { #1 } }
2171 ,opaque / unknown .code:n =
2172 {
2173   \msg_warning:nnxxx { hyp } { unknown-choice }
2174   { trans / B }
2175   { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2176   { \exp_not:n {#1} }
2177 }
2178 % try to set unknown keys as style
2179 ,unknown .code:n =
2180 {
2181   % warning ...
2182   \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2183 }
2184 }

```

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

```

2185 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}

```

Unfinished Form field code

```

2186 \NewDocumentCommand \MakeFieldObject { m m }
2187 {
2188   \pdf_xform_new:nnn { #2 }{ } { #1 }
2189 }
2190
2191
2192 \prop_new:N \g__hyp_AcroForm_CoFields_prop
2193 \prop_new:N \g__hyp_AcroForm_Fields_prop
2194
2195 \let\HyField@afields\ltx@empty
2196 \let\HyField@cofields\ltx@empty
2197 %% UF test for old pdftex removed
2198 %\let\HyField@AuxAddToFields\ltx@gobble
2199 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2200 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2201
2202 % the value doesn't matter, but with a prop we avoid duplicates and it is
2203 % clearly faster than removing them from a sequence
2204 \def\HyField@AuxAddToFields#1
2205 {
2206   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2207 }%
2208
2209 %fields with empty key get a value too -- lets hope that
2210 %this give the expected behaviour

```

```

2211 \def\HyField@AuxAddToCoFields #1 #2
2212 {
2213   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2214 }
2215
2216 \Hy@AtBeginDocument
2217 {
2218   \if@filesw
2219     \immediate\write\@mainaux{%
2220       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2221     }%
2222     \immediate\write\@mainaux{%
2223       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2224     }%
2225   \fi
2226   \let\HyField@AfterAuxOpen\@firstofone
2227 }%
2228
2229 \def\HyField@AddToFields
2230 {
2231   \exp_args:Nx\HyField__hypAddToFields
2232   {
2233     %\pdf_link_last:
2234     \pdfannot_box_last:
2235   }
2236   \ifx\Fld@calculate@code\ltx@empty
2237   \else
2238     \begingroup
2239     \Hy@safe@activetrue
2240     \edef\Hy@temp{%
2241       \endgroup
2242       \if@filesw
2243         \write\@mainaux
2244         {
2245           \string\HyField@AuxAddToCoFields
2246           {
2247             \Fld@calculate@sortkey
2248           }
2249           {
2250             % \pdf_link_last:
2251             \pdfannot_box_last:
2252           }
2253         }
2254       \fi
2255     }%
2256     \Hy@temp
2257   \fi
2258 }%
2259
2260 \def\HyField__hypAddToFields#1{
2261   \HyField@AfterAuxOpen{%
2262     \if@filesw
2263       \write\@mainaux{%
2264         \string\HyField@AuxAddToFields{#1}%

```

```

2265     }%
2266   \fi
2267 }%
2268 }%
2269
2270 \ExplSyntaxOff
2271 \ExplSyntaxOn
2272 \tl_new:N \l__hyp_CheckmarkYes_tl
2273 \tl_set:Nn \l__hyp_CheckmarkYes_tl { __hyp_xform_CheckMarkYes }
2274 \tl_new:N \l__hyp_CheckmarkOff_tl
2275 \tl_set:Nn \l__hyp_CheckmarkOff_tl { __hyp_xform_CheckMarkOff }
2276
2277 \def\@Form[#1]
2278 {
2279   \ifundefined{textcolor}{\let\textcolor\@gobble}{}
2280   \kvsetkeys{Form}{#1}
2281   \pdf@ifdraftmode{}
2282   {
2283     \Hy@FormObjects
2284     \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2285     {
2286       \pdfmanagement_add:nxx { Catalog / AcroForm } { Fields }{##1}
2287       %\pdfmanagement_show:n { Catalog / AcroForm }
2288     }
2289     \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2290     {
2291       \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2292       {
2293         \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2294       }
2295       \seq_sort:Nn \l__hyp_tmpa_seq
2296       {
2297         \int_compare:nNnTF { \pdf@strcmp{##1}{##2} } > { 0 }
2298         { \sort_return_swapped: }
2299         { \sort_return_same: }
2300       }
2301       \seq_map_inline:Nn \l__hyp_tmpa_seq
2302       {
2303         \pdfmanagement_add:nxx { Catalog / AcroForm }
2304         { CO }
2305         {
2306           \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2307         }
2308       }
2309     }
2310     \pdfmanagement_add:nxx {Catalog / AcroForm/DR/Font }
2311     {ZaDb} {\pdf_object_ref:n \l__hyp_font_zapfdingbats_obj} }
2312     \pdfmanagement_add:nxx {Catalog / AcroForm/DR/Font }
2313     {Helv} {\pdf_object_ref:n \l__hyp_font_helvetica_obj} }
2314     \pdfmanagement_add:nxx {Catalog /AcroForm}
2315     {DA}{(/Helv~10~Tf~0~g)}
2316     \prop_if_in:NnTF\g_pdfmeta_standard_prop {form_no_NeedAppearance}
2317     {
2318       \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}

```



```

2319     }
2320     {
2321         \legacy_if:nT { HyField@NeedAppearances }
2322         {
2323             \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2324         }
2325     }
2326 }
2327 \MakeFieldObject
2328 {
2329     \group_begin:
2330     \fontfamily{pzd}
2331     \fontencoding{U}
2332     \fontseries{m}
2333     \fontshape{n}
2334     \selectfont
2335     \char123
2336     \group_end:
2337 }
2338 {__hyp_xform_Ding}
2339 \MakeFieldObject
2340 {
2341     \group_begin:
2342     \fontfamily{pzd}
2343     \fontencoding{U}
2344     \fontseries{m}
2345     \fontshape{n}
2346     \selectfont
2347     \phantom{\char123}
2348     \group_end:
2349 }
2350 {__hyp_xform_DingOff}
2351 \MakeFieldObject
2352 {
2353     \group_begin:
2354     \fontfamily{pzd}
2355     \fontencoding{U}
2356     \fontseries{m}
2357     \fontshape{n}
2358     \selectfont
2359     \char51
2360     \group_end:
2361 }
2362 {__hyp_xform_CheckMarkYes}
2363 \MakeFieldObject
2364 {
2365     \group_begin:
2366     \fontfamily{pzd}
2367     \fontencoding{U}
2368     \fontseries{m}
2369     \fontshape{n}
2370     \selectfont
2371     \phantom{\char51} %perhaps xetex needs some small glyph ..
2372     \group_end:

```

```

2373     }
2374     {__hyp_xform_CheckMarkOff}
2375 \MakeFieldObject
2376     {
2377         \fbox{\textcolor{yellow}{\textsf{Submit}}}} %color?
2378     }
2379     {__hyp_xform_Submit}
2380 \MakeFieldObject
2381     {
2382         \fbox{\textcolor{yellow}{\textsf{SubmitP}}}} %color?
2383     }
2384     {__hyp_xform_SubmitP}
2385 }
2386 \ExplSyntaxOff
2387 \let\@endForm\ltx@empty
2388 \let\HyAnn@AbsPageLabel\ltx@empty
2389 \let\Fld@pageobjref\ltx@empty
2390
2391 \ExplSyntaxOn
2392 \newcount\HyAnn@Count
2393 \HyAnn@Count=\ltx@zero
2394 \def\HyAnn@AbsPageLabel
2395 {
2396     \global\advance\HyAnn@Count by\ltx@one
2397     %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2398     %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2399     %\zref@refused{HyAnn@\the\HyAnn@Count}%
2400     \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
2401     \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
2402 }%
2403 \def\Fld@pageobjref
2404 {
2405     \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
2406     {
2407         /P~\pdf_object_pageref:n
2408         {
2409             \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
2410         }
2411     }
2412 }
2413 \ExplSyntaxOff
2414 \ExplSyntaxOn
2415 %% check if the attr should be set through
2416 %% hooks.
2417 %% check if options are missing.
2418 \def\@TextField[#1]#2{% parameters, label
2419     \def\Fld@name{#2}%
2420     \let\Fld@default\ltx@empty
2421     \let\Fld@value\@empty
2422     \def\Fld@width{\DefaultWidthofText}%
2423     \def\Fld@height{%
2424         \ifFld@multiline
2425             \DefaultHeightofTextMultiline
2426         \else

```

```

2427     \DefaultHeightofText
2428     \fi
2429 }%
2430 \begingroup
2431   \expandafter\HyField@SetKeys\expandafter{%
2432     \DefaultOptionsofText,#1%
2433   }%
2434   \PDFForm@Name
2435   \HyField@FlagsText
2436   \ifFld@hidden\def\Fld@width{1sp}\fi
2437   \ifx\Fld@value\empty\def\Fld@value{\Fld@default}\fi
2438   \LayoutTextField{#2}{%
2439     \leavevmode
2440     \HyAnn@AbsPageLabel
2441     \Hy@escapeform\PDFForm@Text
2442     \pdfannot_box:nnnn
2443       {\Fld@width}
2444       {\Fld@height}
2445       {Opt} %is this correct?
2446       {\PDFForm@Text}
2447     \MakeTextField{\Fld@width}{\Fld@height}
2448     \HyField@AddToFields
2449   }%
2450 \endgroup
2451 }
2452 \providecommand\@curropt{}
2453 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2454   \def\Fld@name{#2}
2455   \let\Fld@default\relax
2456   \let\Fld@value\relax
2457   \def\Fld@width{\DefaultWidthofChoiceMenu}
2458   \def\Fld@height{\DefaultHeightofChoiceMenu}
2459   \begingroup
2460     \Fld@menulength=0 %
2461     \@tempdima\z@
2462     \clist_map_variable:nNn { #3 } \@curropt
2463     %\@for\@curropt:=#3\do
2464     {%
2465       \expandafter\Fld@checkequals\@curropt==\\%
2466       \Hy@StepCount\Fld@menulength
2467       \settowidth{\@tempdimb}{\@currDisplay}%
2468       \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2469     }%
2470   \advance\@tempdima by~15\p@
2471   \begingroup
2472     \HyField@SetKeys{#1}
2473   \edef\x{\endgroup
2474     \noexpand\expandafter
2475     \noexpand\HyField@SetKeys
2476     \noexpand\expandafter{%
2477       \expandafter\noexpand\csname DefaultOptionsof%
2478         \ifFld@radio
2479           Radio%
2480         \else

```

```

2481         \ifFld@combo
2482         \ifFld@popdown
2483             PopdownBox%
2484         \else
2485             ComboBox%
2486         \fi
2487     \else
2488         ListBox%
2489     \fi
2490 \fi
2491 \endcsname
2492 }%
2493 }\x
2494 \HyField@SetKeys{#1}%
2495 \PDFForm@Name
2496 \ifFld@hidden\def\Fld@width{1sp}\fi
2497 \ifx\Fld@value\relax
2498     \let\Fld@value\Fld@default
2499 \fi
2500 \LayoutChoiceField{#2}{%
2501     \ifFld@radio
2502         \HyField@FlagsRadioButton
2503         \__hypRadio{#3}%
2504     \else
2505         \beginngroup
2506         \HyField@FlagsChoice
2507         \ifdim\Fld@width<\@tempdima
2508             \ifdim\@tempdima<1cm\@tempdima1cm\fi
2509             \edef\Fld@width{\the\@tempdima}%
2510         \fi
2511         \ifFld@combo
2512         \else
2513             \@tempdima=\the\Fld@menulength\Fld@charsize
2514             \advance\@tempdima by~\Fld@borderwidth bp %
2515             \advance\@tempdima by~\Fld@borderwidth bp %
2516             \edef\Fld@height{\the\@tempdima}%
2517         \fi
2518         \__hypListbox{#3}%
2519     \endgroup
2520 \fi
2521 }%
2522 \endgroup
2523 }
2524 \tl_new:N \l__hyp_RadioYes_tl
2525 \tl_set:Nn \l__hyp_RadioYes_tl { __hyp_xform_Ding }
2526 \def\__hypRadio#1{%
2527     \Fld@listcount=0~%
2528     %\show\Fld@default
2529     \EdefEscapeName\Fld@default{\Fld@default}%
2530     \clist_map_variable:nNn { #1 } \@curropt
2531     %\@for\@curropt:=#1\do
2532     {%
2533         \expandafter\Fld@checkequals\@curropt==\\%
2534         \EdefEscapeName\currValue{\@currValue}%

```

```

2535 \Hy@StepCount\Fld@listcount
2536 \@currDisplay\space
2537 \leavevmode
2538 \HyAnn@AbsPageLabel
2539 \Hy@escapeform\PDFForm@Radio
2540 \pdfannot_box:nnnn
2541 {\Fld@width}
2542 {\Fld@height}
2543 {Opt} %is this correct?
2544 {
2545 \PDFForm@Radio
2546 /AP
2547 <<
2548 /N
2549 <<
2550 /\@currValue\c_space_tl \pdf_xform_ref:o {__hyp_xform_Ding}
2551 %/Off \c_space_tl \pdf_xform_ref:n {__hyp_xform_DingOff} %hm
2552 >>
2553 >>
2554 }
2555 {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2556 \int_compare:nNtT { \Fld@listcount } = { 1 }
2557 { \HyField@AddToFields }
2558 \c_space_tl % deliberate space between radio buttons
2559 % to do: --> should be configurable
2560 }%
2561 }
2562
2563 \newcount\Fld@listcount
2564 \def\__hypListbox#1
2565 {
2566 \HyField@PDFChoices{#1}
2567 \mode_leave_vertical:
2568 \HyAnn@AbsPageLabel
2569 \Hy@escapeform\PDFForm@List
2570 \pdf_link_user:nnn
2571 {widget} %perhaps we need more types??
2572 {\PDFForm@List}
2573 {\MakeChoiceField{\Fld@width}{\Fld@height}}
2574 \HyField@AddToFields
2575 }
2576
2577
2578 \def\@PushButton[#1]#2{% parameters, label
2579 \def\Fld@name{#2}%
2580 \group_begin:
2581 \exp_args:No\HyField@SetKeys
2582 {
2583 \DefaultOptionsofPushButton,#1
2584 }
2585 \PDFForm@Name
2586 \prop_if_in:NnTF \g_pdfmeta_standard_prop {annot_action_no_A}
2587 {
2588 \msg_error:nn { hyp }{ pdfa-no-push-button }

```

```

2589     \LayoutPushButtonField
2590     {
2591         \mode_leave_vertical:
2592         \MakeButtonField{#2}
2593     }
2594 }
2595 {
2596     \HyField@FlagsPushButton
2597     \legacy_if:nT {Fld@hidden}
2598     {
2599         \def\Fld@width{1sp}
2600     }
2601     \LayoutPushButtonField
2602     {
2603         \mode_leave_vertical:
2604         \HyAnn@AbsPageLabel
2605         \Hy@escapeform\PDFForm@Push
2606         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2607         \pdfannot_box:nnnn
2608         {\box_wd:N\l_tmpa_box}
2609         {\box_ht:N\l_tmpa_box}
2610         {\box_dp:N\l_tmpa_box} %is this correct?
2611         {\PDFForm@Push}
2612         {\box_use:N\l_tmpa_box}
2613         \HyField@AddToFields
2614     }
2615 }
2616 \group_end:
2617 }
2618
2619 \def\@Submit[#1]#2
2620 {
2621     \def\Fld@width {\DefaultWidthofSubmit}
2622     \def\Fld@height{\DefaultHeightofSubmit}
2623     \group_begin:
2624     \exp_args:No\HyField@SetKeys
2625     {
2626         \DefaultOptionsofSubmit,#1
2627     }
2628     \HyField@FlagsPushButton
2629     \HyField@FlagsSubmit
2630     \legacy_if:nT { Fld@hidden }
2631     {
2632         \def\Fld@width{1sp}
2633     }
2634     \mode_leave_vertical:
2635     \HyAnn@AbsPageLabel
2636     \Hy@escapeform\PDFForm@Submit
2637     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2638     \pdfannot_box:nnnn
2639     {\box_wd:N\l_tmpa_box}
2640     {\box_ht:N\l_tmpa_box}
2641     {\box_dp:N\l_tmpa_box} %is this correct?
2642     {

```

```

2643         \PDFForm@Submit
2644     /AP<<
2645         /N~\pdf_xform_ref:n {__hyp_xform_Submit}~
2646         /D~\pdf_xform_ref:n {__hyp_xform_SubmitP}
2647     >>
2648     }
2649     \HyField@AddToFields
2650     \box_use:N\l_tmpa_box
2651
2652 \group_end:
2653 }
2654
2655 \def\@Reset[#1]#2
2656 {
2657     \def\Fld@width {\DefaultWidthofReset}
2658     \def\Fld@height{\DefaultHeightofReset}
2659     \group_begin:
2660     \exp_args:No\HyField@SetKeys
2661     {
2662         \DefaultOptionsofReset,#1
2663     }
2664     \mode_leave_vertical:
2665     \prop_if_in:NnTF \g_pdfmeta_standard_prop {annot_action_no_A}
2666     {
2667         \msg_error:nn { hyp }{ pdfa-no-reset-button }
2668         \MakeButtonField{#2}
2669     }
2670     {
2671         \HyField@FlagsPushButton
2672         \legacy_if:nT { Fld@hidden }
2673         { \def\Fld@width{1sp} }
2674         \HyAnn@AbsPageLabel
2675         \Hy@escapeform\PDFForm@Reset
2676         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2677         \pdfannot_box:nnnn
2678         {\box_wd:N\l_tmpa_box}
2679         {\box_ht:N\l_tmpa_box}
2680         {\box_dp:N\l_tmpa_box} %is this correct?
2681         { \PDFForm@Reset }
2682         \HyField@AddToFields
2683         \box_use:N \l_tmpa_box
2684     }
2685     \group_end:
2686 }
2687
2688 \def\@CheckBox[#1]#2
2689 {% parameters, label
2690     \def\Fld@name{#2}
2691     \def\Fld@default{0}
2692     \group_begin:
2693     \def\Fld@width {\DefaultWidthofCheckBox}
2694     \def\Fld@height{\DefaultHeightofCheckBox}
2695     \exp_args:No\HyField@SetKeys
2696     {

```

```

2697         \DefaultOptionsofCheckBox,#1
2698     }
2699     \PDFForm@Name
2700     \HyField@FlagsCheckBox
2701     \legacy_if:nT { Fld@hidden }
2702     {
2703         \def\Fld@width{1sp}
2704     }
2705     \LayoutCheckField{#2}
2706     {
2707         \mode_leave_vertical:
2708         \HyAnn@AbsPageLabel
2709         \Hy@escapeform\PDFForm@Check
2710         \pdfannot_box:nnnn
2711             {\Fld@width}
2712             {\Fld@height}
2713             {Opt} %is this correct?
2714             {\PDFForm@Check}
2715         \HyField@AddToFields %check if this works with xelatex ...
2716     }
2717     \group_end:
2718 }
2719 \ExplSyntaxOff
2720
2721 %hm. Should a luatex driver use type1 fonts in fields????
2722 \ExplSyntaxOn
2723 \def\Hy@FormObjects
2724 {
2725     \pdf_object_new:nn {l__hyp_encoding_pdfdoc_obj } { dict }
2726     \pdf_object_new:nn {l__hyp_font_zapfdingbats_obj } { dict }
2727     \pdf_object_new:nn {l__hyp_font_helvetica_obj } { dict }
2728     \pdf_object_write:nx {l__hyp_encoding_pdfdoc_obj }
2729     {
2730         /Type/Encoding
2731         /Differences[
2732             24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2733             /ring/tilde
2734             \c_space_tl
2735             39/quotesingle
2736             \c_space_tl
2737             96/grave %
2738             \iow_newline:
2739             128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2740             /fraction/guilsinglleft/guilsinglright/minus/perthousand
2741             /quotedblbase/quotedblleft/quotedblright/quoteleft
2742             /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2743             /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2744             \iow_newline:
2745             164/currency
2746             \c_space_tl
2747             166/brokenbar
2748             \c_space_tl
2749             168/dieresis/copyright/ordfeminine
2750             \c_space_tl

```



```

2751         172/logicalnot/.notdef/registered/macron/degree/plusminus
2752         /twosuperior/threesuperior/acute/mu
2753         \c_space_tl
2754         183/periodcentered/cedilla/onesuperior/ordmasculine
2755         \c_space_tl
2756         188/onequarter/onehalf/threequarters
2757         \iow_newline:
2758         192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
2759         /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
2760         /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
2761         /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
2762         /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
2763         /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
2764         /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
2765         /iacute/icircumflex/idieresis/eth/ntilde/ograve/oacute
2766         /ocircumflex/otilde/odieresis/divide/oslash/ugrave
2767         /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
2768     ]
2769 }
2770 \pdf_object_write:nn {l__hyp_font_zapfdingbats_obj }
2771 {
2772     /Type/Font
2773     /Subtype/Type1
2774     /Name/ZaDb
2775     /BaseFont/ZapfDingbats
2776 }
2777 \pdf_object_write:nx {l__hyp_font_helvetica_obj }
2778 {
2779     /Type/Font
2780     /Subtype/Type1
2781     /Name/Helv
2782     /BaseFont/Helvetica
2783     /Encoding~\pdf_object_ref:n { l__hyp_encoding_pdfdoc_obj }
2784 }
2785 \global\let\Hy@FormObjects\relax
2786 }
2787 \ExplSyntaxOff
2788 \providecommand*{\Fld@pageobjref}{\}
2789 \ifcsname pdf@escapestring\endcsname
2790 \def\Hy@escapeform#1{%
2791     \ifHy@pdfescapeform
2792     \let\Hy@escapestring\pdfescapestring
2793     \else
2794     \let\Hy@escapestring\@firstofone
2795     \fi
2796 }%
2797 \Hy@escapeform{}%
2798 \else
2799 \let\Hy@escapestring\@firstofone
2800 \def\Hy@escapeform#1{%
2801     \ifHy@pdfescapeform
2802     \def\Hy@escapestring##1{%
2803         \noexpand\Hy@escapestring{\noexpand##1}%
2804     }%

```

```

2805     \edef\Hy@temp{#1}%
2806     \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{}\@nil
2807     \def\Hy@escapestring##1{%
2808         \ifundefined{Hy@esc@\string##1}{%
2809             ##1%
2810             \ThisShouldNotHappen
2811         }{%
2812             \csname Hy@esc@\string##1\endcsname
2813         }%
2814     }%
2815     \else
2816         \let\Hy@escapestring\@firstofone
2817     \fi
2818 }%
2819 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
2820     \ifx\#3\%
2821     \else
2822         \expandafter
2823         \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
2824         \ltx@ReturnAfterFi{%
2825             \Hy__hypescapeform#3\@nil
2826         }%
2827     \fi
2828 }%
2829 \fi
2830 \def\PDFForm@Name{%
2831     \PDFForm__hypName\Fld@name
2832     \ifx\Fld@altname\relax
2833     \else
2834         \PDFForm__hypName\Fld@altname
2835     \fi
2836     \ifx\Fld@mappingname\relax
2837     \else
2838         \PDFForm__hypName\Fld@mappingname
2839     \fi
2840 }
2841 \def\PDFForm__hypName#1{%
2842     \begingroup
2843     \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
2844         \ifHy@unicode
2845             \Hy@unicodedefalse
2846         \fi
2847     \fi
2848     \pdfstringdef\Hy@gtemp#1%
2849 \endgroup
2850 \let#1\Hy@gtemp
2851 }
2852 \def\Fld@X@additionalactions{%
2853     \ifx\Fld@keystroke@code\@empty
2854     \else
2855         /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
2856     \fi
2857     \ifx\Fld@format@code\@empty
2858     \else

```

```

2859     /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
2860 \fi
2861 \ifx\Fld@validate@code\@empty
2862 \else
2863     /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
2864 \fi
2865 \ifx\Fld@calculate@code\@empty
2866 \else
2867     /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
2868 \fi
2869 \ifx\Fld@onfocus@code\@empty
2870 \else
2871     /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
2872 \fi
2873 \ifx\Fld@onblur@code\@empty
2874 \else
2875     /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
2876 \fi
2877 \ifx\Fld@onmousedown@code\@empty
2878 \else
2879     /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
2880 \fi
2881 \ifx\Fld@onmouseup@code\@empty
2882 \else
2883     /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
2884 \fi
2885 \ifx\Fld@onenter@code\@empty
2886 \else
2887     /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
2888 \fi
2889 \ifx\Fld@onexit@code\@empty
2890 \else
2891     /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
2892 \fi
2893 }
2894 \ExplSyntaxOn
2895 \def\Fld@additionalactions
2896 {%
2897     \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{\}
2898     {
2899         \prop_if_in:NnF \g_pdfmeta_standard_prop {annot_widget_no_AA}
2900         {\AA<<\Fld@X@additionalactions>>}
2901     }
2902 }
2903 \ExplSyntaxOff
2904 \def\Fld@annotnames{%
2905     /T(\Fld@name)%
2906     \ifx\Fld@altname\relax
2907     \else
2908         /TU(\Fld@altname)%
2909     \fi
2910     \ifx\Fld@mappingname\relax
2911     \else
2912         /TM(\Fld@mappingname)%

```

```

2913 \fi
2914 }
2915 \ExplSyntaxOn
2916 \def\PDFForm@Check
2917 {
2918   /Subtype/Widget
2919   ~\Fld@annotflags
2920   ~\Fld@pageobjref
2921   ~\Fld@annotnames
2922   /FT/Btn
2923   \Fld@flags
2924   /Q~\Fld@align
2925   /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
2926   /AP
2927   <<
2928     /N
2929     <<
2930       /Yes~\pdf_xform_ref:o{\l__hyp_CheckmarkYes_tl}
2931       /Off~\pdf_xform_ref:o{\l__hyp_CheckmarkOff_tl}
2932     >>
2933   >>
2934   /MK<<
2935     \int_compare:nNf {\Fld@rotation}={0}
2936     {
2937       /R~\Fld@rotation
2938     }
2939     \tl_if_empty:NF\Fld@bordercolor
2940     {
2941       /BC[\Fld@bordercolor]
2942     }
2943     \tl_if_empty:NF\Fld@bcolor
2944     {
2945       /BG[\Fld@bcolor]
2946     }
2947     /CA(\Hy@escapestring{\Fld@cbsymbol})%
2948   >>
2949   /DA
2950   (
2951     /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
2952     \tl_if_empty:NF \Fld@color
2953     {
2954       \c_space_tl \Fld@color
2955     }
2956   )
2957   /H/P
2958   \legacy_if:nTF {Fld@checked}
2959   {
2960     /V/Yes /AS/Yes
2961   }
2962   {
2963     /V/Off /AS/Off
2964   }
2965   \Fld@additionalactions
2966 }

```

```

2967 \ExplSyntaxOff
2968 \ExplSyntaxOn
2969 \prop_if_in:NnF \g_pdfmeta_standard_prop {annot_action_no_A}
2970 {
2971   \def\PDFForm@Push
2972   {
2973     /Subtype/Widget
2974     ~\Fld@annotflags
2975     ~\Fld@pageobjref
2976     ~\Fld@annotnames
2977     /FT/Btn
2978     ~\Fld@flags
2979     /H/P
2980     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
2981     \bool_if:nT
2982     {
2983       !\int_compare_p:nNn {\Fld@rotation} = {0}
2984       ||
2985       \tl_if_exist_p:N \Fld@bordercolor
2986     }
2987     {
2988       /MK
2989       <<
2990       \int_compare:nNnF {\Fld@rotation} = {0}
2991       {
2992         /R~\Fld@rotation
2993       }
2994       \tl_if_exist:NT \Fld@bordercolor
2995       {
2996         /BC[\Fld@bordercolor]
2997       }
2998       >>
2999     }
3000     /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3001     \Fld@additionalactions
3002   }
3003 }
3004 \ExplSyntaxOff
3005 \def\PDFForm@List{%
3006   /Subtype/Widget%
3007   \Fld@annotflags
3008   \Fld@pageobjref
3009   \Fld@annotnames
3010   /FT/Ch%
3011   \Fld@flags
3012   /Q \Fld@align
3013   /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3014   \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3015     \ifx\Fld@bordercolor\relax\else 1\fi
3016     \ifx\Fld@bcolor\relax \else 1\fi
3017     \space
3018   \else
3019     /MK<<%
3020     \ifnum\Fld@rotation=\z@

```

```

3021     \else
3022         /R \Fld@rotation
3023     \fi
3024     \ifx\Fld@bordercolor\relax
3025     \else
3026         /BC[\Fld@bordercolor]%
3027     \fi
3028     \ifx\Fld@bcolor\relax
3029     \else
3030         /BG[\Fld@bcolor]%
3031     \fi
3032     >>%
3033 \fi
3034 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3035     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3036 \Fld@choices
3037 \Fld@additionalactions
3038 }
3039 \ExplSyntaxOn
3040 \def\PDFForm@Radio
3041 {
3042     /Subtype/Widget
3043     ~\Fld@annotflags
3044     ~\Fld@pageobjref
3045     ~\Fld@annotnames
3046     /FT/Btn
3047     \Fld@flags
3048     /H/P
3049     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3050     /MK<<
3051     \ifnum\Fld@rotation=\z@
3052     \else
3053         /R~\Fld@rotation
3054     \fi
3055     \ifx\Fld@bordercolor\relax
3056     \else
3057         /BC[\Fld@bordercolor]%
3058     \fi
3059     \ifx\Fld@bcolor\relax
3060     \else
3061         /BG[\Fld@bcolor]%
3062     \fi
3063     /CA(\Hy@escapestring{\Fld@radiosymbol})%
3064     >>
3065     /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3066         \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3067     \ifx\Fld@default\@empty
3068         /V/Off%
3069         /DV/Off%
3070     \else
3071         /V/\Fld@default
3072         /DV/\Fld@default
3073     \fi
3074     \Fld@additionalactions

```

```

3075 }
3076 \ExplSyntaxOff
3077 \ExplSyntaxOn
3078 % Does an appeareance dict make sense here?
3079 \def\PDFForm@Text
3080 {
3081   /Subtype/Widget
3082   ~\Fld@annotflags
3083   ~\Fld@pageobjref
3084   ~\Fld@annotnames
3085   /FT/Tx
3086   ~\Fld@flags
3087   /Q~\Fld@align
3088   /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3089   \bool_if:nT
3090   {
3091     !\int_compare_p:nNn {\Fld@rotation} = {0}
3092     ||
3093     \tl_if_exist_p:N \Fld@bordercolor
3094     ||
3095     \tl_if_exist_p:N \Fld@bcolor
3096   }
3097   {
3098     /MK
3099     <<
3100     \int_compare:nNnF {\Fld@rotation} = {0}
3101     {
3102       /R~\Fld@rotation
3103     }
3104     \tl_if_exist:NT \Fld@bordercolor
3105     {
3106       /BC[\Fld@bordercolor]
3107     }
3108     \tl_if_exist:NT \Fld@bcolor
3109     {
3110       /BG[\Fld@bcolor]
3111     }
3112     >>
3113   }
3114   /DA
3115   (
3116     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3117     \tl_if_empty:NF {\c_space_tl\Fld@color}
3118   )
3119   /DV(\Hy@escapestring{\Fld@default})
3120   /V(\Hy@escapestring{\Fld@value})
3121   ~\Fld@additionalactions
3122   \int_compare:nNnT { \Fld@maxlen}>{0}
3123   {
3124     /MaxLen~\Fld@maxlen
3125   }
3126 }
3127 \ExplSyntaxOff
3128

```

```

3129 \def\PDFForm@Submit{%
3130 /Subtype/Widget%
3131 \Fld@annotflags
3132 \Fld@pageobjref
3133 \Fld@annotnames
3134 /FT/Btn%
3135 \Fld@flags
3136 /H/P%
3137 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3138 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3139 \ifx\Fld@bordercolor\relax\else 1\fi
3140 \space
3141 \else
3142 /MK<<%
3143 \ifnum\Fld@rotation=\z@
3144 \else
3145 /R \Fld@rotation
3146 \fi
3147 \ifx\Fld@bordercolor\relax
3148 \else
3149 /BC[\Fld@bordercolor]%
3150 \fi
3151 >>%
3152 \fi
3153 /A<<%
3154 /S/SubmitForm%
3155 /F<<%
3156 /FS/URL%
3157 /F(\Hy@escapestring{\Form@action})%
3158 >>%
3159 \Fld@submitflags
3160 >>%
3161 \Fld@additionalactions
3162 }
3163 \ExplSyntaxOn
3164 \prop_if_in:NnF \g_pdfmeta_standard_prop {annot_action_no_A}
3165 {
3166 \def\PDFForm@Reset{%
3167 /Subtype/Widget%
3168 \Fld@annotflags
3169 \Fld@pageobjref
3170 \Fld@annotnames
3171 /FT/Btn%
3172 \Fld@flags
3173 /H/P%
3174 /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3175 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3176 \ifx\Fld@bordercolor\relax\else 1\fi
3177 \space
3178 \else
3179 /MK<<%
3180 \ifnum\Fld@rotation=\z@
3181 \else
3182 /R~\Fld@rotation

```



```

3183         \fi
3184         \ifx\Fld@bordercolor\relax
3185         \else
3186             /BC[\Fld@bordercolor]%
3187         \fi
3188     >>%
3189     \fi
3190     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3191     /A<</S/ResetForm>>%
3192     \Fld@additionalactions
3193 }%
3194 }
3195
3196 %these patterns are used in hyperref checks.
3197 %it is unclear if they are really useful and if a backend support is
3198 %needed.
3199 \str_case:VnF \c_sys_backend_str
3200 {
3201     { pdfmode }
3202     {
3203         \def\HyPat@ObjRef
3204         {
3205             [0-9]*[1-9][0-9]*~0~R
3206         }
3207     }
3208     { dvipdfmx }
3209     {
3210         \def\HyPat@ObjRef
3211         {
3212             @[^~]+
3213         }
3214     }
3215     { xdvipdfmx }
3216     {
3217         \def\HyPat@ObjRef
3218         {
3219             @[^~]+
3220         }
3221     }
3222 }
3223 { %also set in hyperref sty, so probably not needed.
3224     \def\HyPat@ObjRef/{.+}
3225 }
3226
3227
3228 \ExplSyntaxOff
3229 % UF: removed Hy@writebookmark
3230 % \Hy@currentbookmarklevel{0}
3231 % \Hy@numberline
3232 % \__hypwritetorep
3233 % counter{bookmark@seq@number}
3234 % removed \HyPsd@SanitizeForOutFile, not needed
3235 % removed \currentpdfbookmark, defined by bookmark,
3236 % should use \newcommand there

```

```

3237 % removed \subpdfbookmark, defined by bookmark,
3238 % should use \newcommand there
3239 % removed \belowpdfbookmark, defined by bookmark,
3240 % should use \newcommand there
3241 % removed \pdfbookmark, defined by bookmark,
3242 % \BOOKMARK
3243 % \@BOOKMARK
3244 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3245 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3246 %% removed \ReadBookmarks / unneeded with bookmark.
3247 %% removed \Hy@OutlineName
3248 %% removed \check@bm@number
3249 %% removed \calc@bm@number
3250
3251 \ifHy@implicit
3252 \else
3253   \expandafter\endinput
3254 \fi
3255 \newlength\Hy@SectionHShift
3256 \def\Hy@SectionAnchorHref#1{%
3257   \ifx\protect\@typeset@protect
3258     \Hy__hypSectionAnchor{#1}%
3259   \fi
3260 }
3261 \DeclareRobustCommand*\Hy__hypSectionAnchor[1]{%
3262   \leavevmode
3263   \hbox to 0pt{%
3264     \kern-\Hy@SectionHShift
3265     \Hy@raisedlink{%
3266       \hyper@anchorstart{#1}\hyper@anchorend
3267     }%
3268     \hss
3269   }%
3270 }
3271 \let\H@old@ssect\@ssect
3272 \def\@ssect#1#2#3#4#5{%
3273   \Hy@MakeCurrentHrefAuto{section*}%
3274   \setlength{\Hy@SectionHShift}{#1}%
3275   \begingroup
3276     \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3277     \toks\tw@\expandafter{%
3278       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3279       #5%
3280     }%
3281   \edef\x{\endgroup
3282     \the\toks@\the\toks\tw@}%
3283   }\x
3284 }
3285 \let\H@old@schapter\@schapter
3286 \def\@schapter#1{%
3287   \begingroup
3288     \let\@mkboth\@gobbletwo
3289     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3290     \Hy@raisedlink{%

```

```

3291     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3292   }%
3293 \endgroup
3294 \H@old@schapter{#1}%
3295 }
3296 \ltx@ifundefined{@chapter}{}{%
3297   \let\Hy@org@chapter\@chapter
3298   \def\@chapter{%
3299     \def\Hy@next{%
3300       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3301       \Hy@raisedlink{%
3302         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3303       }%
3304     }%
3305     \ifnum\c@secnumdepth>\m@ne
3306       \ltx@ifundefined{if@mainmatter}%
3307       \iftrue{\csname if@mainmatter\endcsname}%
3308       \let\Hy@next\relax
3309     \fi
3310   \fi
3311   \Hy@next
3312   \Hy@org@chapter
3313 }%
3314 }
3315 \let\H@old@part\@part
3316 \begingroup\expandafter\expandafter\expandafter\endgroup
3317 \expandafter\ifx\csname chapter\endcsname\relax
3318   \let\Hy@secnum@part\z@
3319 \else
3320   \let\Hy@secnum@part\m@ne
3321 \fi
3322 \def\@part{%
3323   \ifnum\Hy@secnum@part>\c@secnumdepth
3324     \phantomsection
3325   \fi
3326   \H@old@part
3327 }
3328 \let\H@old@spart\@spart
3329 \def\@spart#1{%
3330   \Hy@MakeCurrentHrefAuto{part*}%
3331   \Hy@raisedlink{%
3332     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3333   }%
3334   \H@old@spart{#1}%
3335 }
3336 \let\H@old@sect\@sect
3337 \def\@sect#1#2#3#4#5#6[#7]#8{%
3338   \ifnum #2>\c@secnumdepth
3339     \expandafter\@firstoftwo
3340   \else
3341     \expandafter\@secondoftwo
3342   \fi
3343   {%
3344     \Hy@MakeCurrentHrefAuto{section*}%

```

```

3345 \setlength{\Hy@SectionHShift}{#3}%
3346 \begingroup
3347 \toks@{\H@old@ssect{#1}{#2}{#3}{#4}{#5}{#6}[\{#7\}]}%
3348 \toks\tw@\expandafter{%
3349 \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3350 #8%
3351 }%
3352 \edef\x{\endgroup
3353 \the\toks@{\the\toks\tw@}%
3354 }\x
3355 }{%
3356 \H@old@ssect{#1}{#2}{#3}{#4}{#5}{#6}[\{#7\}]{#8}%
3357 }%
3358 }
3359 \expandafter\def\csname Parent-4\endcsname{}
3360 \expandafter\def\csname Parent-3\endcsname{}
3361 \expandafter\def\csname Parent-2\endcsname{}
3362 \expandafter\def\csname Parent-1\endcsname{}
3363 \expandafter\def\csname Parent0\endcsname{}
3364 \expandafter\def\csname Parent1\endcsname{}
3365 \expandafter\def\csname Parent2\endcsname{}
3366 \expandafter\def\csname Parent3\endcsname{}
3367 \expandafter\def\csname Parent4\endcsname{}
3368 %%
3369 %% End of file 'hgeneric-experimental.def'.
3370 \endpackage
3371 \begin{colourscheme}
3372 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3373 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3374 % linkcolor=
3375 %,filecolor=
3376 %,urlcolor=
3377 %,menucolor=
3378 %,runcolor=
3379 %,linkbordercolor=
3380 %,filebordercolor=
3381 %,urlbordercolor=
3382 %,menubordercolor=
3383 %,runbordercolor=
3384
3385 \prop_const_from_keyval:Nn \c__hyp_colourscheme_hypori_prop
3386 {
3387 linkcolor = [rgb]{1,0,0}, %red
3388 filecolor = [rgb]{0,1,1}, %cyan
3389 urlcolor = [rgb]{1,0,1}, %magenta
3390 menucolor = [rgb]{1, 0, 0}, %red
3391 runcolor = [rgb]{0,1,1}, %cyan
3392 %-----
3393 linkbordercolor = [rgb]{1, 0 ,0 },
3394 filebordercolor = [rgb]{0, .5, .5},
3395 urlbordercolor = [rgb]{0, 1, 1},
3396 menubordercolor = [rgb]{1, 0, 0},
3397 runbordercolor = [rgb]{0, .7, .7}
3398 }

```

```

3399
3400 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3401 {
3402   linkcolor      = [rgb]{0,0.2,0.6},
3403   filecolor      = [rgb]{0.8,0,0.8},
3404   urlcolor       = [rgb]{0.8,0,0.8},
3405   menucolor      = [rgb]{0,0.2,0.6},
3406   runcolor       = [rgb]{0.8,0,0.8},
3407   %----- %-----
3408   linkbordercolor = [rgb]{0,0.2,0.6},
3409   filebordercolor = [rgb]{0.8,0,0.8},
3410   urlbordercolor  = [rgb]{0.8,0,0.8},
3411   menubordercolor = [rgb]{0,0.2,0.6},
3412   runbordercolor  = [rgb]{0.8,0,0.8}
3413 }
3414
3415 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3416 { %two colors: intern/extern
3417   linkcolor      = [rgb]{0.79216, 0, 0.12549},
3418   filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3419   urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3420   menucolor      = [rgb]{0.79216, 0, 0.12549 },
3421   runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3422   %----- %-----
3423   linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3424   filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3425   urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3426   menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3427   runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3428 }
3429
3430 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3431 { %all darkgray
3432   linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3433   filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3434   urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3435   menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3436   runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3437   %----- %-----
3438   linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3439   filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3440   urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3441   menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3442   runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3443 }
3444
3445 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3446 { %dvipsnam.def
3447   linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3448   filecolor      = [rgb]{1, 0, 0}, %Red
3449   urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3450   menucolor      = [rgb]{1, 0, 0}, %Red
3451   runcolor       = [rgb]{1, 0, 0}, %Red
3452   %----- %-----

```

```

3453 linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3454 filebordercolor = [rgb]{1, 0, 0}, %Red
3455 urlbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3456 menubordercolor = [rgb]{1, 0, 0}, %Red
3457 runbordercolor = [rgb]{1, 0, 0} %Red
3458 }
3459
3460 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3461 { %dvipsnam.def
3462 linkcolor = [rgb]{0.72, 0, 0}, %BrickRed
3463 filecolor = [rgb]{0, 1, 0}, %Green
3464 urlcolor = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3465 menucolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3466 runcolor = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3467 %----- %-----
3468 linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3469 filebordercolor = [rgb]{0, 1, 0}, %Green
3470 urlbordercolor = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3471 menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3472 runbordercolor = [rgb]{0.64, 0.08, 0.98} %Mulberry
3473 }
3474
3475
3476 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3477 {
3478 linkcolor = [rgb]{0.50196, 0, 0.02353},
3479 filecolor = [rgb]{0.07451, 0.09412, 0.46667},
3480 urlcolor = [rgb]{0.54118, 0, 0.52941},
3481 menucolor = [rgb]{0.44706, 0.45882, 0},
3482 runcolor = [rgb]{0.07451, 0.46667, 0.46275},
3483 %----- %-----
3484 linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3485 filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3486 urlbordercolor = [rgb]{0.724708, 0.4, 0.717646},
3487 menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3488 runbordercolor = [rgb]{0.444706, 0.680002, 0.67765}
3489 }
3490
3491 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3492 {
3493 linkcolor = [rgb]{0,0,0},
3494 filecolor = [rgb]{0,0,0},
3495 urlcolor = [rgb]{0,0,0},
3496 menucolor = [rgb]{0,0,0},
3497 runcolor = [rgb]{0,0,0},
3498 %----- %-----
3499 linkbordercolor = [rgb]{0,0,0},
3500 filebordercolor = [rgb]{0,0,0},
3501 urlbordercolor = [rgb]{0,0,0},
3502 menubordercolor = [rgb]{0,0,0},
3503 runbordercolor = [rgb]{0,0,0}
3504 }
3505 </colorscheme>

```

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols		C	
\#	491	\char	2335, 2347, 2359, 2371
\%	492	\chardef	137
\.	306, 310, 312	\cite	25
@curropt commands:		clist commands:	
\@curropt:	2463, 2531	\clist_map_function:nN	112, 121
\\	19, 20, 21,	\clist_map_inline:nn	2034
	28, 32, 42, 62, 69, 76, 83, 90, 105,	\clist_map_variable:nNn	2462, 2530
	110, 111, 119, 120, 129, 130, 131,	color commands:	
	132, 133, 139, 617, 2465, 2533, 2820	\color_export:nnN	670
_	304, 306, 310, 312	\color_export:nnnN	676
A		\color_select:n	687, 723, 991
\A	304	\color_select:nn	693
\Acrobatmenu	15, 156	\color_set:nn	185, 704
\addcontentsline	10	\color_set:nnn	9, 184, 710
\AddToDocumentMetadata	180	color names:	
\AddToHook	205	hyp/annot/file	287
\advance	2396, 2470, 2514, 2515	hyp/annot/link	287
allcolor (hypersetup key)	750	hyp/annot/menu	287
\author	2	hyp/annot/run	287
B		hyp/annot/url	287
\b	308	colorfile (hypersetup key)	750
\begingroup	2238, 2430, 2459, 2471,	colorlink (hypersetup key)	750
	2505, 2842, 3275, 3287, 3316, 3346	colorlinks (hypersetup key)	736
\belowpdfbookmark	3239	colormenu (hypersetup key)	750
\BOOKMARK	3242	colorrun (hypersetup key)	750
bookmarkstype (hypersetup key)	10	colorscheme (hypersetup key)	1154
bool commands:		colorurl (hypersetup key)	750
\bool_if:NTF	437, 451, 462,	cs commands:	
	475, 518, 568, 628, 720, 730, 970, 981	\cs_generate_variant:Nn	136,
\bool_if:nTF	580, 1673, 1698, 2981, 3089		182, 218, 233, 238, 342, 679, 696, 713
\bool_lazy_and:nnTF	200	\cs_gset:Npn	962
\bool_new:N	290, 294, 298	\cs_gset_eq:NN	9
\bool_set_true:N	299	\cs_if_exist_p:N	201
bordercolormodel (hypersetup key)	10, 771	\cs_new:Npn	105, 234
box commands:		\cs_new_protected:Npn	177, 212,
\box_dp:N	380, 2610, 2641, 2680		221, 318, 322, 326, 330, 343, 345,
\box_ht:N	379, 2609, 2640, 2679		349, 391, 397, 404, 408, 430, 435,
\box_new:N	239, 301		449, 460, 473, 516, 566, 626, 663,
\box_set_dp:Nn	1870		674, 680, 691, 697, 708, 906, 1905, 1961
\box_set_ht:Nn	1877	\cs_set_eq:NN	
\box_use:N	987, 2612, 2650, 2683		140, 149, 150, 491, 492, 1231
\box_use_drop:N	992	\cs_set_protected:Npn	
\box_wd:N	378, 2608, 2639, 2678		138, 163, 359, 617
\l_tmpa_box	2606, 2608, 2609,	\csname	2477, 2812,
	2610, 2612, 2637, 2639, 2640, 2641,		2823, 3307, 3317, 3359, 3360, 3361,
	2650, 2676, 2678, 2679, 2680, 2683		3362, 3363, 3364, 3365, 3366, 3367
		\currentpdfbookmark	3235

D	
\d	306, 310, 312
debug (hypersetup key)	<u>1178</u>
\DeclareDocumentMetadata	2, 8, 11, 12, 15, 21, 29, 35
\DeclareRobustCommand	3261
\def	144, 354, 2200, 2204, 2211, 2229, 2260, 2277, 2394, 2403, 2418, 2419, 2422, 2423, 2436, 2437, 2453, 2454, 2457, 2458, 2496, 2526, 2564, 2578, 2579, 2599, 2619, 2621, 2622, 2632, 2655, 2657, 2658, 2673, 2688, 2690, 2691, 2693, 2694, 2703, 2723, 2790, 2800, 2802, 2807, 2819, 2830, 2841, 2852, 2895, 2904, 2916, 2971, 3005, 3040, 3079, 3129, 3166, 3203, 3210, 3217, 3224, 3256, 3272, 3286, 3298, 3299, 3322, 3329, 3337, 3359, 3360, 3361, 3362, 3363, 3364, 3365, 3366, 3367
\DefaultHeightofCheckBox	2694
\DefaultHeightofChoiceMenu	2458
\DefaultHeightofReset	2658
\DefaultHeightofSubmit	2622
\DefaultHeightofText	2427
\DefaultHeightofTextMultiline	2425
\DefaultOptionsofCheckBox	2697
\DefaultOptionsofPushButton	2583
\DefaultOptionsofReset	2662
\DefaultOptionsofSubmit	2626
\DefaultOptionsofText	2432
\DefaultWidthofCheckBox	2693
\DefaultWidthofChoiceMenu	2457
\DefaultWidthofReset	2657
\DefaultWidthofSubmit	2621
\DefaultWidthofText	2422
destlabel (hypersetup key)	10
dim commands:	
\dim_to_decimal_in_bp:n	9, 140
\do	2463, 2531
\documentclass	22
draft (hypersetup key)	<u>1178</u>
E	
\edef	2240, 2473, 2509, 2516, 2805, 3281, 3352
\EdefEscapeName	2529, 2534
\else	2237, 2426, 2480, 2484, 2487, 2504, 2512, 2793, 2798, 2815, 2821, 2833, 2837, 2854, 2858, 2862, 2866, 2870, 2874, 2878, 2882, 2886, 2890, 2907, 2911, 3014, 3015, 3016, 3018, 3021, 3025, 3029, 3035, 3052, 3056, 3060, 3066, 3070, 3138, 3139, 3141, 3144, 3148, 3175, 3176, 3178, 3181, 3185, 3252, 3319, 3340
\endcsname	2491, 2789, 2812, 2823, 3307, 3317, 3359, 3360, 3361, 3362, 3363, 3364, 3365, 3366, 3367
\endgroup	2241, 2450, 2473, 2519, 2522, 2849, 3281, 3293, 3316, 3352
\endinput	3253
exp commands:	
\exp_args:Ne	2897
\exp_args:Nne	1999
\exp_args:NNV	1658, 1689
\exp_args:NnV	393, 400
\exp_args:Nnx	1947, 2182
\exp_args:No	422, 426, 2581, 2624, 2660, 2695
\exp_args:NV	376, 383
\exp_args:Nx	1940, 1988, 2231
\exp_not:N	617
\exp_not:n	1093, 1127, 1238, 1298, 1319, 1356, 1376, 1399, 1443, 1476, 1501, 1552, 1577, 1604, 1653, 1730, 1758, 1785, 1805, 2020, 2127, 2155, 2165, 2176
\expandafter	2431, 2465, 2474, 2476, 2477, 2533, 2806, 2822, 3253, 3277, 3278, 3316, 3317, 3339, 3341, 3348, 3349, 3359, 3360, 3361, 3362, 3363, 3364, 3365, 3366, 3367
\ExplSyntaxOff	11, 2270, 2386, 2413, 2719, 2787, 2903, 2967, 3004, 3076, 3127, 3228
\ExplSyntaxOn	8, 12, 2271, 2391, 2414, 2722, 2894, 2915, 2968, 3039, 3077, 3163
extension (hypersetup key)	<u>10</u> , <u>1200</u>
F	
\fbox	2377, 2382, 2555
\fi	485, 2225, 2254, 2257, 2266, 2428, 2436, 2437, 2468, 2486, 2489, 2490, 2496, 2499, 2508, 2510, 2517, 2520, 2795, 2817, 2827, 2829, 2835, 2839, 2846, 2847, 2856, 2860, 2864, 2868, 2872, 2876, 2880, 2884, 2888, 2892, 2909, 2913, 3014, 3015, 3016, 3023, 3027, 3031, 3033, 3035, 3054, 3058, 3062, 3066, 3073, 3138, 3139, 3146, 3150, 3152, 3175, 3176, 3183, 3187, 3189, 3254, 3259, 3309, 3310, 3321, 3325, 3342
file (hypersetup key)	7, <u>1246</u>
file commands:	
\file_input:n	10

fileborderstyle (hypersetup key) .	11, 860	\l_hyp_annot_ocgcolorfile_bool .	292
filecolor (hypersetup key)	750	\l_hyp_annot_ocgcolorlink_bool .	292
final (hypersetup key)	1178	\l_hyp_annot_ocgcolormenu_bool .	292
\fontencoding	2331, 2343, 2355, 2367	\l_hyp_annot_ocgcolorrun_bool .	292
\fontfamily	2330, 2342, 2354, 2366	\l_hyp_annot_ocgcolorurl_bool .	292
\fontseries	2332, 2344, 2356, 2368	hyp internal commands:	
\fontshape	2333, 2345, 2357, 2369	\g__hyp_AcroForm_CoFields_prop . .	
fp commands:		2192, 2213, 2289, 2291, 2306
\fp_eval:n	1828, 1856, 1872	\g__hyp_AcroForm_Fields_prop . . .	
G			
\GetDocumentMetadata	2	2193, 2206, 2284
\global	2396, 2785	\l_hyp_annot_GoTo_bool .	437, 451, 462
group commands:		\l_hyp_annot_GoTo_bool\l_hyp_annot_URI_bool\l_hyp_annot_GoToR_bool\l_hyp_annot_Named_bool\l_hyp_annot_Launch_bool	296
\group_begin:		\l_hyp_annot_GoToR_bool	518
.	332, 440, 454, 477, 520, 570, 630, 722, 973, 990, 2106, 2329, 2341, 2353, 2365, 2580, 2623, 2659, 2692	\l_hyp_annot_Launch_bool	568
\group_end:		\l_hyp_annot_Named_bool	628
.	339, 444, 465, 507, 558, 611, 658, 732, 993, 996, 2113, 2336, 2348, 2360, 2372, 2616, 2652, 2685, 2717	\c__hyp_annot_types_seq	
H			
\hbox	3263	251, 288, 292, 750, 1045, 1141
hbox commands:		\l_hyp_annot_URI_bool	475
\hbox_overlap_right:n	987	\g__hyp_bordercolormodel_str . . .	
\hbox_set:Nn	2606, 2637, 2676	287, 775, 791
\hbox_set:Nw	974	\l_hyp_CheckmarkOff_tl	
\hbox_set_end:	983	2274, 2275, 2931
\hbox_set_to_wd:Nnn	1854	\l_hyp_CheckmarkYes_tl	
hidefile (hypersetup key)	1130	2272, 2273, 2930
hidelink (hypersetup key)	1130	__hyp_clist_display:n .	105, 112, 121
hidelinks (hypersetup key)	1130	__hyp_color_export:nnN	
hidemenu (hypersetup key)	1130	31, 663, 663, 679, 791
hiderun (hypersetup key)	1130	__hyp_color_select:n .	31, 680, 680, 696
hideurl (hypersetup key)	1130	__hyp_color_select:nN	31
hook commands:		__hyp_color_set:nN	32
\hook_gput_code:nnn		__hyp_color_set:nn	
.	192, 716, 726, 966, 977	32, 188, 190, 697, 697, 713, 755
\hook_new:n	183, 317	__hyp_colormodel_export:nnnN .	31, 663
\hook_use:n	334, 441, 455	__hyp_colormodel_export:wnnN . . .	
\hss	3268	667, 674
Hy internal commands:		__hyp_colormodel_select_aux:wn .	
\Hy_hypescapeform . .	2806, 2819, 2825	680, 684, 691
\Hy_hypSectionAnchor . . .	3258, 3261	__hyp_colormodel_set_aux:nnw . . .	
HyField internal commands:		697, 701, 708
\HyField_hypAddToFields .	2231, 2260	\c__hyp_colorscheme_daleif_prop .	3400
hyp commands:		\c__hyp_colorscheme_henryford_prop	3491
\l_hyp_annot_colorfile_bool . . .	288	\c__hyp_colorscheme_hypori_prop .	3385
\l_hyp_annot_colorlink_bool . . .	288	\c__hyp_colorscheme_julian_prop .	3415
\l_hyp_annot_colormenu_bool . . .	288	\c__hyp_colorscheme_phetype_prop	3476
\l_hyp_annot_colorrun_bool	288	3445
\l_hyp_annot_colorurl_bool	288	\c__hyp_colorscheme_szabolcsA_prop	3460
		\c__hyp_colorscheme_szabolcsB_prop	

\c__hyp_colorscheme_tivv_prop .	3430	\l__hyp_text_enc_dest_tl	24, 273, 372, 423
\l__hyp_dest_box	24, 301, 378, 379, 380, 1854, 1870, 1877	\l__hyp_text_enc_file_print_tl ..	273, 525, 573
\l__hyp_dest_name_tmpa_tl ..	244, 411, 412, 418, 422, 424, 427, 531, 539	\l__hyp_text_enc_info_print_tl ..	273, 347
\l__hyp_dest_pdfremotestartview_-tl	270, 536, 1661, 1665	\l__hyp_text_enc_para_print_tl ..	273, 578
\g__hyp_dest_pdfstartpage_tl ...	270, 1672, 1674, 1681, 1699, 1706	\l__hyp_text_enc_uri_print_tl ...	273, 480, 1257
\g__hyp_dest_pdfstartview_tl ...	270, 1674, 1681, 1692, 1696, 1699, 1706	__hyp_text_pdfstring:nnN	330, 330, 342, 344, 347, 370, 421, 478, 523, 528, 571, 576, 1257
\l__hyp_dest_pdfview_tl	286, 394, 401, 1823, 1826, 1833, 1836, 1837, 1838, 1839, 1840, 1841, 1846, 1850, 1886	__hyp_text_pdfstring_info:nN ...	345, 345, 1916, 1934, 1944
\c__hyp_dest_startview_regex ...	302, 1659, 1690	__hyp_text_purify:nN .	318, 318, 335
\c__hyp_dest_undefined_tl	250, 417, 418	\g__hyp_text_tmpa_str .	249, 338, 340
__hyp_destination:nn	24, 365, 365, 394, 401	\l__hyp_text_tmpa_str	248, 335, 336, 337, 338
\l__hyp_filename_tmpa_tl ...	244, 521, 522, 524, 526, 527, 574, 575, 591	\l__hyp_tmpa_box ...	239, 974, 987, 992
__hyp_link_goto_begin:nw	408, 442, 456	\l__hyp_tmpa_int	239
__hyp_link_goto_end: .	430, 443, 464	\l__hyp_tmpa_seq	239, 1659, 1661, 1690, 1692, 1813, 1814, 1818, 1820, 1844, 1851, 1852, 1853, 1862, 1864, 1874, 1879, 2293, 2295, 2301
\c__hyp_map_annot_hyp_prop ...	251	\l__hyp_tmpa_str	239, 1916, 1917, 1919, 1934, 1935, 1937, 1944, 1945, 1949
\c__hyp_map_hyp_annot_prop	251, 714, 778, 814, 841, 849, 860, 887, 895, 964, 1068, 1104, 1114, 1246	\l__hyp_tmpa_tl	239, 373, 377, 385, 791, 795, 1257, 1258, 1263, 1657, 1659, 1688, 1690, 1820, 1821, 1828
__hyp_ocg_init: ...	906, 906, 962, 972	\l__hyp_uri_tmpa_tl ...	244, 481, 482
__hyp_PageLabels_gpush: .	349, 349, 362	hyp/annot/file (color name)	287
\l__hyp_para_tmpa_tl	247, 579, 582, 591	hyp/annot/link (color name)	287
\l__hyp_para_tmpa_tl__hyp_-text_tmpa_str\g__hyp_text_-tmpa_str	244	hyp/annot/menu (color name)	287
\l__hyp_RadioYes_tl	2524, 2525	hyp/annot/run (color name)	287
__hyp_ref_check:nn	211, 221, 233, 2401	hyp/annot/url (color name)	287
__hyp_ref_if_exist:nn .	211, 219, 220	hyp/text/pdfstring	317
__hyp_ref_if_exist:nnTF ..	223, 2405	\hypercalcbp	9, 15, 140
__hyp_ref_label:nn	211, 212, 218, 2400	\HyperDestNameFilter	371, 422
__hyp_ref_value:nn	211, 234, 238, 2409	\hypersetup ..	1, 2, 6, 7, 10, 15, 59, 90, 163
__hyp_setup_info_date_key:nn ...	1961, 1993, 1994	\hypersetup keys:	
__hyp_setup_info_key:nn ..	1905, 1954, 1955, 1956, 1957, 1959, 1960	allcolor	750
__hyp_store_metadata:nn	177, 182, 1902, 1922, 1940, 1974, 1988, 2009, 2081	bookmarkstype	10
__hyp_text_cleanup:N .	322, 322, 336	bordercolormodel	10, 771
__hyp_text_convert:nN .	326, 326, 337	colorfile	750
\l__hyp_text_enc_dest_print_tl ..	273, 530	colorlink	750
		colorlinks	736
		colormenu	750
		colorrun	750
		colorscheme	1154
		colorurl	750
		debug	1178
		destlabel	10

L		1374, 1397, 1459, 1488, 1522, 1537, 1550, 1563, 1588, 1621, 1637, 1664, 1695, 1728, 1742, 1769, 1803, 1885
<code>\label</code>	6, 10	
<code>\LayoutCheckField</code>	2705	
<code>\LayoutChoiceField</code>	2500	
<code>\LayoutPushButtonField</code>	2589, 2601	<code>\msg_warning:nnnnn</code> 173,
<code>\LayoutTextField</code>	2438	1090, 1124, 1235, 1316, 1353, 1440,
<code>\leavevmode</code>	2439, 2537, 3262	1473, 1498, 1574, 1601, 1650, 1755,
legacy commands:		1782, 2017, 2124, 2143, 2162, 2173
<code>\legacy_if:nTF</code>	151, 203, 207, 357, 2321, 2597, 2630, 2672, 2701, 2958	
<code>\let</code>	147, 2195, 2196, 2198, 2199, 2226, 2279, 2387, 2388, 2389, 2420, 2421, 2455, 2456, 2498, 2785, 2792, 2794, 2799, 2816, 2850, 3271, 3285, 3288, 3297, 3308, 3315, 3318, 3320, 3328, 3336	
<code>link</code> (hypersetup key)	7, 1246	
<code>linkborder</code> (hypersetup key)	11	
<code>linkborderstyle</code> (hypersetup key) .	11, 860	
<code>linkcolor</code> (hypersetup key)	750	
<code>linkfileprefix</code> (hypersetup key) ...	1200	
<code>linktoc</code> (hypersetup key)	1200	
<code>linktocpage</code> (hypersetup key)	1200	
<code>localanchorname</code> (hypersetup key) ..	1200	
M		
<code>\MakeButtonField</code>	2592, 2606, 2637, 2668, 2676	
<code>\MakeChoiceField</code>	2573	
<code>\MakeFieldObject</code>	2186, 2327, 2339, 2351, 2363, 2375, 2380	
<code>\MakeRadioField</code>	2555	
<code>\MakeTextField</code>	2447	
<code>\mbox</code>	984	
<code>menu</code> (hypersetup key)	7, 1246	
<code>menuborder</code> (hypersetup key)	11	
<code>menuborderstyle</code> (hypersetup key) .	11, 860	
<code>menucolor</code> (hypersetup key)	750	
mode commands:		
<code>\mode_if_horizontal:TF</code>	367, 389	
<code>\mode_leave_vertical:</code>	410, 494, 546, 598, 633, 2567, 2591, 2603, 2634, 2664, 2707	
msg commands:		
<code>\msg_error:nn</code>	202, 2588, 2667	
<code>\msg_line_context:</code>	70	
<code>\g_msg_module_name_prop</code>	13	
<code>\msg_new:nnn</code>	38, 45, 50, 54, 58, 65, 72, 79, 86, 93, 99, 106, 115, 125	
<code>\msg_new:nnnn</code>	14, 24	
<code>\msg_warning:nn</code>	153, 1174	
<code>\msg_warning:nnn</code>	162, 414, 655	
<code>\msg_warning:nnnn</code>	1006, 1053, 1296, 1340,	
N		
<code>naturalnames</code> (hypersetup key)	1200	
<code>nesting</code> (hypersetup key)	11	
<code>\newcommand</code>	143, 3236, 3238, 3240	
<code>\newcount</code>	2392, 2563	
<code>\NewDocumentCommand</code>	2186	
<code>\newlength</code>	3255	
<code>\noexpand</code> ...	2474, 2475, 2476, 2477, 2803	
O		
<code>ocgcolorfile</code> (hypersetup key)	1000	
<code>ocgcolorlink</code> (hypersetup key)	1000	
<code>ocgcolorlinks</code> (hypersetup key)	1000	
<code>ocgcolormenu</code> (hypersetup key)	1000	
<code>ocgcolorrun</code> (hypersetup key)	1000	
<code>ocgcolorurl</code> (hypersetup key)	1000	
P		
page/Trans internal commands:		
<code>l__hyp_page/Trans</code>	315	
<code>pageanchor</code> (hypersetup key)	1200	
<code>\paperwidth</code>	3	
<code>\PassOptionsToPackage</code> ...	209, 1192, 1197	
pdf commands:		
<code>\pdf_bdc_obj:nn</code>	986, 989	
<code>\pdf_destination:nn</code>	19, 136, 384	
<code>\pdf_destination:nnnn</code>	377	
<code>\pdf_emc:</code>	988, 994	
<code>\pdf_link_last:</code>	2233, 2250	
<code>\pdf_link_user:nnn</code>	2570	
<code>\pdf_object_last:</code>	2112	
<code>\pdf_object_new:nn</code>	908, 909, 910, 911, 2725, 2726, 2727	
<code>\pdf_object_now:nn</code>	2108	
<code>\pdf_object_pageref:n</code>	1681, 1706, 2407	
<code>\pdf_object_ref:n</code>	914, 916, 938, 939, 942, 946, 951, 956, 961, 2311, 2313, 2783	
<code>\pdf_object_write:nn</code>	912, 918, 928, 940, 2728, 2770, 2777	
<code>\pdf_text_convert:nn</code>	635, 1949	
<code>\pdf_text_convert:nnN</code>	328	
<code>\pdf_version:</code>	1344, 1463, 1492, 1526, 1541, 1567, 1592, 1625, 1641, 1746, 1773	
<code>\pdf_version_compare:NnTF</code>	1000, 1047,	

1334, 1425, 1447, 1483, 1516, 1531,
 1557, 1582, 1608, 1631, 1736, 1763
 \pdf_version_compare_p:Nn 584
 \pdf_version_major: .. 150, 1009, 1057
 \pdf_version_minor: .. 149, 1009, 1057
 \pdf_xform_new:nnn 2188
 \pdf_xform_ref:n
 .. 2550, 2551, 2645, 2646, 2930, 2931
 pdfannot commands:
 \pdfannot_box:nnnn
 .. 2442, 2540, 2607, 2638, 2677, 2710
 \pdfannot_box_last: 2234, 2251
 \pdfannot_dict_put:nnn
 .. 792, 827, 851, 873, 897, 1075, 1106
 \pdfannot_dict_remove:nn
 .. 786, 822, 843, 868, 889, 1083, 1116
 \pdfannot_link:nnn . 495, 547, 599, 641
 \pdfannot_link_goto_begin:nw .. 427
 \pdfannot_link_goto_end: 432
 \pdfannot_link_margin:n 5, 1378
 \c_pdfannot_link_types_seq 296
 pdfauthor (hypersetup key) 1905
 \pdfbookmark 3241
 pdfborder (hypersetup key) 11
 pdfborderstyle (hypersetup key) .. 11, 860
 pdfcreationdate (hypersetup key) 11, 1961
 pdfcreator (hypersetup key) 1905
 \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, 634, 639,
 1404, 1405, 1409, 1410, 2121, 2131,
 2135, 2138, 2140, 2159, 2168, 2170
 \pdfdict_remove:nn ... 587, 1414, 1415
 \pdfdict_use:n 499, 551, 604, 645, 2110
 pdfencoding (hypersetup key) 1168
 \pdfescapestring 2792
 PDFForm internal commands:
 \PDFForm_hypName
 2831, 2834, 2838, 2841
 pdfinfo (hypersetup key) 2023
 pdfkeywords (hypersetup key) 1905
 pdflang (hypersetup key) 11, 1891
 pdflinkmargin (hypersetup key) 11
 pdfmanagement commands:
 \pdfmanagement_add:nn 1387
 \pdfmanagement_add:nnn
 351, 938, 939,
 961, 1263, 1288, 1304, 1308, 1328,
 1336, 1365, 1420, 1430, 1454, 1468,
 1480, 1485, 1510, 1518, 1533, 1559,
 1584, 1616, 1633, 1679, 1704, 1718,
 1738, 1765, 1794, 1900, 1919, 1937,
 1948, 1972, 1986, 2004, 2093, 2112,
 2286, 2303, 2310, 2312, 2314, 2323
 \pdfmanagement_if_active_p: ... 201
 \pdfmanagement_remove:nn
 ... 1260, 1284, 1292, 1312, 1324,
 1349, 1361, 1369, 1383, 1392, 1436,
 1451, 1470, 1507, 1546, 1571, 1597,
 1612, 1646, 1676, 1701, 1714, 1723,
 1751, 1778, 1790, 1799, 1897, 1913,
 1931, 1969, 1983, 2090, 2103, 2318
 \pdfmanagement_show:n 2287
 pdfmeta commands:
 \pdfmeta_if_named_action_
 allowed:nTF 631
 \g_pdfmeta_standard_prop
 .. 2316, 2586, 2665, 2899, 2969, 3164
 pdfmoddate (hypersetup key) 11, 1961
 pdfproducer (hypersetup key) 1905
 pdfremotestartview (hypersetup key) ... 9
 pdfstartview (hypersetup key) 9
 \pdfstringdef 6, 7, 15, 21, 2848
 pdfsubject (hypersetup key) 1905
 pdftitle (hypersetup key) 1905
 pdftrapped (hypersetup key) 1995
 pdfversion (hypersetup key) 1168
 pdfview (hypersetup key) 9, 1809
 \phantom 2347, 2371
 \phantomsection 3324
 plainpages (hypersetup key) 1200
 prg commands:
 \prg_generate_conditional_
 variant:Nnn 220
 \prg_new_eq_conditional:NNn ... 219
 prop commands:
 \prop_const_from_keyval:Nn
 253, 261, 3385, 3400,
 3415, 3430, 3445, 3460, 3476, 3491
 \prop_gput:Nnn 13, 2206, 2213
 \prop_if_empty:NnTF 2289
 \prop_if_in:NnTF
 .. 2316, 2586, 2665, 2899, 2969, 3164
 \prop_item:Nn 2306
 \prop_map_inline:Nn 714, 778, 814,
 841, 849, 860, 887, 895, 964, 1068,
 1104, 1114, 1158, 1246, 2284, 2291
 \prop_new:N 2192, 2193
 \protect 225, 3257
 \providecommand
 141, 142, 2220, 2223, 2452, 2788
 \ProvidesFile 3

R		
<code>\ReadBookmarks</code>	3246	
ref commands:		
<code>\ref_if_exist:nn</code>	219	
<code>\ref_label:nn</code>	215	
<code>\ref_value:nn</code>	17, 236	
regex commands:		
<code>\regex_const:Nn</code>	302	
<code>\regex_extract_once:NnNTF</code>	1659, 1690	
<code>\relax</code>	2455, 2456, 2497, 2785, 2832, 2836, 2906, 2910, 3015, 3016, 3024, 3028, 3055, 3059, 3139, 3147, 3176, 3184, 3308, 3317	
<code>\RemoveFromHook</code>	1268	
<code>\RenewDocumentCommand</code>	156	
<code>\RequirePackage</code>	7, 20, 205, 3244	
<code>run</code> (hypersetup key)	7, 1246	
<code>runborder</code> (hypersetup key)	11	
<code>runborderstyle</code> (hypersetup key)	11, 860	
<code>runcolor</code> (hypersetup key)	750	
		<code>\str_gset_eq:NN</code> 338
		<code>\str_head:n</code> 2006, 2011
		<code>\str_if_eq:nnTF</code>
		374, 1917, 1935, 1945, 2897
		<code>\str_if_eq_p:nn</code> 582
		<code>\str_lowercase:n</code>
		1814, 1940, 1988, 2007, 2012
		<code>\str_new:N</code> 243, 248, 249, 287
		<code>\str_remove_once:Nn</code> 488, 543, 595, 638
		<code>\str_set:Nn</code> 320
		<code>\str_set_convert:Nnnn</code> 6
		<code>\str_set_eq:NN</code> 340
		<code>\str_tail:n</code> 2007, 2012
		<code>\str_uppercase:n</code> 2000, 2006, 2011
		<code>\string</code> 2220,
		2223, 2245, 2264, 2808, 2812, 2823
		<code>\subpdfbookmark</code> 3237
		sys commands:
		<code>\c_sys_backend_str</code> 3199
S		
<code>\selectfont</code>	2334, 2346, 2358, 2370	
seq commands:		
<code>\seq_const_from_clist:Nn</code>	251	
<code>\seq_count:N</code>	1818, 1844	
<code>\seq_get_right:NN</code>	1820	
<code>\seq_item:Nn</code>	1661, 1692, 1814, 1862, 1864, 1874, 1879	
<code>\seq_map_inline:Nn</code>	288, 292, 296, 750, 1045, 1141, 2301	
<code>\seq_new:N</code>	241	
<code>\seq_put_right:Nn</code>	1851, 1852, 1853, 2293	
<code>\seq_set_split:Nnn</code>	1813	
<code>\seq_sort:Nn</code>	2295	
<code>\setlength</code>	3274, 3345	
<code>\setpdflinkmargin</code>	5	
<code>\settowidth</code>	2467	
<code>\show</code>	2528	
sort commands:		
<code>\sort_return_same:</code>	2299	
<code>\sort_return_swapped:</code>	2298	
<code>\space</code>	229, 2536, 3017, 3034, 3035, 3065, 3066, 3140, 3174, 3177	
<code>\spacefactor</code>	367, 389	
<code>\stockwidth</code>	3	
str commands:		
<code>\c_colon_str</code>	617	
<code>\c_hash_str</code>	491	
<code>\c_percent_str</code>	492	
<code>\str_case:nnTF</code>	3199	
<code>\str_case_e:nnTF</code>	1814	
<code>\str_gset:Nn</code>	775	
		T
		TeX and L ^A T _E X 2 _ε commands:
		<code>\@BOOKMARK</code> 3243
		<code>\@CheckBox</code> 2688
		<code>\@ChoiceMenu</code> 2453
		<code>\@Form</code> 2277
		<code>\@PushButton</code> 2578
		<code>\@Reset</code> 2655
		<code>\@Submit</code> 2619
		<code>\@TextField</code> 2418
		<code>\@bookmarksopenlevel</code> 1279
		<code>\@bsphack</code> 214
		<code>\@chapter</code> 3297, 3298
		<code>\@currDisplay</code> 2467, 2536
		<code>\@currValue</code> 2534, 2550
		<code>\@currentHref</code>
		3278, 3291, 3302, 3332, 3349
		<code>\@curropt</code> 2452, 2462, 2465, 2530, 2533
		<code>\@empty</code> 2421, 2437, 2853, 2857, 2861, 2865, 2869, 2873, 2877, 2881, 2885, 2889, 3035, 3066, 3067
		<code>\@endForm</code> 2387
		<code>\@esphack</code> 216
		<code>\@expl@color@set@n</code> 9
		<code>\@firstofone</code> 2226, 2794, 2799, 2816
		<code>\@firstoftwo</code> 3339
		<code>\@for</code> 2463, 2531
		<code>\@gobble</code> 2279
		<code>\@gobbletwo</code> 3288
		<code>\@hyper@launch</code> 29, 617
		<code>\@ifundefined</code> 2279, 2808
		<code>\@latex@warning</code> 226
		<code>\@mainaux</code> 2219, 2222, 2243, 2263
		<code>\@mkboth</code> 3288

\@nil	2806, 2819, 2825	\Fld@keystroke@code	2853, 2855
\@part	3315, 3322	\Fld@listcount	2527, 2535, 2556, 2563
\@pdfauthor	16	\Fld@mappingname	2836, 2838, 2910, 2912
\@pdfborder	141	\Fld@maxlen	3122, 3124
\@pdfborderstyle	142	\Fld@menulength	2460, 2466, 2513
\@savsf	367, 389	\Fld@name	
\@schapter	3285, 3286	..	2419, 2454, 2579, 2690, 2831, 2905
\@secondoftwo	3341	\Fld@onblur@code	2873, 2875
\@sect	3336, 3337	\Fld@onclick@code	3000
\@spart	3328, 3329	\Fld@onenter@code	2885, 2887
\@ssect	3271, 3272	\Fld@onexit@code	2889, 2891
\@tempdima	2461, 2468, 2470, 2507,	\Fld@onfocus@code	2869, 2871
2508, 2509, 2513, 2514, 2515, 2516		\Fld@onmousedown@code	2877, 2879
\@tempdimb	2467, 2468	\Fld@onmouseup@code	2881, 2883
\@typeset@protect	3257	\Fld@pageobjref	
\@c@secnumdepth	3305, 3323, 3338	2389, 2403, 2788, 2920,
\@calc@bm@number	3249	2975, 3008, 3044, 3083, 3132, 3169	
\@check@bm@number	3248	\Fld@radiosymbol	3063
\Fld@additionalactions	2895, 2965,	\Fld@rotation	2935,
3001, 3037, 3074, 3121, 3161, 3192		2937, 2983, 2990, 2992, 3014, 3020,	
\Fld@align	2924, 3012, 3087	3022, 3051, 3053, 3091, 3100, 3102,	
\Fld@altname ..	2832, 2834, 2906, 2908	3138, 3143, 3145, 3175, 3180, 3182	
\Fld@annotflags	2919,	\Fld@submitflags	3159
2974, 3007, 3043, 3082, 3131, 3168		\Fld@validate@code	2861, 2863
\Fld@annotnames	2904, 2921,	\Fld@value	
2976, 3009, 3045, 3084, 3133, 3170		..	2421, 2437, 2456, 2497, 2498, 3120
\Fld@bcolor	2943, 2945, 3028,	\Fld@width	2422,
3030, 3059, 3061, 3095, 3108, 3110		2436, 2443, 2447, 2457, 2496, 2507,	
\fld@bcolor	3016	2509, 2541, 2555, 2573, 2599, 2621,	
\Fld@bordercolor	2939,	2632, 2657, 2673, 2693, 2703, 2711	
2941, 2985, 2994, 2996, 3015, 3024,		\Fld@X@additionalactions	
3026, 3055, 3057, 3093, 3104, 3106,		2852, 2897, 2900
3139, 3147, 3149, 3176, 3184, 3186		\Form@action	3157
\Fld@borderstyle	2925,	\G@refundefinedtrue	225
2980, 3013, 3049, 3088, 3137, 3190		\H@old@part	3315, 3326
\Fld@borderwidth ..	2514, 2515, 2925,	\H@old@schapter	3285, 3294
2980, 3013, 3049, 3088, 3137, 3190		\H@old@sect	3336, 3347, 3356
\Fld@calculate@code ..	2236, 2865, 2867	\H@old@spart	3328, 3334
\Fld@calculate@sortkey	2247	\H@old@ssect	3271, 3276
\Fld@cbsymbol	2947	\Hy@abspage	361
\Fld@charsize		\Hy@activeanchorfalse	406
..	2513, 2951, 3034, 3065, 3116, 3174	\Hy@activeanchortrue	399
\Fld@checkequals	2465, 2533	\Hy@AtBeginDocument	2200, 2216
\Fld@choices	3036	\Hy@bookmarksnumberedfalse	1272
\Fld@color	2952, 2954, 3035, 3066, 3117	\Hy@bookmarksnumberedtrue	1273
\Fld@default		\Hy@bookmarksopenfalse	1276
... ..	2420, 2437, 2455, 2498, 2528,	\Hy@bookmarksopentruetrue	1277
2529, 2691, 3067, 3071, 3072, 3119		\Hy@bookmarkstype	1280
\Fld@flags	2923,	\Hy@chapapp	3289, 3300
2978, 3011, 3047, 3086, 3135, 3172		\Hy@CleanupFile	522
\Fld@format@code	2857, 2859	\Hy@colorlink	26
\Fld@height		\Hy@currentbookmarklevel	3230
2423, 2444, 2447, 2458, 2516, 2542,		\Hy@DisableOption	148, 155
2555, 2573, 2622, 2658, 2694, 2712		\Hy@drafttrue	1191

\Hy@escapeform	2441, 2539, 2569, 2605, 2636, 2675, 2709, 2790, 2797, 2800
\Hy@escapestring	2792, 2794, 2799, 2802, 2803, 2806, 2807, 2816, 2819, 2855, 2859, 2863, 2867, 2871, 2875, 2879, 2883, 2887, 2891, 2947, 3000, 3063, 3119, 3120, 3157
\Hy@finaltrue	1196
\Hy@FormObjects	2283, 2723, 2785
\Hy@gttemp	2848, 2850
\Hy@href@nextactionraw	486, 488, 489, 541, 543, 544, 593, 595, 596, 636, 638, 639
\Hy@href@page	536
\Hy@hypertexnamesfalse	1206
\Hy@hypertexnamestrue	1205
\Hy@linkfileprefix	1208
\Hy@linktoc	1231
\Hy@localanchornamefalse	1211
\Hy@localanchornametrue	1210
\Hy@MakeCurrentHrefAuto	3273, 3289, 3300, 3330, 3344
\Hy@MakeRemoteAction	532
\Hy@naturalnamesfalse	1215
\Hy@naturalnamestrue	1214
\Hy@next	3299, 3308, 3311
\Hy@numberline	144, 3231
\Hy@org@chapter	3297, 3312
\Hy@OutlineName	3247
\Hy@OutlineRerunCheck	3245
\Hy@pageanchorfalse	1219
\Hy@pageanchortrue	1218
\Hy@pdfmajorversion	150
\Hy@pdfminorversion	149
\Hy@pdfstringtrue	22, 333
\Hy@pdfversion	2843
\Hy@plainpagesfalse	1223
\Hy@plainpagetrue	1222
\Hy@pstringdef	343, 2823
\Hy@PutCatalog	349
\Hy@raisedlink	3265, 3290, 3301, 3331
\Hy@RestoreLastskip	388
\Hy@safe@activetrue	493, 2239
\Hy@SaveLastskip	368
\Hy@secnum@part	3318, 3320, 3323
\Hy@SectionAnchorHref	3256, 3278, 3349
\Hy@SectionHShift	3255, 3264, 3274, 3345
\Hy@StepCount	2466, 2535
\Hy@temp	2240, 2256, 2805, 2806
\Hy@unicodetrue	2845
\Hy@unicodetrue	146
\Hy@VerboseAnchor	369
\Hy@verbosefalse	1182
\Hy@VerboseLinkStart	439, 453
\Hy@VerboseLinkStop	445, 466, 505, 556, 609, 651
\Hy@verbosetrue	1181
\Hy@VersionChecked	137
\Hy@WrapperDef	365
\Hy@xspace@end	442, 504, 555, 608, 650
\HyAnn@AbsPageLabel	2388, 2394, 2440, 2538, 2568, 2604, 2635, 2674, 2708
\HyAnn@Count	2392, 2393, 2396, 2397, 2398, 2399, 2400, 2401, 2405, 2409
\HyField@AddToFields	2229, 2448, 2557, 2574, 2613, 2649, 2682, 2715
\HyField@afields	2195
\HyField@AfterAuxOpen	2200, 2226, 2261
\HyField@AuxAddToCoFields	2199, 2211, 2223, 2245
\HyField@AuxAddToFields	2198, 2204, 2220, 2264
\HyField@cofields	2196
\HyField@FlagsCheckBox	2700
\HyField@FlagsChoice	2506
\HyField@FlagsPushButton	2596, 2628, 2671
\HyField@FlagsRadioButton	2502
\HyField@FlagsSubmit	2629
\HyField@FlagsText	2435
\HyField@PDFChoices	2566
\HyField@SetKeys	2431, 2472, 2475, 2494, 2581, 2624, 2660, 2695
\HyPat@ObjRef	3203, 3210, 3217, 3224
\hyper@anchor	391
\hyper@anchorend	391, 3266, 3291, 3302, 3332
\hyper@anchorstart	391, 3266, 3291, 3302, 3332
\hyper@link	26, 26, 435
\hyper@linkend	25, 26, 460
\hyper@linkfile	516
\hyper@linklaunch	28, 566, 620
\hyper@linknamed	30, 158, 626
\hyper@linkstart	25, 26, 449
\hyper@linkurl	473
\HyPL@Labels	351, 361
\HyPL@StorePageLabel	349
\HyPsd@LoadUnicode	145
\HyPsd@pdfencoding	147
\HyPsd@pdfencoding@unicode	147
\HyPsd@SanitizeForOutFile	3234
\if@filesw	2218, 2242, 2262
\ifFld@combo	2481, 2511
\ifFld@hidden	2436, 2496

