

The `hyperref-generic` module

A generic driver for `hyperref`

The L^AT_EX Project*

Version 0.95a, released 2021-02-22

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

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

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

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

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

1 Avoiding transition problems

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

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

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

- This driver uses `l3color` for the colors. All colors defined with `\color_set:nn` or `\color_set:nnn` will work. Colors defined with `xcolor` will work if they don't use one of the special color models not supported by `l3color` as `pdfmanagement-firstaid` contains a patch for `xcolor`. If the package `color` is used it is currently recommended to define colors after `hyperref`. This driver uses `l3color` for the colors, some special color models of `xcolor` won't work.
- Load a color package or `graphicx` to get the right page sizes.
- Report problems! Only known problem can be resolved.

2 Bookmarks / outlines

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

3 “Metadata”

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

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

The current handling of the metadata is problematic:

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

To resolve some of this problem the driver will

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

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

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

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

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

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

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

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

4 Dates

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

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

5 PDF page size (mediabox)

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

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

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

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

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

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

Some main changes are

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

6.1 Background information

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

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

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

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

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

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

6.2 New Keys

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

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

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

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

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

6.3 Public interfaces

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

```
\l_hyp_annot_colorlink_bool,
\l_hyp_annot_colorurl_bool,
\l_hyp_annot_colorfile_bool,
\l_hyp_annot_colorrurl_bool,
\l_hyp_annot_colormenu_bool,
\l_hyp_annot_ocgcolorlink_bool,
\l_hyp_annot_ocgcolorurl_bool,
\l_hyp_annot_ocgcolorfile_bool,
\l_hyp_annot_ocgcolorrurl_bool,
\l_hyp_annot_ocgcolormenu_bool,
```

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

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

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

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

6.4 Changed behaviour

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

Color keys accept the following input syntax:

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

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

frenchlinks The option `frenchlinks` does nothing at all.

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

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

7 PDF strings

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

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

The new method is under heavy development!

Important differences here are

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

8 Package options from `hyperref`

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

Options that still must be set as package options are

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

- `encap`
- `hyperfigures` (according to the `hyperref` manual it makes figures hyper links, but actually is a no-op for most drivers, and it does nothing with this driver either.)
- `hyperfootnotes`
- `hyperindex`
- `implicit` (redefine L^AT_EX internals)
- `nesting` unneeded key, see comment below in 13. At some time it will be either removed or extended (if some use can be found).
- `pagebackref`
- `pdfpagelabels` (set PDF page labels)
- `psdextra` this loads some extra definitions used by `\pdfstringdef`. The new driver uses `\pdfstringdef` only for the bookmarks, for other strings it is not relevant.

Options that can be without problems set as package options are

- `debug`, `verbose` (a boolean)
- `bookmarks` (a boolean)
- `plainpages`
- `draft`, `final`
- `hypertextnames`
- `naturalnames`
- `pageanchor`

Ignored options:

- All driver options like `pdftex`, `dvipdfmx`, ...
- `raiselinks` (only used in the `dviwind`, `textures` and `tex4ht` driver anyway)
- `frenchlinks`
- `setpagesize`
- `addtopdfcreator`

9 Disabling links

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

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

10 Draftmode

`pdftex` and other engines knows a `draftmode` which can be set with `\pdfdraftmode=1` and `hyperref` honors this in some places. The new driver ignores it, for example `pagelabels` are created in any case. With today's computer power there is not much to gain and it only complicates the code.

This should not be confused with the `draft` and `final` package options! They are still honored.

11 Dropped options

A number of options are ignored by this driver

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

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

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

unicode This is always true.

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

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

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

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

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

12 Destinations

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

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

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

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

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

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

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

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

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

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

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

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

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

13 Assorted key descriptions

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

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

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

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

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

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

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

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

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

will create

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

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

pdfborder (*setup key*) This key set accept as value three numbers or three numbers and an array describing a dash pattern, examples are `0 0 1` or `0 0 1 [3 2]`. The first two numbers should according to the reference set round corners, but PDF viewer seem to ignore it. The **runborder** (*setup key*) third number is the line width of the border. Settings done with **pdfborderstyle** should take precedence.

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

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

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

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

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

File I

hyperref-generic driver implementation

```

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

```

1 messages

Redirect the message name:

```

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

```

At first a message for the testing of the resource management

```

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

```

The pdfversion should be set in \DeclareDocumentMetadata

```

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

```

A generic message for ignored keys.

```

37 \msg_new:nnn

```

```

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

```

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

```

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

```

pdf/A message for not allowed Named actions

```

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

```

A message if the destination name is empty.

```

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

```

A message if the destination check fails

```

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

```

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

```

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

```

```

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

```

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

```

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

```

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

```

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

```

```

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

```

2 Variants

```

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

```

3 Overwriting/providing commands from hyperref

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

```

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

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

```

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

```

139 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n

```

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

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

```

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

```

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

```

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

```

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

```

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

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

```

`\Acrobatmenu` should use the new internal link command

```

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

```

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

```

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

```

```

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

```

4 Compability commands

4.1 Metadata

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

```

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

```

4.2 citecolor

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

```

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

```

5 Checks

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

```

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

```

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

```

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

```

6 Reference and label commands

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

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

```

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

```

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

```

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

```

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

```

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

```

```

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

```

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

```

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

```

(End definition for __hyp_ref_label:nn and others.)

7 Variables

7.1 Private temporary variables

At first a few generic tmp variables

```

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

```

(End definition for \l__hyp_tmpa_tl and others.)

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

```

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

```

TODO: document and check use!

(End definition for \l__hyp_dest_name_tmpa_tl and others.)

7.2 Constants

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

```

249 \tl_const:Nn \c__hyp_dest_undefined_tl {UNDEFINED}

```

(End definition for \c__hyp_dest_undefined_tl.)

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

```

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

```

```

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

```

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

7.3 Variables

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

```

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

```

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

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

```

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

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

```

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

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

```

285 \tl_new:N \l__hyp_dest_pdfview_tl

```

(End definition for `\l__hyp_dest_pdfview_tl`.)

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

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

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

`\g__hyp_bordercolormodel_str`

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

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

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

```
286 \str_new:N \g__hyp_bordercolormodel_str
```

(End definition for `\g__hyp_bordercolormodel_str`.)

7.4 Booleans

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

`\l_hyp_annot_colorurl_bool`

`\l_hyp_annot_colorfile_bool`

`\l_hyp_annot_colorrrun_bool`

`\l_hyp_annot_colormenu_bool`

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

```
288 {
```

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

```
290 }
```

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

`\l_hyp_annot_ocgcolorlink_bool`

`\l_hyp_annot_ocgcolorurl_bool`

`\l_hyp_annot_ocgcolorfile_bool`

`\l_hyp_annot_ocgcolorrrun_bool`

`\l_hyp_annot_ocgcolormenu_bool`

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

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

```
292 {
```

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

```
294 }
```

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

`\not_Named_bool`, `\l_hyp_annot_Launch_bool`

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

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

```
296 {
```

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

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

```
299 }
```

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

7.5 Boxes

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

```
300 \box_new:N \l__hyp_dest_box
```

(End definition for `\l__hyp_dest_box`.)

7.6 Regex

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

```

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

(End definition for `\c__hyp_dest_startview_regex`.)

7.7 PDF dictionaries

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

```

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

(End definition for `l__hyp_page/Trans`.)

8 PDF string conversion

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

`hyp/text/pdfstring`

```

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

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

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

`__hyp_text_purify:nN`

```

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

(End definition for `__hyp_text_purify:nN`.)

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

`__hyp_text_cleanup:N`

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

(End definition for __hyp_text_cleanup:N.)

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

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

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

`__hyp_text_string_from_unicode:nN`

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

(End definition for __hyp_text_string_from_unicode:nN.)

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

`__hyp_text_pdfstring:nnN`

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

(End definition for __hyp_text_pdfstring:nnN.)

!!! temporary until all instances are gone

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

This is a special version for info keys:

```

\__hyp_text_pdfstring_info:nN

```

```

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

```

(End definition for __hyp_text_pdfstring_info:nN.)

9 Pagelabels

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

The current implementation in hyperref/hyperref drivers:

xetex: hxdetex.def, line 80-110

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

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

dvipdfm: identical to xetex

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

luatex identical to pdftex

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

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

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

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

```

\__hyp_PageLabels_gpush:

```

```

  \Hy@PutCatalog
  \HyPL@StorePageLabel

```

```

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

```

```

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

```

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

10 Core Hyperref Commands

Every hyperref has to define eight core command:

```

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

```

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

10.1 Anchors / destinations

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

```

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

```

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

```

\__hyp_destination:nn

```

```

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

```

```

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

```

(End definition for `__hyp_destination:nn`.)

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

```

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

```

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

10.2 GoTo Links

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

At first the internal link commands:

```

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

```

```

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

```

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

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

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

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

```

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

```

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

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

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

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

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

```

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

```

```

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

```

10.3 URI links

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

```

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

```

```

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

```

10.4 GoToR Links files

```

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

```

```

544     }
545     \mode_leave_vertical:
546     \pdfannot_link:nxn %expansion??
547     { GoToR }
548     {
549         /A<<
550         \pdfdict_use:n {l_hyp/annot/A/GoToR}
551         >>
552     }
553     {
554         #1\Hy@xspace@end
555         \Hy@VerboseLinkStop %where is the start??
556     }
557     \group_end:
558 }
559 {#1}
560 }

```

10.5 Launch links

We define \hyper@linklaunch for naming consistency

```

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

```

```

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

```

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

```

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

```

10.6 Named links (menu)

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

```

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

```

```

641         {
642             /A
643             <<
644             \pdfdict_use:n { l_hyp/annot/A/Named }
645             >>
646         }
647         {
648             #2
649             \Hy@xspace@end
650             \Hy@VerboseLinkStop
651         }
652     }
653     {
654         \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
655         #2
656     }
657     \group_end:
658 }
659 {#2}
660 }
661

```

11 Link decorations

11.1 Functions to export and select colors

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

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

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

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

```

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

```

```

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

```

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

__hyp_color_select:n __hyp_color_select:nN {<color>}

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

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

```

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

```

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

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

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

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

```

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

```

```

709     \color_set:nnn {#1}{#2}{#3}
710   }
711
712   \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.

```

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

```

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

```

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

```

`colorurl` (setup key)

`colorlink` (setup key)

`colorryn` (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)

```

749 \seq_map_inline:Nn \c__hyp_annot_types_seq
750 {

```

```

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

```

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*)

```

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

```

```

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

```

11.3.2 Borderwidth and -arc

```

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

```

```

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

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 859 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 860 {
runborderstyle (setup key) 861   \keys_define:nn { hyp / setup }
fileborderstyle (setup key) 862   {
menuborderstyle (setup key) 863     #1borderstyle .code:n =
864     {
865       \tl_if_empty:nTF { ##1 }
866       {
867         \pdfannot_dict_remove:nn
868         {link/#2}
869         { BS }
870       }
871       {
872         \pdfannot_dict_put:nnn
873         {link/#2}
874         { BS }
875         { <<##1>> }
876       }
877     }
878   }
879 }
880 \keys_define:nn { hyp / setup }
881 {
882   ,pdfborderstyle .code:n =
883   {
884     \tl_if_empty:nTF { #1 }
885     {
886       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
887       {
888         \pdfannot_dict_remove:nn
889         {link/##2}
890         { BS }
891       }
892     }
893     {
894       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
895       {

```

```

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

```

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.

```

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

```

```

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

```

(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.

```

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

```

```

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

```

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

```

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

```

```

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

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1067 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1068 {
1069   \keys_define:nn { hyp / setup }
1070   {
1071     ,#1highlight .choices:nn =
1072     { /I, /N, /O, /P }
1073     {
1074       \pdfannot_dict_put:nnn
1075       { link/#2 }
1076       { H }
1077       { ##1 }
1078     }
1079     ,#1highlight / .code:n =
1080     {
1081       \pdfannot_dict_remove:nn
1082       { link/#2 }
1083       { H }
1084     }
1085     ,#1highlight / unknown .code:n =
1086     {
1087       \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1088       { pdfhighlight }
1089       { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1090       { \exp_not:n {##1} }
1091     }
1092   }
1093 }

```

```

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

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1129 \keys_define:nn { hyp / setup }
hideurl (setup key) 1130 {
hidefile (setup key) 1131   hidelinks .meta:n =
hiderun (setup key) 1132   {
hidemenu (setup key) 1133     ,colorlinks      = false
1134     ,ocgcolorlinks = false
1135     ,pdfborder      = { 0~0~0 }
1136     ,pdfborderstyle=
1137   }
1138 }
1139
1140 \seq_map_inline:Nn \c__hyp_annot_types_seq
1141 {

```

```

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

```

11.7 color schemes and settings

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

`colorscheme (setup key)`

```

1153 \keys_define:nn { hyp / setup }
1154 {
1155   colorscheme .code:n =
1156   {
1157     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1158     {
1159       \keys_set:nn { hyp /setup }
1160       {
1161         ##1 = ##2
1162       }
1163     }
1164   }
1165 }
1166 \keys_set:nn { hyp / setup } {colorscheme=phetype}

```

12 Keys

12.1 Ignored keys

The following are ignored (with or without warnings)

`unicode (setup key)`

`pdfencoding (setup key)`

`pdfversion (setup key)`

```

1167 \keys_define:nn { hyp / setup }
1168 {
1169   ,unicode      .code:n = {}
1170   ,pdfencoding .code:n = {}
1171   ,pdfversion .code:n =
1172   {
1173     \msg_warning:nn { hyp }{ pdfversion-disabled }
1174   }
1175 }
1176 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
  debug (setup key) 1177 \keys_define:nn { hyp / setup }
  draft (setup key) 1178 {
  final (setup key) 1179   ,verbose .choice:
1180   ,verbose / true .code:n = { \Hy@verbosetrue}
1181   ,verbose / false .code:n = { \Hy@verbosefalse}
1182   ,verbose .default:n = {true}
1183   ,debug .meta:n = {verbose=#1}
1184   ,debug .default:n = {true}
1185 }
1186 \keys_define:nn { hyp / setup }
1187 {
1188   ,draft .code:n =
1189   {
1190     \Hy@drafttrue
1191     \PassOptionsToPackage{draft}{bookmark}
1192   }
1193   ,final .code:n =
1194   {
1195     \Hy@finaltrue
1196     \PassOptionsToPackage{final}{bookmark}
1197   }
1198 }

extension (setup key)
hypertextnames (setup key) 1199 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1200 {
  pageanchor (setup key) 1201   ,extension .tl_set:N = \XR@ext
  linktoc (setup key) 1202   ,extension .initial:n= pdf
  linktocpage (setup key) 1203   ,hypertextnames .choice:
  plainpages (setup key) 1204   ,hypertextnames / true .code:n = { \Hy@hypertextnamestrue}
localanchorname (setup key) 1205   ,hypertextnames / false .code:n = { \Hy@hypertextnamesfalse}
linkfileprefix (setup key) 1206   ,hypertextnames .default:n = {true}
1207   ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1208   ,localanchorname .choice:
1209   ,localanchorname / true .code:n = { \Hy@localanchorname true }
1210   ,localanchorname / false .code:n = { \Hy@localanchorname false }
1211   ,localanchorname .default:n = {true}
1212   ,naturalnames .choice:
1213   ,naturalnames / true .code:n = { \Hy@naturalnames true}
1214   ,naturalnames / false .code:n = { \Hy@naturalnames false}
1215   ,naturalnames .default:n = {true}
1216   ,pageanchor .choice:
1217   ,pageanchor / true .code:n = { \Hy@pageanchortrue}
1218   ,pageanchor / false .code:n = { \Hy@pageanchorfalse}
1219   ,pageanchor .default:n = {true}
1220   ,plainpages .choice:
1221   ,plainpages / true .code:n = { \Hy@plainpagestrue}
1222   ,plainpages / false .code:n = { \Hy@plainpagesfalse}
1223   ,plainpages .default:n = {true}

```

```

1224 }
1225
1226 \keys_define:nn { hyp / setup }
1227 {
1228   ,linktoc .choices:nn = { none, section, all, page }
1229   {
1230     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1231   }
1232   ,linktoc / unknown .code:n =
1233   {
1234     \msg_warning:nnxxx { hyp } { unknown-choice }
1235     { linktoc }
1236     { none, section, all, page }
1237     { \exp_not:n {#1} }
1238   }
1239   ,linktocpage .choice:
1240   ,linktocpage / true .meta:n = {linktoc=page}
1241   ,linktocpage / false .meta:n = {linktoc=section}
1242   ,linktocpage .default:n = true
1243 }
1244

```

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

```

url (setup key) 1245 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1246 {
menu (setup key) 1247   \keys_define:nn { hyp / setup }
run (setup key) 1248   {
1249     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1250   }
1251 }

```

```

1252 \keys_define:nn { hyp / setup }
1253 {
1254   ,baseurl .code:n =
1255   {
1256     \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1257     \tl_if_empty:NTF \l__hyp_tmpa_tl
1258     {
1259       \pdfmanagement_remove:nn {Catalog} { URI }
1260     }
1261     {
1262       \pdfmanagement_add:nnx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1263     }
1264   }
1265   %only false does something ...
1266   ,bookmarks .choice:
1267   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1268   ,bookmarks / true .code:n = {}
1269   ,bookmarks .default:n = {true}
1270   ,bookmarksnumbered .choice:
1271   ,bookmarksnumbered / false .code:n = { \Hy@bookmarksnumberedfalse }
1272   ,bookmarksnumbered / true .code:n = { \Hy@bookmarksnumberedtrue }
1273   ,bookmarksnumbered .default:n = {true}
1274   ,bookmarksopen .choice:

```

```

1275 ,bookmarksopen / false .code:n = { \Hy@bookmarksopenfalse }
1276 ,bookmarksopen / true .code:n = { \Hy@bookmarksopentruetrue }
1277 ,bookmarksopen .default:n = {true}
1278 ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1279 ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1280 ,pdfcenterwindow .choice:
1281 ,pdfcenterwindow / false .code:n =
1282 {
1283   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1284 }
1285 ,pdfcenterwindow / true .code:n =
1286 {
1287   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1288 }
1289 ,pdfcenterwindow / .code:n =
1290 {
1291   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1292 }
1293 ,pdfcenterwindow / unknown .code:n =
1294 {
1295   \msg_warning:nnxx { hyp } { no-bool }
1296   { pdfcenterwindow }
1297   { \exp_not:n {#1} }
1298 }
1299 ,pdfcenterwindow .default:n = true
1300 ,pdfdirection .choice:
1301 ,pdfdirection / L2R .code:n =
1302 {
1303   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1304 }
1305 ,pdfdirection / R2L .code:n =
1306 {
1307   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1308 }
1309 ,pdfdirection / .code:n =
1310 {
1311   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1312 }
1313 ,pdfdirection / unknown .code:n =
1314 {
1315   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1316   { pdfdirection }
1317   { L2R , R2L }
1318   { \exp_not:n {#1} }
1319 }
1320 ,pdfdisplaydoctitle .choice:
1321 ,pdfdisplaydoctitle / false .code:n =
1322 {
1323   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1324 }
1325 ,pdfdisplaydoctitle / true .code:n =
1326 {
1327   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1328 }

```

```

1329 ,pdffdisplaydoctitle .default:n = true
1330 ,pdfduplex .choices:nn =
1331 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1332 {
1333   \pdf_version_compare:NnTF > {1.6}
1334   {
1335     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1336     { PrintDuplex } { /#1 }
1337   }
1338   {
1339     \msg_warning:nnxx
1340     {hyp}
1341     {ignore-deprecated-or-unknown-option-in-pdf-version}
1342     {pdfduplex}
1343     {\pdf_version:}
1344   }
1345 }%
1346 ,pdfduplex / .code:n =
1347 {
1348   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1349 }
1350 ,pdfduplex / unknown .code:n =
1351 {
1352   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1353   { pdfduplex }
1354   { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1355   { \exp_not:n {#1} }
1356 }
1357 ,pdffitwindow .choice:
1358 ,pdffitwindow / false .code:n =
1359 {
1360   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1361 }
1362 ,pdffitwindow / true .code:n =
1363 {
1364   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1365 }
1366 ,pdffitwindow / .code:n =
1367 {
1368   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1369 }
1370 ,pdffitwindow .default:n = true
1371 ,pdffitwindow / unknown .code:n =
1372 {
1373   \msg_warning:nnxx { hyp } { no-bool }
1374   { pdffitwindow }
1375   { \exp_not:n {#1} }
1376 }
1377 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1378 ,pdflinkmargin .initial:n = {1pt}
1379 ,pdfmenubar .choice:
1380 ,pdfmenubar / true .code:n =
1381 {
1382   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }

```

```

1383     }
1384 ,pdfmenubar / false .code:n =
1385     {
1386         \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1387         { HideMenubar } { true }
1388     }
1389 ,pdfmenubar / .code:n =
1390     {
1391         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1392     }
1393 ,pdfmenubar .default:n = true
1394 ,pdfmenubar / unknown .code:n =
1395     {
1396         \msg_warning:nnxx { hyp } { no-bool }
1397         { pdfmenubar }
1398         { \exp_not:n {#1} }
1399     }
1400 ,pdfnewwindow .choice:
1401 ,pdfnewwindow / true .code:n =
1402     {
1403         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1404         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1405     }
1406 ,pdfnewwindow / false .code:n =
1407     {
1408         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1409         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1410     }
1411 ,pdfnewwindow / .code:n =
1412     {
1413         \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1414         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1415     }
1416 ,pdfnonfullscreenpagemode .choices:nn =
1417     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1418     {
1419         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1420         { NonFullScreenPageMode } {/#1}
1421     }
1422 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1423     {
1424         \pdf_version_compare:NnTF < {1.6}
1425         {
1426             %message
1427         }
1428         {
1429             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1430             { NonFullScreenPageMode } {/UseAttachments}
1431         }
1432     }
1433 ,pdfnonfullscreenpagemode / .code:n =
1434     {
1435         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1436     }

```

```

1437 ,pdfnonfullscreenpagemode / unknown .code:n =
1438 {
1439     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1440     { pdfnonfullscreenpagemode }
1441     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1442     { \exp_not:n {#1} }
1443 }
1444 ,pdfnumcopies .code:n =
1445 {
1446     \pdf_version_compare:NnTF > {1.6}
1447     {
1448         \tl_if_empty:nTF {#1}
1449         {
1450             \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1451         }
1452         {
1453             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1454             {NumCopies}{#1}
1455         }
1456     }
1457     {
1458         \msg_warning:nnxx
1459         {hyp}
1460         {ignore-deprecated-or-unknown-option-in-pdf-version}
1461         {pdfnumcopies}
1462         {\pdf_version:}
1463     }
1464 }
1465 ,pdfpagelayout .choices:nn =
1466 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1467 { \pdfmanagement_add:nnx {Catalog} { PageLayout }{ /#1 } }
1468 ,pdfpagelayout / .code:n =
1469 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1470 ,pdfpagelayout / unknown .code:n =
1471 {
1472     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1473     { pdfpagelayout }
1474     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1475     { \exp_not:n {#1} }
1476 }
1477 ,pdfpagemode .choices:nn =
1478 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1479 { \pdfmanagement_add:nnx {Catalog} { PageMode }{ /#1 } }
1480 ,pdfpagemode / UseAttachments .code:n =
1481 {
1482     \pdf_version_compare:NnTF > {1.5}
1483     {
1484         \pdfmanagement_add:nnx {Catalog} { PageMode }{ /UseAttachments }
1485     }
1486     {
1487         \msg_warning:nnxx
1488         {hyp}
1489         {ignore-deprecated-or-unknown-value-in-pdf-version}
1490         {UseAttachments}

```

```

1491         {\pdf_version:}
1492     }
1493 }
1494 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1495 ,pdfpagemode / unknown .code:n =
1496 {
1497     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1498     { pdfpagemode }
1499     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1500     { \exp_not:n {#1} }
1501 }
1502 ,pdfpagescrop .code:n =
1503 {
1504     \tl_if_empty:nTF %or blank?
1505     {
1506         \pdfmanagement_remove:nn {Pages} { CropBox }
1507     }
1508     {
1509         \pdfmanagement_add:nnx {Pages} { CropBox } { [#1] }
1510     }
1511 }
1512 ,pdfpicktraybypdfsize .choice:
1513 ,pdfpicktraybypdfsize / true .code:n =
1514 {
1515     \pdf_version_compare:NnTF > {1.6}
1516     {
1517         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1518         { PickTrayByPDFSize } { true }
1519     }
1520     {
1521         \msg_warning:nnxx
1522         {hyp}
1523         {ignore-deprecated-or-unknown-option-in-pdf-version}
1524         {pdfpicktraybypdfsize}
1525         {\pdf_version:}
1526     }
1527 }
1528 ,pdfpicktraybypdfsize / false .code:n =
1529 {
1530     \pdf_version_compare:NnTF > {1.6}
1531     {
1532         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1533         { PickTrayByPDFSize } { false }
1534     }
1535     {
1536         \msg_warning:nnxx
1537         {hyp}
1538         {ignore-deprecated-or-unknown-option-in-pdf-version}
1539         {pdfpicktraybypdfsize}
1540         {\pdf_version:}
1541     }
1542 }
1543 ,pdfpicktraybypdfsize / .code:n =
1544 {

```

```

1545     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1546   }
1547 ,pdfpicktraybypdfsize / unknown .code:n =
1548   {
1549     \msg_warning:nnxx { hyp } { no-bool }
1550     { picktraybypdfsize }
1551     { \exp_not:n {#1} }
1552   }
1553 ,pdfprintarea .choices:nn =
1554   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1555   {
1556     \pdf_version_compare:NnTF < {2.0}
1557     {
1558       \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1559       { PrintArea } { /#1 }
1560     }
1561     {
1562       \msg_warning:nnxx
1563       {hyp}
1564       {ignore-deprecated-or-unknown-option-in-pdf-version}
1565       {pdfprintarea}
1566       {\pdf_version:}
1567     }
1568   }%
1569 ,pdfprintarea / .code:n =
1570   { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1571 ,pdfprintarea / unknown .code:n =
1572   {
1573     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1574     { pdfprintarea }
1575     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1576     { \exp_not:n {#1} }
1577   }
1578 ,pdfprintclip .choices:nn =
1579   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1580   {
1581     \pdf_version_compare:NnTF < {2.0}
1582     {
1583       \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1584       { PrintClip } { /#1 }
1585     }
1586     {
1587       \msg_warning:nnxx
1588       {hyp}
1589       {ignore-deprecated-or-unknown-option-in-pdf-version}
1590       {pdfprintclip}
1591       {\pdf_version:}
1592     }
1593   }%
1594 ,pdfprintclip / .code:n =
1595   {
1596     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1597   }
1598 ,pdfprintclip / unknown .code:n =

```

```

1599     {
1600         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1601         { pdfprintclip }
1602         { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1603         { \exp_not:n {#1} }
1604     }
1605 ,pdfprintpagerange .code:n =
1606     {
1607         \pdf_version_compare:NnTF > {1.6}
1608         {
1609             \tl_if_empty:nTF { #1}
1610             {
1611                 \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1612                 { PrintPageRange }
1613             }
1614             {
1615                 \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1616                 {PrintPageRange}{[#1]}
1617             }
1618         }
1619         {
1620             \msg_warning:nnxx
1621             {hyp}
1622             {ignore-deprecated-or-unknown-option-in-pdf-version}
1623             {pdfprintpagerange}
1624             {\pdf_version:}
1625         }
1626     }
1627 ,pdfprintscaling .choices:nn =
1628     { None, AppDefault }
1629     {
1630         \pdf_version_compare:NnTF > {1.5}
1631         {
1632             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1633             { PrintScaling } { /#1 }
1634         }
1635         {
1636             \msg_warning:nnxx
1637             {hyp}
1638             {ignore-deprecated-or-unknown-option-in-pdf-version}
1639             {pdfprintscaling}
1640             {\pdf_version:}
1641         }
1642     }%
1643 ,pdfprintscaling / .code:n =
1644     {
1645         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1646     }
1647 ,pdfprintscaling / unknown .code:n =
1648     {
1649         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1650         { pdfprintarea }
1651         { None, AppDefault }
1652         { \exp_not:n {#1} }

```

```

1653     }
1654 ,pdfremotestartview .code:n =
1655 {
1656     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1657     \exp_args:NNV
1658     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1659     {
1660         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1661     }
1662     {
1663         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1664         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1665     }
1666 }
1667 ,pdfremotestartview .initial:n = {Fit}
1668 % pdfstartpage is special as it shares code with pdfstartview
1669 ,pdfstartpage .code:n =
1670 {
1671     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1672     \bool_if:nTF
1673     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1674     {
1675         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1676     }
1677     {
1678         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1679         {
1680             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1681         }
1682     }
1683 }
1684 ,pdfstartpage .initial:n =1
1685 ,pdfstartview .code:n =
1686 {
1687     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1688     \exp_args:NNV
1689     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1690     {
1691         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1692     }
1693     {
1694         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1695         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1696     }
1697     \bool_if:nTF
1698     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1699     {
1700         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1701     }
1702     {
1703         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1704         {
1705             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1706         }

```

```

1707     }
1708 }
1709 ,pdfstartview .initial:n = Fit
1710 ,pdftoolbar .choice:
1711 ,pdftoolbar / true .code:n =
1712 {
1713     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1714 }
1715 ,pdftoolbar / false .code:n =
1716 {
1717     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1718     { HideToolbar } { true }
1719 }
1720 ,pdftoolbar / true .code:n =
1721 {
1722     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1723 }
1724 ,pdftoolbar .default:n = true
1725 ,pdftoolbar / unknown .code:n =
1726 {
1727     \msg_warning:nxxx { hyp } { no-bool }
1728     { pdftoolbar }
1729     { \exp_not:n {#1} }
1730 }
1731 % pdfview see below.
1732 ,pdfviewarea .choices:nn =
1733 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1734 {
1735     \pdf_version_compare:NnTF < {2.0}
1736     {
1737         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1738         { ViewArea } { /#1 }
1739     }
1740     {
1741         \msg_warning:nxxx
1742         {hyp}
1743         {ignore-deprecated-or-unknown-option-in-pdf-version}
1744         {pdfviewarea}
1745         {\pdf_version:}
1746     }
1747 }%
1748 ,pdfviewarea / .code:n =
1749 {
1750     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1751 }
1752 ,pdfviewarea / unknown .code:n =
1753 {
1754     \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1755     { pdfviewarea }
1756     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1757     { \exp_not:n {#1} }
1758 }
1759 ,pdfviewclip .choices:nn =
1760 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }

```

```

1761 {
1762   \pdf_version_compare:NnTF < {2.0}
1763   {
1764     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1765     { ViewClip } { /#1 }
1766   }
1767   {
1768     \msg_warning:nnxx
1769     {hyp}
1770     {ignore-deprecated-or-unknown-option-in-pdf-version}
1771     {pdfviewclip}
1772     {\pdf_version:}
1773   }
1774   }%
1775 ,pdfviewclip / .code:n =
1776 {
1777   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
1778 }
1779 ,pdfviewclip / unknown .code:n =
1780 {
1781   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1782   { pdfviewclip }
1783   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1784   { \exp_not:n {#1} }
1785 }
1786 ,pdfwindowui .choice:
1787 ,pdfwindowui / true .code:n =
1788 {
1789   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
1790 }
1791 ,pdfwindowui / false .code:n =
1792 {
1793   \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1794   { HideWindowUI } { true }
1795 }
1796 ,pdfwindowui / .code:n =
1797 {
1798   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
1799 }
1800 ,pdfwindowui / unknown .code:n =
1801 {
1802   \msg_warning:nnxx { hyp } { no-bool }
1803   { pdfwindowui }
1804   { \exp_not:n {#1} }
1805 }
1806 ,pdfwindowui .default:n = true
1807 }

```

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

```

1808 \keys_define:nn { hyp / setup }
1809 {
1810   ,pdfview .code:n =
1811   {
1812     \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}

```

```

1813 \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
1814 {
1815   { xyz }
1816   {
1817     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
1818     {
1819       \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
1820       \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
1821       {
1822         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1823       }
1824       {
1825         \tl_set:Nx \l__hyp_dest_pdfview_tl
1826         {
1827           \fp_eval:n { \l__hyp_tmpa_tl * 100 }
1828         }
1829       }
1830     }
1831     {
1832       \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1833     }
1834   }
1835   { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
1836   { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
1837   { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
1838   { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
1839   { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
1840   { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
1841   { fitr }
1842   {
1843     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
1844     {
1845       \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
1846     }
1847     {
1848       %ensure 4 values ...
1849       \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
1850       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1851       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1852       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1853       \hbox_set_to_wd:Nnn \l__hyp_dest_box
1854       {
1855         \fp_eval:n
1856         {
1857           round
1858           (
1859             abs
1860             (
1861               \seq_item:Nn\l__hyp_tmpa_seq{4}
1862               -
1863               (\seq_item:Nn\l__hyp_tmpa_seq{2})
1864             ),
1865           3
1866         }

```

```

1867         }bp
1868     }{}
1869     \box_set_dp:Nn \l__hyp_dest_box
1870     {
1871         \fp_eval:n
1872         {
1873             round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
1874         }bp
1875     }
1876     \box_set_ht:Nn \l__hyp_dest_box
1877     {
1878         \seq_item:Nn\l__hyp_tmpa_seq{5}bp
1879     }
1880 }
1881 }
1882 }
1883 {
1884     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
1885     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
1886 }
1887 }
1888 ,pdfview .initial:n = {xyz}
1889 }

```

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.

```

1890 \keys_define:nn { hyp / setup }
1891 {
1892     ,pdflang .code:n =
1893     {
1894         \tl_if_empty:nTF { #1 }
1895         {
1896             \pdfmanagement_remove:nn {Catalog} { Lang }
1897         }
1898         {
1899             \pdfmanagement_add:nnx {Catalog} { Lang } { (#1) }
1900         }
1901         \__hyp_store_metadata:nn {pdflang}{#1}
1902     }
1903 }

```

12.3.1 “info keys”

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

```

1904 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
1905 {
1906     \keys_define:nn { hyp / setup }
1907     {

```

```

1908 pdf#1 .code:n =
1909 {
1910     \tl_if_blank:nTF {##1}
1911     {
1912         \pdfmanagement_remove:nn {Info}{#2}
1913     }
1914     {
1915         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
1916         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
1917         {
1918             \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
1919         }
1920     }
1921     \__hyp_store_metadata:nn {pdf#1}{##1}
1922 }
1923 }
1924 \keys_define:nn { hyp / info }
1925 {
1926     #2 .code:n =
1927     {
1928         \tl_if_blank:nTF {##1}
1929         {
1930             \pdfmanagement_remove:nn {Info}{#2}
1931         }
1932         {
1933             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
1934             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
1935             {
1936                 \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
1937             }
1938         }
1939         \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
1940     }
1941     ,unknown .code:n =
1942     {
1943         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
1944         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
1945         {
1946             \exp_args:Nno
1947             \pdfmanagement_add:nnx {Info}
1948             { \l_keys_key_str } {\l__hyp_tmpa_str}
1949         }
1950     }
1951 }
1952 }
1953 \__hyp_setup_info_key:nn {author} {Author}
1954 \__hyp_setup_info_key:nn {title} {Title}
1955 \__hyp_setup_info_key:nn {producer} {Producer}
1956 \__hyp_setup_info_key:nn {creator} {Creator}
1957 % ignored key: addtopdfcreator
1958 \__hyp_setup_info_key:nn {subject} {Subject}
1959 \__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.

```

1960 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
1961 {
1962   \keys_define:nn { hyp / setup }
1963   {
1964     pdf#1 .code:n =
1965     {
1966       \tl_if_blank:nTF {##1}
1967       {
1968         \pdfmanagement_remove:nn {Info}{#2}
1969       }
1970       {
1971         \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
1972       }
1973       \__hyp_store_metadata:nn {pdf#1}{##1}
1974     }
1975   }
1976   \keys_define:nn { hyp / info }
1977   {
1978     #2 .code:n =
1979     {
1980       \tl_if_blank:nTF {##1}
1981       {
1982         \pdfmanagement_remove:nn {Info}{#2}
1983       }
1984       {
1985         \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
1986       }
1987       \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
1988     }
1989   }
1990 }
1991
1992 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
1993 \__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 ...

```

1994 \keys_define:nn { hyp / setup }
1995 {
1996   ,pdftrapped .code:n =
1997   {
1998     \exp_args:Nne
1999     \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2000   }
2001   ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2002   {
2003     \pdfmanagement_add:nnx {Info}{Trapped}
2004     {/
2005       \str_uppercase:f { \str_head:n { #1 } }
2006       \str_lowercase:f { \str_tail:n { #1 } }
2007     }
2008     \__hyp_store_metadata:nx {pdftrapped}
2009     {
2010       \str_uppercase:f { \str_head:n { #1 } }

```

```

2011         \str_lowercase:f { \str_tail:n { #1 } }
2012     }
2013 }
2014 ,_pdftrapped / unknown .code:n =
2015 {
2016     \msg_warning:nnxxx { hyp } { unknown-choice }
2017     { pdftrapped }
2018     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2019     { \exp_not:n {#1} }
2020 }
2021 }

```

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

```

2022 \keys_define:nn { hyp / setup }
2023 {
2024     pdfinfo .code:n =
2025     {
2026         \keys_set:nn { hyp / info } { #1 }
2027     }
2028 }

```

Now we set some default values

```

2029 \keys_set:nn { hyp / setup } {pdfcreator = LaTeX-with-hyperref}
2030 \keys_set:nn { hyp / setup } {pdfauthor  = }
2031 \keys_set:nn { hyp / setup } {pdftitle   = }
2032 \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:

```

2033 \clist_map_inline:nn
2034 {
2035     ,pdfcopyright
2036     ,pdftype
2037     ,pdflicenseurl
2038     ,pdfauthortitle
2039     ,pdfcaptionwriter
2040     ,pdfmetalang
2041     ,pdfapart
2042     ,pdfaconformance
2043     ,pdfuapart
2044     ,pdfxstandard
2045     ,pdfsource
2046     ,pdfdocumentid
2047     ,pdfinstanceid
2048     ,pdfversionid
2049     ,pdfrendition
2050     ,pdfpublication
2051     ,pdfpubtype
2052     ,pdfbytes
2053     ,pdfnumpages
2054     ,pdfissn

```

```

2055 ,pdfeissn
2056 ,pdfisbn
2057 ,pdfbookedition
2058 ,pdfpublisher
2059 ,pdfvolumenum
2060 ,pdfissuenum
2061 ,pdfpagerange
2062 ,pdfdoi
2063 ,pdfurl
2064 ,pdfidentifier
2065 ,pdfsubtitle
2066 ,pdfpubstatus
2067 ,pdfcontactaddress
2068 ,pdfcontactcity
2069 ,pdfcontactregion
2070 ,pdfcontactpostcode
2071 ,pdfcontactcountry
2072 ,pdfcontactphone
2073 ,pdfcontactemail
2074 ,pdfcontacturl
2075 ,pdfdate
2076 }
2077 {
2078   \keys_define:nn { hyp / setup }
2079   {
2080     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2081   }
2082 }

```

12.5 Transitions

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

```

2083 \keys_define:nn { hyp / setup }
2084 {
2085   pdfpageduration .code:n =
2086   {
2087     \tl_if_blank:nTF { #1 }
2088     {
2089       \pdfmanagement_remove:nn {Page}{Dur}
2090     }
2091     {
2092       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2093     }
2094   }
2095 }

```

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

setting looks e.g. like this

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

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

The keys allowed in the argument of pdfpagetransition are

```
2096 \keys_define:nn { hyp / setup }
2097 {
2098   pdfpagetransition .code:n =
2099   {
2100     \tl_if_blank:nTF {#1}
2101     {
2102       \pdfmanagement_remove:nn {Page}{Trans}
2103     }
2104     {
2105       \group_begin:
2106       \keys_set:nn { hyp / trans }{style=R,#1}
2107       \pdf_object_unnamed_write:nx { dict }
2108       {
2109         \pdfdict_use:n {l__hyp_page/Trans}
2110       }
2111       \pdfmanagement_add:nnx {Page}{Trans}{\pdf_object_ref_last:}
2112       \group_end:
2113     }
2114   }
2115 }
2116 \keys_define:nn { hyp / trans }
2117 {
2118   ,style .choices:nn =
2119   {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2120   { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2121   ,style / unknown .code:n =
2122   {
2123     \msg_warning:nnxxx { hyp } { unknown-choice }
2124     { trans / style }
2125     { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2126     { \exp_not:n {#1} }
2127   }
2128   ,duration .code:n =
```

```

2129     {
2130         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{#1}
2131     }
2132     ,direction .choices:nn =
2133     {H,V}
2134     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{ /#1 } }
2135     ,direction .choices:nn =
2136     {0,90,180,270,315}
2137     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ #1 } }
2138     ,direction / None .code:n =
2139     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ /None } }
2140     ,direction / unknown .code:n =
2141     {
2142         \msg_warning:nnxxx { hyp } { unknown-choice }
2143         { trans / direction }
2144         {
2145             H~(horizontal,~only~Split,~Blinds),
2146             V~(vertical,~only~Split,~Blinds),
2147             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2148             90~(bottom~to~top,~only~Wipe),
2149             180~(right~to~left,~only~Wipe),
2150             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2151             315~(top~left~to~bottom,~only~Glitter),
2152             None~(only~Fly)
2153         }
2154         { \exp_not:n {#1} }
2155     }
2156     ,motion .choices:nn =
2157     {I,0}
2158     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{ /#1 } }
2159     ,motion / unknown .code:n =
2160     {
2161         \msg_warning:nnxxx { hyp } { unknown-choice }
2162         { trans / motion }
2163         { I~(inwards) , 0~(outwards) }
2164         { \exp_not:n {#1} }
2165     }
2166     ,scale .code:n =
2167     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2168     ,opaque .choices:nn = {true,false}
2169     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2170     ,opaque / unknown .code:n =
2171     {
2172         \msg_warning:nnxxx { hyp } { unknown-choice }
2173         { trans / B }
2174         { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2175         { \exp_not:n {#1} }
2176     }
2177     % try to set unknown keys as style
2178     ,unknown .code:n =
2179     {
2180         % warning ...
2181         \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2182     }

```

```

2183 }
    Finally we process the package option list, to get most keys working
2184 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}
    Unfinished Form field code
2185 \NewDocumentCommand \MakeFieldObject { m m }
2186 {
2187   \pdfxform_new:nnn { #2 }{} { #1 }
2188 }
2189
2190
2191 \prop_new:N    \g__hyp_AcroForm_CoFields_prop
2192 \prop_new:N    \g__hyp_AcroForm_Fields_prop
2193
2194 \let\HyField@afields\ltx@empty
2195 \let\HyField@cofields\ltx@empty
2196 %% UF test for old pdftex removed
2197 %\let\HyField@AuxAddToFields\ltx@gobble
2198 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2199 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2200
2201 % the value doesn't matter, but with a prop we avoid duplicates and it is
2202 % clearly faster than removing them from a sequence
2203 \def\HyField@AuxAddToFields#1
2204 {
2205   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2206 }%
2207
2208 %fields with empty key get a value too -- lets hope that
2209 %this give the expected behaviour
2210 \def\HyField@AuxAddToCoFields #1 #2
2211 {
2212   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2213 }
2214
2215 \Hy@AtBeginDocument
2216 {
2217   \if@files
2218     \immediate\write\@mainaux{%
2219       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2220     }%
2221     \immediate\write\@mainaux{%
2222       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2223     }%
2224   \fi
2225   \let\HyField@AfterAuxOpen\@firstofone
2226 }%
2227
2228 \def\HyField@AddToFields
2229 {
2230   \exp_args:Nx\HyField__hypAddToFields
2231   {
2232     \pdfannot_box_ref_last:
2233   }

```

```

2234 \ifx\Fld@calculate@code\ltx@empty
2235 \else
2236 \begingroup
2237 \Hy@safe@activetrue
2238 \edef\Hy@temp{%
2239 \endgroup
2240 \if@filesw
2241 \write\@mainaux
2242 {
2243 \string\HyField@AuxAddToCoFields
2244 {
2245 \Fld@calculate@sortkey
2246 }
2247 {
2248 \pdfannot_box_ref_last:
2249 }
2250 }
2251 \fi
2252 }%
2253 \Hy@temp
2254 \fi
2255 }%
2256
2257 \def\HyField__hypAddToFields#1{
2258 \HyField@AfterAuxOpen{%
2259 \if@filesw
2260 \write\@mainaux{%
2261 \string\HyField@AuxAddToFields{#1}%
2262 }%
2263 \fi
2264 }%
2265 }%
2266
2267 \ExplSyntaxOff
2268 \ExplSyntaxOn
2269 \tl_new:N \l__hyp_CheckmarkYes_tl
2270 \tl_set:Nn \l__hyp_CheckmarkYes_tl { __hyp_xform_CheckMarkYes }
2271 \tl_new:N \l__hyp_CheckmarkOff_tl
2272 \tl_set:Nn \l__hyp_CheckmarkOff_tl { __hyp_xform_CheckMarkOff }
2273
2274 \def\@Form[#1]
2275 {
2276 \@ifundefined{textcolor}{\let\textcolor\@gobble}{ }
2277 \kvsetkeys{Form}{#1}
2278 \pdf@ifdraftmode{ }
2279 {
2280 \Hy@FormObjects
2281 \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2282 {
2283 \pdfmanagement_add:nxx { Catalog / AcroForm } { Fields }{##1}
2284 %\pdfmanagement_show:n { Catalog / AcroForm }
2285 }
2286 \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2287 {

```

```

2288         \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2289         {
2290             \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2291         }
2292     \seq_sort:Nn \l__hyp_tmpa_seq
2293     {
2294         \int_compare:nNnTF { \pdf@strcmp{##1}{##2} } > { 0 }
2295         { \sort_return_swapped: }
2296         { \sort_return_same: }
2297     }
2298     \seq_map_inline:Nn \l__hyp_tmpa_seq
2299     {
2300         \pdfmanagement_add:nnx { Catalog / AcroForm }
2301         { CO }
2302         {
2303             \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2304         }
2305     }
2306 }
2307 \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2308 {ZaDb} {\pdf_object_ref:n {l__hyp_font_zapdingbats_obj} }
2309 \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2310 {Helv} {\pdf_object_ref:n {l__hyp_font_helvetica_obj} }
2311 \pdfmanagement_add:nnx {Catalog /AcroForm}
2312 {DA}{(/Helv~10~Tf~0~g)}
2313 \pdfmeta_standard_verify:nTF {form_no_NeedAppearance}
2314 {
2315     \legacy_if:nT { HyField@NeedAppearances }
2316     {
2317         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2318     }
2319 }
2320 {
2321     \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2322 }
2323 }
2324 \MakeFieldObject
2325 {
2326     \group_begin:
2327     \fontfamily{pzd}
2328     \fontencoding{U}
2329     \fontseries{m}
2330     \fontshape{n}
2331     \selectfont
2332     \char123
2333     \group_end:
2334 }
2335 {__hyp_xform_Ding}
2336 \MakeFieldObject
2337 {
2338     \group_begin:
2339     \fontfamily{pzd}
2340     \fontencoding{U}
2341     \fontseries{m}

```

```

2342         \fontshape{n}
2343         \selectfont
2344         \phantom{\char123}
2345         \group_end:
2346     }
2347     {__hyp_xform_DingOff}
2348 \MakeFieldObject
2349 {
2350     \group_begin:
2351     \fontfamily{pzd}
2352     \fontencoding{U}
2353     \fontseries{m}
2354     \fontshape{n}
2355     \selectfont
2356     \char51
2357     \group_end:
2358 }
2359 {__hyp_xform_CheckMarkYes}
2360 \MakeFieldObject
2361 {
2362     \group_begin:
2363     \fontfamily{pzd}
2364     \fontencoding{U}
2365     \fontseries{m}
2366     \fontshape{n}
2367     \selectfont
2368     \phantom{\char51} %perhaps xetex needs some small glyph ..
2369     \group_end:
2370 }
2371 {__hyp_xform_CheckMarkOff}
2372 \MakeFieldObject
2373 {
2374     \fbox{\textcolor{yellow}{\textsf{Submit}}}} %color?
2375 }
2376 {__hyp_xform_Submit}
2377 \MakeFieldObject
2378 {
2379     \fbox{\textcolor{yellow}{\textsf{SubmitP}}}} %color?
2380 }
2381 {__hyp_xform_SubmitP}
2382 }
2383 \ExplSyntaxOff
2384 \let\@endForm\ltx@empty
2385 \let\HyAnn@AbsPageLabel\ltx@empty
2386 \let\Fld@pageobjref\ltx@empty
2387
2388 \ExplSyntaxOn
2389 \newcount\HyAnn@Count
2390 \HyAnn@Count=\ltx@zero
2391 \def\HyAnn@AbsPageLabel
2392 {
2393     \global\advance\HyAnn@Count by\ltx@one
2394     %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2395     %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}

```

```

2396     \%zref@refused{HyAnn@\the\HyAnn@Count}%
2397     \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
2398     \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
2399 }%
2400 \def\Fld@pageobjref
2401 {
2402     \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
2403     {
2404         /P~\pdf_pageobject_ref:n
2405         {
2406             \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
2407         }
2408     }
2409 }
2410 \ExplSyntaxOff
2411 \ExplSyntaxOn
2412 %% check if the attr should be set through
2413 %% hooks.
2414 %% check if options are missing.
2415 \def\@TextField[#1]#2{% parameters, label
2416     \def\Fld@name{#2}%
2417     \let\Fld@default\ltx@empty
2418     \let\Fld@value\@empty
2419     \def\Fld@width{\DefaultWidthofText}%
2420     \def\Fld@height{%
2421         \ifFld@multiline
2422             \DefaultHeightofTextMultiline
2423         \else
2424             \DefaultHeightofText
2425         \fi
2426     }%
2427     \begingroup
2428     \expandafter\HyField@SetKeys\expandafter{%
2429         \DefaultOptionsofText,#1%
2430     }%
2431     \PDFForm@Name
2432     \HyField@FlagsText
2433     \ifFld@hidden\def\Fld@width{1sp}\fi
2434     \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2435     \LayoutTextField{#2}{%
2436         \leavevmode
2437         \HyAnn@AbsPageLabel
2438         \Hy@escapeform\PDFForm@Text
2439         \pdfannot_box:nnnn
2440         {\Fld@width}
2441         {\Fld@height}
2442         {Opt} %is this correct?
2443         {\PDFForm@Text}
2444         \MakeTextField{\Fld@width}{\Fld@height}
2445         \HyField@AddToFields
2446     }%
2447     \endgroup
2448 }
2449 \providecommand\@curropt{}

```

```

2450 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2451   \def\Fld@name{#2}
2452   \let\Fld@default\relax
2453   \let\Fld@value\relax
2454   \def\Fld@width{\DefaultWidthofChoiceMenu}
2455   \def\Fld@height{\DefaultHeightofChoiceMenu}
2456   \begingroup
2457     \Fld@menulength=0 %
2458     \@tempdima\z@
2459     \clist_map_variable:nNn { #3 } \@curopt
2460     %\@for\@curopt:=#3\do
2461     {%
2462       \expandafter\Fld@checkequals\@curopt==\%
2463       \Hy@StepCount\Fld@menulength
2464       \settowidth{\@tempdimb}{\@currDisplay}%
2465       \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2466     }%
2467     \advance\@tempdima by~15\p@
2468     \begingroup
2469       \HyField@SetKeys{#1}
2470     \edef\x{\endgroup
2471       \noexpand\expandafter
2472       \noexpand\HyField@SetKeys
2473       \noexpand\expandafter{%
2474         \expandafter\noexpand\csname DefaultOptionsof%
2475           \ifFld@radio
2476             Radio%
2477           \else
2478             \ifFld@combo
2479               \ifFld@popdown
2480                 PopdownBox%
2481               \else
2482                 ComboBox%
2483               \fi
2484             \else
2485               ListBox%
2486             \fi
2487           \fi
2488         \endcsname
2489       }%
2490     }\x
2491     \HyField@SetKeys{#1}%
2492     \PDFForm@Name
2493     \ifFld@hidden\def\Fld@width{1sp}\fi
2494     \ifx\Fld@value\relax
2495       \let\Fld@value\Fld@default
2496     \fi
2497     \LayoutChoiceField{#2}{%
2498       \ifFld@radio
2499         \HyField@FlagsRadioButton
2500         \__hypRadio{#3}%
2501       \else
2502         \begingroup
2503         \HyField@FlagsChoice

```

```

2504         \ifdim\Fld@width<\@tempdima
2505         \ifdim\@tempdima<1cm\@tempdima1cm\fi
2506         \edef\Fld@width{\the\@tempdima}%
2507     \fi
2508     \ifFld@combo
2509     \else
2510         \@tempdima=\the\Fld@menulength\Fld@charsize
2511         \advance\@tempdima by~\Fld@borderwidth bp %
2512         \advance\@tempdima by~\Fld@borderwidth bp %
2513         \edef\Fld@height{\the\@tempdima}%
2514     \fi
2515     \__hypListBox{#3}%
2516 \endgroup
2517 \fi
2518 }%
2519 \endgroup
2520 }
2521 \tl_new:N \l__hyp_RadioYes_tl
2522 \tl_set:Nn \l__hyp_RadioYes_tl { __hyp_xform_Ding }
2523 \def\__hypRadio#1{%
2524     \Fld@listcount=0-%
2525     %\show\Fld@default
2526     \EdefEscapeName\Fld@default{\Fld@default}%
2527     \clist_map_variable:nNn { #1 } \@curropt
2528     %\@for\@curropt:=#1\do
2529     {%
2530         \expandafter\Fld@checkequals\@curropt==\%
2531         \EdefEscapeName\@currValue{\@currValue}%
2532         \Hy@StepCount\Fld@listcount
2533         \@currDisplay\space
2534         \leavevmode
2535         \HyAnn@AbsPageLabel
2536         \Hy@escapeform\PDFForm@Radio
2537         \pdfannot_box:nnnn
2538             {\Fld@width}
2539             {\Fld@height}
2540             {Opt} %is this correct?
2541             {
2542                 \PDFForm@Radio
2543                 /AP
2544                 <<
2545                 /N
2546                 <<
2547                 /\@currValue\c_space_tl \pdfxform_ref:o {__hyp_xform_Ding}
2548                 %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2549                 >>
2550                 >>
2551             }
2552         {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2553         \int_compare:nNnT { \Fld@listcount } = { 1 }
2554         { \HyField@AddToFields }
2555         \c_space_tl % deliberate space between radio buttons
2556         % to do: --> should be configurable
2557     }%

```

```

2558 }
2559
2560 \newcount\Fld@listcount
2561 \def\__hypListbox#1
2562 {
2563   \HyField@PDFChoices{#1}
2564   \mode_leave_vertical:
2565   \HyAnn@AbsPageLabel
2566   \Hy@escapeform\PDFForm@List
2567   \pdf_link_user:nnn
2568     {widget} %perhaps we need more types??
2569     {\PDFForm@List}
2570     {\MakeChoiceField{\Fld@width}{\Fld@height}}
2571   \HyField@AddToFields
2572 }
2573
2574
2575 \def\@PushButton[#1]#2{% parameters, label
2576   \def\Fld@name{#2}%
2577   \group_begin:
2578     \exp_args:No\HyField@SetKeys
2579     {
2580       \DefaultOptionsofPushButton,#1
2581     }
2582     \PDFForm@Name
2583     \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2584     {
2585       \HyField@FlagsPushButton
2586       \legacy_if:nT {Fld@hidden}
2587       {
2588         \def\Fld@width{1sp}
2589       }
2590       \LayoutPushButtonField
2591       {
2592         \mode_leave_vertical:
2593         \HyAnn@AbsPageLabel
2594         \Hy@escapeform\PDFForm@Push
2595         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2596         \pdfannot_box:nnnn
2597           {\box_wd:N\l_tmpa_box}
2598           {\box_ht:N\l_tmpa_box}
2599           {\box_dp:N\l_tmpa_box} %is this correct?
2600           {\PDFForm@Push}
2601           {\box_use:N\l_tmpa_box}
2602         \HyField@AddToFields
2603       }
2604     }
2605     {
2606       \msg_error:nn { hyp }{ pdfa-no-push-button }
2607       \LayoutPushButtonField
2608       {
2609         \mode_leave_vertical:
2610         \MakeButtonField{#2}
2611       }

```

```

2612     }
2613     \group_end:
2614 }
2615
2616 \def\@Submit[#1]#2
2617 {
2618     \def\Fld@width {\DefaultWidthofSubmit}
2619     \def\Fld@height{\DefaultHeightofSubmit}
2620     \group_begin:
2621         \exp_args:No\HyField@SetKeys
2622         {
2623             \DefaultOptionsofSubmit,#1
2624         }
2625     \HyField@FlagsPushButton
2626     \HyField@FlagsSubmit
2627     \legacy_if:nT { Fld@hidden }
2628     {
2629         \def\Fld@width{1sp}
2630     }
2631     \mode_leave_vertical:
2632     \HyAnn@AbsPageLabel
2633     \Hy@escapeform\PDFForm@Submit
2634     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2635     \pdfannot_box:nnnn
2636         {\box_wd:N\l_tmpa_box}
2637         {\box_ht:N\l_tmpa_box}
2638         {\box_dp:N\l_tmpa_box} %is this correct?
2639     {
2640         \PDFForm@Submit
2641         /AP<<
2642             /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2643             /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2644         >>
2645     }
2646     \HyField@AddToFields
2647     \box_use:N\l_tmpa_box
2648
2649     \group_end:
2650 }
2651
2652 \def\@Reset[#1]#2
2653 {
2654     \def\Fld@width {\DefaultWidthofReset}
2655     \def\Fld@height{\DefaultHeightofReset}
2656     \group_begin:
2657         \exp_args:No\HyField@SetKeys
2658         {
2659             \DefaultOptionsofReset,#1
2660         }
2661     \mode_leave_vertical:
2662     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2663     {
2664         \HyField@FlagsPushButton
2665         \legacy_if:nT { Fld@hidden }

```

```

2666         { \def\Fld@width{1sp} }
2667         \HyAnn@AbsPageLabel
2668         \Hy@escapeform\PDFForm@Reset
2669         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2670         \pdfannot_box:nnnn
2671         {\box_wd:N\l_tmpa_box}
2672         {\box_ht:N\l_tmpa_box}
2673         {\box_dp:N\l_tmpa_box} %is this correct?
2674         { \PDFForm@Reset }
2675         \HyField@AddToFields
2676         \box_use:N \l_tmpa_box
2677     }
2678     {
2679         \msg_error:nn { hyp }{ pdfa-no-reset-button }
2680         \MakeButtonField{#2}
2681     }
2682 \group_end:
2683 }
2684
2685 \def\@CheckBox[#1]#2
2686 {% parameters, label
2687     \def\Fld@name{#2}
2688     \def\Fld@default{0}
2689     \group_begin:
2690         \def\Fld@width {\DefaultWidthofCheckBox}
2691         \def\Fld@height{\DefaultHeightofCheckBox}
2692         \exp_args:No\HyField@SetKeys
2693         {
2694             \DefaultOptionsofCheckBox,#1
2695         }
2696         \PDFForm@Name
2697         \HyField@FlagsCheckBox
2698         \legacy_if:nT { Fld@hidden }
2699         {
2700             \def\Fld@width{1sp}
2701         }
2702         \LayoutCheckField{#2}
2703         {
2704             \mode_leave_vertical:
2705             \HyAnn@AbsPageLabel
2706             \Hy@escapeform\PDFForm@Check
2707             \pdfannot_box:nnnn
2708             {\Fld@width}
2709             {\Fld@height}
2710             {Opt} %is this correct?
2711             {\PDFForm@Check}
2712             \HyField@AddToFields %check if this works with xelatex ...
2713         }
2714     \group_end:
2715 }
2716 \ExplSyntaxOff
2717
2718 %hm. Should a luatex driver use type1 fonts in fields????
2719 \ExplSyntaxOn

```

```

2720 \def\Hy@FormObjects
2721 {
2722   \pdf_object_new:nn {l__hyp_encoding_pdfdoc_obj } { dict }
2723   \pdf_object_new:nn {l__hyp_font_zapfdingbats_obj } { dict }
2724   \pdf_object_new:nn {l__hyp_font_helvetica_obj } { dict }
2725   \pdf_object_write:nx {l__hyp_encoding_pdfdoc_obj }
2726   {
2727     /Type/Encoding
2728     /Differences[
2729       24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2730       /ring/tilde
2731       \c_space_tl
2732       39/quotesingle
2733       \c_space_tl
2734       96/grave %
2735       \iow_newline:
2736       128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2737       /fraction/guilsinglleft/guilsinglright/minus/perthousand
2738       /quotedblbase/quotedblleft/quotedblright/quoteleft
2739       /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2740       /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2741       \iow_newline:
2742       164/currency
2743       \c_space_tl
2744       166/brokenbar
2745       \c_space_tl
2746       168/dieresis/copyright/ordfeminine
2747       \c_space_tl
2748       172/logicalnot/.notdef/registered/macron/degree/plusminus
2749       /twosuperior/threesuperior/acute/mu
2750       \c_space_tl
2751       183/periodcentered/cedilla/onesuperior/ordmasculine
2752       \c_space_tl
2753       188/onequarter/onehalf/threequarters
2754       \iow_newline:
2755       192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
2756       /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
2757       /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
2758       /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
2759       /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
2760       /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
2761       /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
2762       /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
2763       /ocircumflex/otilde/odieresis/divide/oshlash/ugrave
2764       /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
2765     ]
2766   }
2767   \pdf_object_write:nn {l__hyp_font_zapfdingbats_obj }
2768   {
2769     /Type/Font
2770     /Subtype/Type1
2771     /Name/ZaDb
2772     /BaseFont/ZapfDingbats
2773   }

```

```

2774 \pdf_object_write:nx {l__hyp_font_helvetica_obj }
2775 {
2776     /Type/Font
2777     /Subtype/Type1
2778     /Name/Helv
2779     /BaseFont/Helvetica
2780     /Encoding~\pdf_object_ref:n { l__hyp_encoding_pdfdoc_obj }
2781 }
2782 \global\let\Hy@FormObjects\relax
2783 }
2784 \ExplSyntaxOff
2785 \providecommand*{\Fld@pageobjref}{}
2786 \ifcsname pdf@escapestring\endcsname
2787     \def\Hy@escapeform#1{%
2788         \ifHy@pdfescapeform
2789             \let\Hy@escapestring\pdfescapestring
2790         \else
2791             \let\Hy@escapestring\@firstofone
2792         \fi
2793     }%
2794     \Hy@escapeform{}%
2795 \else
2796     \let\Hy@escapestring\@firstofone
2797     \def\Hy@escapeform#1{%
2798         \ifHy@pdfescapeform
2799             \def\Hy@escapestring##1{%
2800                 \noexpand\Hy@escapestring{\noexpand##1}%
2801             }%
2802             \edef\Hy@temp{#1}%
2803             \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{}\@nil
2804             \def\Hy@escapestring##1{%
2805                 \@ifundefined{Hy@esc@\string##1}{%
2806                     ##1%
2807                     \ThisShouldNotHappen
2808                 }{%
2809                     \csname Hy@esc@\string##1\endcsname
2810                 }%
2811             }%
2812         \else
2813             \let\Hy@escapestring\@firstofone
2814         \fi
2815     }%
2816     \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
2817         \ifx\\#3\\%
2818         \else
2819             \expandafter
2820             \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
2821             \ltx@ReturnAfterFi{%
2822                 \Hy__hypescapeform#3\@nil
2823             }%
2824         \fi
2825     }%
2826 \fi
2827 \def\PDFForm@Name{%

```

```

2828 \PDFForm__hypName\Fld@name
2829 \ifx\Fld@altname\relax
2830 \else
2831 \PDFForm__hypName\Fld@altname
2832 \fi
2833 \ifx\Fld@mappingname\relax
2834 \else
2835 \PDFForm__hypName\Fld@mappingname
2836 \fi
2837 }
2838 \def\PDFForm__hypName#1{%
2839 \begingroup
2840 \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
2841 \ifHy@unicode
2842 \Hy@unicodedefalse
2843 \fi
2844 \fi
2845 \pdfstringdef\Hy@gtemp#1%
2846 \endgroup
2847 \let#1\Hy@gtemp
2848 }
2849 \def\Fld@X@additionalactions{%
2850 \ifx\Fld@keystroke@code\@empty
2851 \else
2852 /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
2853 \fi
2854 \ifx\Fld@format@code\@empty
2855 \else
2856 /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
2857 \fi
2858 \ifx\Fld@validate@code\@empty
2859 \else
2860 /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
2861 \fi
2862 \ifx\Fld@calculate@code\@empty
2863 \else
2864 /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
2865 \fi
2866 \ifx\Fld@onfocus@code\@empty
2867 \else
2868 /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
2869 \fi
2870 \ifx\Fld@onblur@code\@empty
2871 \else
2872 /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
2873 \fi
2874 \ifx\Fld@onmousedown@code\@empty
2875 \else
2876 /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
2877 \fi
2878 \ifx\Fld@onmouseup@code\@empty
2879 \else
2880 /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
2881 \fi

```

```

2882 \ifx\Fld@onenter@code\@empty
2883 \else
2884   /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
2885 \fi
2886 \ifx\Fld@onexit@code\@empty
2887 \else
2888   /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
2889 \fi
2890 }
2891 \ExplSyntaxOn
2892 \def\Fld@additionalactions
2893 {%
2894   \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{}
2895   {
2896     \pdfmeta_standard_verify:nT {annot_widget_no_AA}
2897     {/AA<<\Fld@X@additionalactions>>}
2898   }
2899 }
2900 \ExplSyntaxOff
2901 \def\Fld@annotnames{%
2902   /T(\Fld@name)%
2903   \ifx\Fld@altname\relax
2904   \else
2905     /TU(\Fld@altname)%
2906   \fi
2907   \ifx\Fld@mappingname\relax
2908   \else
2909     /TM(\Fld@mappingname)%
2910   \fi
2911 }
2912 \ExplSyntaxOn
2913 \def\PDFForm@Check
2914 {
2915   /Subtype/Widget
2916   ~\Fld@annotflags
2917   ~\Fld@pageobjref
2918   ~\Fld@annotnames
2919   /FT/Btn
2920   \Fld@flags
2921   /Q~\Fld@align
2922   /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
2923   /AP
2924   <<
2925   /N
2926   <<
2927     /Yes~\pdfxform_ref:o{\l__hyp_CheckmarkYes_tl}
2928     /Off~\pdfxform_ref:o{\l__hyp_CheckmarkOff_tl}
2929   >>
2930   >>
2931   /MK<<
2932     \int_compare:nNnF {\Fld@rotation}={0}
2933     {
2934       /R~\Fld@rotation
2935     }

```

```

2936 \tl_if_empty:NF\Fld@bordercolor
2937 {
2938   /BC[\Fld@bordercolor]
2939 }
2940 \tl_if_empty:NF\Fld@bcolor
2941 {
2942   /BG[\Fld@bcolor]
2943 }
2944 /CA(\Hy@escapestring{\Fld@cbsymbol})%
2945 >>
2946 /DA
2947 (
2948   /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
2949   \tl_if_empty:NF \Fld@color
2950   {
2951     \c_space_tl \Fld@color
2952   }
2953 )
2954 /H/P
2955 \legacy_if:nTF {Fld@checked}
2956 {
2957   /V/Yes /AS/Yes
2958 }
2959 {
2960   /V/Off /AS/Off
2961 }
2962 \Fld@additionalactions
2963 }
2964 \ExplSyntaxOff
2965 \ExplSyntaxOn
2966 \def\PDFForm@Push
2967 {
2968   /Subtype/Widget
2969   ~\Fld@annotflags
2970   ~\Fld@pageobjref
2971   ~\Fld@annotnames
2972   /FT/Btn
2973   ~\Fld@flags
2974   /H/P
2975   /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
2976   \bool_if:nT
2977   {
2978     !\int_compare_p:nNn {\Fld@rotation} = {0}
2979     ||
2980     \tl_if_exist_p:N \Fld@bordercolor
2981   }
2982   {
2983     /MK
2984     <<
2985     \int_compare:nNnF {\Fld@rotation} = {0}
2986     {
2987       /R~\Fld@rotation
2988     }
2989     \tl_if_exist:NT \Fld@bordercolor

```

```

2990         {
2991             /BC[\Fld@bordercolor]
2992         }
2993     >>
2994 }
2995 /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
2996 \Fld@additionalactions
2997 }
2998
2999 \ExplSyntaxOff
3000 \def\PDFForm@List{%
3001     /Subtype/Widget%
3002     \Fld@annotflags
3003     \Fld@pageobjref
3004     \Fld@annotnames
3005     /FT/Ch%
3006     \Fld@flags
3007     /Q \Fld@align
3008     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3009     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3010         \ifx\Fld@bordercolor\relax\else 1\fi
3011         \ifx\Fld@bcolor\relax \else 1\fi
3012         \space
3013 \else
3014     /MK<<%
3015         \ifnum\Fld@rotation=\z@
3016         \else
3017             /R \Fld@rotation
3018         \fi
3019         \ifx\Fld@bordercolor\relax
3020         \else
3021             /BC[\Fld@bordercolor]%
3022         \fi
3023         \ifx\Fld@bcolor\relax
3024         \else
3025             /BG[\Fld@bcolor]%
3026         \fi
3027     >>%
3028 \fi
3029 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3030     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3031 \Fld@choices
3032 \Fld@additionalactions
3033 }
3034 \ExplSyntaxOn
3035 \def\PDFForm@Radio
3036 {
3037     /Subtype/Widget
3038     ~\Fld@annotflags
3039     ~\Fld@pageobjref
3040     ~\Fld@annotnames
3041     /FT/Btn
3042     \Fld@flags
3043     /H/P

```

```

3044 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3045 /MK<<
3046 \ifnum\Fld@rotation=\z@
3047 \else
3048 /R~\Fld@rotation
3049 \fi
3050 \ifx\Fld@bordercolor\relax
3051 \else
3052 /BC[\Fld@bordercolor]%
3053 \fi
3054 \ifx\Fld@bcolor\relax
3055 \else
3056 /BG[\Fld@bcolor]%
3057 \fi
3058 /CA(\Hy@escapestring{\Fld@radiosymbol})%
3059 >>
3060 /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3061 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3062 \ifx\Fld@default\@empty
3063 /V/Off%
3064 /DV/Off%
3065 \else
3066 /V/\Fld@default
3067 /DV/\Fld@default
3068 \fi
3069 \Fld@additionalactions
3070 }
3071 \ExplSyntaxOff
3072 \ExplSyntaxOn
3073 % Does an appeareance dict make sense here?
3074 \def\PDFForm@Text
3075 {
3076 /Subtype/Widget
3077 ~\Fld@annotflags
3078 ~\Fld@pageobjref
3079 ~\Fld@annotnames
3080 /FT/Tx
3081 ~\Fld@flags
3082 /Q~\Fld@align
3083 /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3084 \bool_if:nT
3085 {
3086 !\int_compare_p:nNn {\Fld@rotation} = {0}
3087 ||
3088 \tl_if_exist_p:N \Fld@bordercolor
3089 ||
3090 \tl_if_exist_p:N \Fld@bcolor
3091 }
3092 {
3093 /MK
3094 <<
3095 \int_compare:nNnF {\Fld@rotation} = {0}
3096 {
3097 /R~\Fld@rotation

```

```

3098     }
3099     \tl_if_exist:NT \Fld@bordercolor
3100     {
3101         /BC[\Fld@bordercolor]
3102     }
3103     \tl_if_exist:NT \Fld@bcolor
3104     {
3105         /BG[\Fld@bcolor]
3106     }
3107     >>
3108 }
3109 /DA
3110 (
3111     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3112     \tl_if_empty:NF {\c_space_tl\Fld@color}
3113 )
3114 /DV(\Hy@escapestring{\Fld@default})
3115 /V(\Hy@escapestring{\Fld@value})
3116 ~\Fld@additionalactions
3117 \int_compare:nNnT { \Fld@maxlen}>{0}
3118 {
3119     /MaxLen~\Fld@maxlen
3120 }
3121 }
3122 \ExplSyntaxOff
3123
3124 \def\PDFForm@Submit{%
3125     /Subtype/Widget%
3126     \Fld@annotflags
3127     \Fld@pageobjref
3128     \Fld@annotnames
3129     /FT/Btn%
3130     \Fld@flags
3131     /H/P%
3132     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3133     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3134         \ifx\Fld@bordercolor\relax\else 1\fi
3135         \space
3136     \else
3137         /MK<<%
3138         \ifnum\Fld@rotation=\z@
3139         \else
3140             /R \Fld@rotation
3141         \fi
3142         \ifx\Fld@bordercolor\relax
3143         \else
3144             /BC[\Fld@bordercolor]%
3145         \fi
3146     >>%
3147     \fi
3148     /A<<%
3149     /S/SubmitForm%
3150     /F<<%
3151     /FS/URL%

```

```

3152         /F(\Hy@escapestring{\Form@action})%
3153     >>%
3154     \Fld@submitflags
3155     >>%
3156     \Fld@additionalactions
3157 }
3158 \ExplSyntaxOn
3159 \def\PDFForm@Reset{%
3160     /Subtype/Widget%
3161     \Fld@annotflags
3162     \Fld@pageobjref
3163     \Fld@annotnames
3164     /FT/Btn%
3165     \Fld@flags
3166     /H/P%
3167     /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3168     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3169         \ifx\Fld@bordercolor\relax\else 1\fi
3170         \space
3171     \else
3172         /MK<<%
3173         \ifnum\Fld@rotation=\z@
3174         \else
3175             /R~\Fld@rotation
3176         \fi
3177         \ifx\Fld@bordercolor\relax
3178         \else
3179             /BC[\Fld@bordercolor]%
3180         \fi
3181         >>%
3182     \fi
3183     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3184     /A<</S/ResetForm>>%
3185     \Fld@additionalactions
3186 }%
3187
3188
3189 %these patterns are used in hyperref checks.
3190 %it is unclear if they are really useful and if a backend support is
3191 %needed.
3192 \str_case:VnF \c_sys_backend_str
3193 {
3194     { pdfmode }
3195     {
3196         \def\HyPat@ObjRef
3197         {
3198             [0-9]*[1-9][0-9]*~0~R
3199         }
3200     }
3201     { dvipdfmx }
3202     {
3203         \def\HyPat@ObjRef
3204         {
3205             @[\~]+

```

```

3206     }
3207   }
3208   { xdvipdfmx }
3209   {
3210     \def\HyPat@ObjRef
3211     {
3212       @[\~]+
3213     }
3214   }
3215 }
3216 { %also set in hyperref sty, so probably not needed.
3217   \def\HyPat@ObjRef/{.+}
3218 }
3219
3220
3221 \ExplSyntaxOff
3222 % UF: removed Hy@writebookmark
3223 %   \Hy@currentbookmarklevel{0}
3224 %   \Hy@numberline
3225 %   \__hypwritetorep
3226 %   counter{bookmark@seq@number}
3227 % removed \HyPsd@SanitizeForOutFile, not needed
3228 % removed \currentpdfbookmark, defined by bookmark,
3229 % should use \newcommand there
3230 % removed \subpdfbookmark, defined by bookmark,
3231 % should use \newcommand there
3232 % removed \belowpdfbookmark, defined by bookmark,
3233 % should use \newcommand there
3234 % removed \pdfbookmark, defined by bookmark,
3235 % \BOOKMARK
3236 % \@BOOKMARK
3237 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3238 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3239 %% removed \ReadBookmarks / unneeded with bookmark.
3240 %% removed \Hy@OutlineName
3241 %% removed \check@bm@number
3242 %% removed \calc@bm@number
3243
3244 \ifHy@implicit
3245 \else
3246   \expandafter\endinput
3247 \fi
3248 \newlength\Hy@SectionHShift
3249 \def\Hy@SectionAnchorHref#1{%
3250   \ifx\protect\@typeset@protect
3251     \Hy__hypSectionAnchor{#1}%
3252   \fi
3253 }
3254 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%
3255   \leavevmode
3256   \hbox to 0pt{%
3257     \kern-\Hy@SectionHShift
3258     \Hy@raisedlink{%
3259       \hyper@anchorstart{#1}\hyper@anchorend

```

```

3260     }%
3261     \hss
3262 }%
3263 }
3264 \let\H@old@ssect\@ssect
3265 \def\@ssect#1#2#3#4#5{%
3266     \Hy@MakeCurrentHrefAuto{section*}%
3267     \setlength{\Hy@SectionHShift}{#1}%
3268     \begingroup
3269         \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3270         \toks\tw@%
3271         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3272         #5%
3273     }%
3274     \edef\x{\endgroup
3275         \the\toks@{\the\toks\tw@}%
3276     }\x
3277 }
3278 \let\H@old@schapter\@schapter
3279 \def\@schapter#1{%
3280     \begingroup
3281         \let\@mkboth\@gobbletwo
3282         \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3283         \Hy@raisedlink{%
3284             \hyper@anchorstart{\@currentHref}\hyper@anchorend
3285         }%
3286     \endgroup
3287     \H@old@schapter{#1}%
3288 }
3289 \ltx@ifundefined{chapter}{}{%
3290     \let\Hy@org@chapter\@chapter
3291     \def\@chapter{%
3292         \def\Hy@next{%
3293             \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3294             \Hy@raisedlink{%
3295                 \hyper@anchorstart{\@currentHref}\hyper@anchorend
3296             }%
3297         }%
3298         \ifnum\c@secnumdepth>\m@ne
3299             \ltx@ifundefined{if@mainmatter}%
3300             \iftrue{\csname if@mainmatter\endcsname}%
3301             \let\Hy@next\relax
3302         \fi
3303         \fi
3304         \Hy@next
3305         \Hy@org@chapter
3306     }%
3307 }
3308 \let\H@old@part\@part
3309 \begingroup\expandafter\expandafter\expandafter\endgroup
3310 \expandafter\ifx\csname chapter\endcsname\relax
3311     \let\Hy@secnum@part\z@
3312 \else
3313     \let\Hy@secnum@part\m@ne

```

```

3314 \fi
3315 \def\@part{%
3316   \ifnum\Hy@secnum@part>\c@secnumdepth
3317     \phantomsection
3318   \fi
3319   \H@old@part
3320 }
3321 \let\H@old@spart\@spart
3322 \def\@spart#1{%
3323   \Hy@MakeCurrentHrefAuto{part*}%
3324   \Hy@raisedlink{%
3325     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3326   }%
3327   \H@old@spart{#1}%
3328 }
3329 \let\H@old@sect\@sect
3330 \def\@sect#1#2#3#4#5#6[#7]#8{%
3331   \ifnum #2>\c@secnumdepth
3332     \expandafter\@firstoftwo
3333   \else
3334     \expandafter\@secondoftwo
3335   \fi
3336   {%
3337     \Hy@MakeCurrentHrefAuto{section*}%
3338     \setlength{\Hy@SectionHShift}{#3}%
3339     \begingroup
3340     \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3341     \toks\tw@\expandafter{%
3342       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3343       #8%
3344     }%
3345     \edef\x{\endgroup
3346       \the\toks@{\the\toks\tw@}%
3347     }\x
3348   }{%
3349     \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3350   }%
3351 }
3352 \expandafter\def\csname Parent-4\endcsname{}
3353 \expandafter\def\csname Parent-3\endcsname{}
3354 \expandafter\def\csname Parent-2\endcsname{}
3355 \expandafter\def\csname Parent-1\endcsname{}
3356 \expandafter\def\csname Parent0\endcsname{}
3357 \expandafter\def\csname Parent1\endcsname{}
3358 \expandafter\def\csname Parent2\endcsname{}
3359 \expandafter\def\csname Parent3\endcsname{}
3360 \expandafter\def\csname Parent4\endcsname{}
3361 %%
3362 %% End of file 'hgeneric-testphase.def'.
3363 \</package>
3364 \<colorscheme>
3365 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3366 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3367 % linkcolor=

```

```

3368 %,filecolor=
3369 %,urlcolor=
3370 %,menucolor=
3371 %,runcolor=
3372 %,linkbordercolor=
3373 %,filebordercolor=
3374 %,urlbordercolor=
3375 %,menubordercolor=
3376 %,runbordercolor=
3377
3378 \prop_const_from_keyval:Nn \c__hyp_colorscheme_original_prop
3379 {
3380     linkcolor      = [rgb]{1,0,0}, %red
3381     filecolor      = [rgb]{0,1,1}, %cyan
3382     urlcolor       = [rgb]{1,0,1}, %magenta
3383     menucolor      = [rgb]{1, 0, 0}, %red
3384     runcolor       = [rgb]{0,1,1}, %cyan
3385     %-----
3386     linkbordercolor = [rgb]{1, 0 ,0 },
3387     filebordercolor = [rgb]{0, .5, .5},
3388     urlbordercolor  = [rgb]{0, 1, 1},
3389     menubordercolor = [rgb]{1, 0, 0},
3390     runbordercolor  = [rgb]{0, .7, .7}
3391 }
3392
3393 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3394 {
3395     linkcolor      = [rgb]{0,0.2,0.6},
3396     filecolor      = [rgb]{0.8,0,0.8},
3397     urlcolor       = [rgb]{0.8,0,0.8},
3398     menucolor      = [rgb]{0,0.2,0.6},
3399     runcolor       = [rgb]{0.8,0,0.8},
3400     %----- %-----
3401     linkbordercolor = [rgb]{0,0.2,0.6},
3402     filebordercolor = [rgb]{0.8,0,0.8},
3403     urlbordercolor  = [rgb]{0.8,0,0.8},
3404     menubordercolor = [rgb]{0,0.2,0.6},
3405     runbordercolor  = [rgb]{0.8,0,0.8}
3406 }
3407
3408 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3409 { %two colors: intern/extern
3410     linkcolor      = [rgb]{0.79216, 0, 0.12549},
3411     filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3412     urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3413     menucolor      = [rgb]{0.79216, 0, 0.12549 },
3414     runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3415     %----- %-----
3416     linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3417     filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3418     urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3419     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3420     runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3421 }

```

```

3422
3423 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3424 { %all darkgray
3425   linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3426   filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3427   urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3428   menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3429   runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3430   %----- %-----
3431   linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3432   filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3433   urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3434   menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3435   runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3436 }
3437
3438 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3439 { %dvipsnam.def
3440   linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3441   filecolor      = [rgb]{1, 0, 0}, %Red
3442   urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3443   menucolor      = [rgb]{1, 0, 0}, %Red
3444   runcolor       = [rgb]{1, 0, 0}, %Red
3445   %----- %-----
3446   linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3447   filebordercolor = [rgb]{1, 0, 0}, %Red
3448   urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3449   menubordercolor = [rgb]{1, 0, 0}, %Red
3450   runbordercolor  = [rgb]{1, 0, 0} %Red
3451 }
3452
3453 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3454 { %dvipsnam.def
3455   linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3456   filecolor      = [rgb]{0, 1, 0}, %Green
3457   urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3458   menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3459   runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3460   %----- %-----
3461   linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3462   filebordercolor = [rgb]{0, 1, 0}, %Green
3463   urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3464   menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3465   runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3466 }
3467
3468
3469 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3470 {
3471   linkcolor      = [rgb]{0.50196, 0, 0.02353},
3472   filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3473   urlcolor       = [rgb]{0.54118, 0, 0.52941},
3474   menucolor      = [rgb]{0.44706, 0.45882, 0},
3475   runcolor       = [rgb]{0.07451, 0.46667, 0.46275},

```

```

3476 %----- %-----
3477 linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3478 filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3479 urlbordercolor = [rgb]{0.724708, 0.4, 0.717646},
3480 menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3481 runbordercolor = [rgb]{0.444706, 0.680002, 0.67765}
3482 }
3483
3484 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3485 {
3486 linkcolor = [rgb]{0,0,0},
3487 filecolor = [rgb]{0,0,0},
3488 urlcolor = [rgb]{0,0,0},
3489 menucolor = [rgb]{0,0,0},
3490 runcolor = [rgb]{0,0,0},
3491 %----- %-----
3492 linkbordercolor = [rgb]{0,0,0},
3493 filebordercolor = [rgb]{0,0,0},
3494 urlbordercolor = [rgb]{0,0,0},
3495 menubordercolor = [rgb]{0,0,0},
3496 runbordercolor = [rgb]{0,0,0}
3497 }
3498 </colorscheme>

```

Index

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

Symbols	C
\# 490	\char 2332, 2344, 2356, 2368
\% 491	\chardef 136
\. 305, 309, 311	\cite 25
@curropt commands:	clist commands:
\@curropt: 2460, 2528	\clist_map_function:nN 111, 120
\\ 18, 19, 20,	\clist_map_inline:nn 2033
27, 31, 41, 61, 68, 75, 82, 89, 104,	\clist_map_variable:nNn .. 2459, 2527
109, 110, 118, 119, 128, 129, 130,	color commands:
131, 132, 138, 616, 2462, 2530, 2817	\color_export:nnN 669
_ 303, 305, 309, 311	\color_export:nnnN 675
	\color_select:n 686, 722, 990
A	\color_select:nn 692
\A 303	\color_set:nn 2, 184, 703
\Acrobatmenu 15, 155	\color_set:nnn 2, 183, 709
\addcontentsline 10	color names:
\AddToDocumentProperties 179	hyp/annot/file 286
\AddToHook 204	hyp/annot/link 286
\advance 2393, 2467, 2511, 2512	hyp/annot/menu 286
allcolor (hypersetup key) 749	hyp/annot/run 286
\author 2	hyp/annot/url 286
	colorfile (hypersetup key) 749
B	colorlink (hypersetup key) 749
\b 307	colorlinks (hypersetup key) 735
\begingroup ... 2236, 2427, 2456, 2468,	colormenu (hypersetup key) 749
2502, 2839, 3268, 3280, 3309, 3339	colorrun (hypersetup key) 749
\belowpdfbookmark 3232	colorsche (hypersetup key) 1, 1153
\BOOKMARK 3235	colorurl (hypersetup key) 749
bookmarkstype (hypersetup key) 10	cs commands:
bool commands:	\cs_generate_variant:Nn 135,
\bool_if:NTF 436, 450, 461,	181, 217, 232, 237, 341, 678, 695, 712
474, 517, 567, 627, 719, 729, 969, 980	\cs_gset:Npn 961
\bool_if:nTF 579, 1672, 1697, 2976, 3084	\cs_if_exist_p:N 200
\bool_lazy_and:nnTF 199	\cs_new:Npn 104, 233
\bool_new:N 289, 293, 297	\cs_new_protected:Npn .. 176, 211,
\bool_set_true:N 298	220, 317, 321, 325, 329, 342, 344,
bordercolormodel (hypersetup key) 10, 770	348, 390, 396, 403, 407, 429, 434,
box commands:	448, 459, 472, 515, 565, 625, 662,
\box_dp:N 379, 2599, 2638, 2673	673, 679, 690, 696, 707, 905, 1904, 1960
\box_ht:N 378, 2598, 2637, 2672	\cs_set_eq:NN 139, 148, 149, 490, 491, 1230
\box_new:N 238, 300	\cs_set_protected:Npn 137, 162, 358, 616
\box_set_dp:Nn 1869	\csname 2474, 2809,
\box_set_ht:Nn 1876	2820, 3300, 3310, 3352, 3353, 3354,
\box_use:N 986, 2601, 2647, 2676	3355, 3356, 3357, 3358, 3359, 3360
\box_use_drop:N 991	\currentpdfbookmark 3228
\box_wd:N 377, 2597, 2636, 2671	
\l_tmpa_box 2595, 2597, 2598,	
2599, 2601, 2634, 2636, 2637, 2638,	
2647, 2669, 2671, 2672, 2673, 2676	

D	
<code>\d</code>	305, 309, 311
<code>debug</code> (hypersetup key)	<u>1177</u>
<code>\DeclareDocumentMetadata</code>	
.	3, 8, <u>11</u> , <u>12</u> , <u>15</u> , 20, 28, 34
<code>\DeclareRobustCommand</code>	3254
<code>\def</code>	143,
353, 2199, 2203, 2210, 2228, 2257,	
2274, 2391, 2400, 2415, 2416, 2419,	
2420, 2433, 2434, 2450, 2451, 2454,	
2455, 2493, 2523, 2561, 2575, 2576,	
2588, 2616, 2618, 2619, 2629, 2652,	
2654, 2655, 2666, 2685, 2687, 2688,	
2690, 2691, 2700, 2720, 2787, 2797,	
2799, 2804, 2816, 2827, 2838, 2849,	
2892, 2901, 2913, 2966, 3000, 3035,	
3074, 3124, 3159, 3196, 3203, 3210,	
3217, 3249, 3265, 3279, 3291, 3292,	
3315, 3322, 3330, 3352, 3353, 3354,	
3355, 3356, 3357, 3358, 3359, 3360	
<code>\DefaultHeightofCheckBox</code>	2691
<code>\DefaultHeightofChoiceMenu</code>	2455
<code>\DefaultHeightofReset</code>	2655
<code>\DefaultHeightofSubmit</code>	2619
<code>\DefaultHeightofText</code>	2424
<code>\DefaultHeightofTextMultiline</code>	2422
<code>\DefaultOptionsofCheckBox</code>	2694
<code>\DefaultOptionsofPushButton</code>	2580
<code>\DefaultOptionsofReset</code>	2659
<code>\DefaultOptionsofSubmit</code>	2623
<code>\DefaultOptionsofText</code>	2429
<code>\DefaultWidthofCheckBox</code>	2690
<code>\DefaultWidthofChoiceMenu</code>	2454
<code>\DefaultWidthofReset</code>	2654
<code>\DefaultWidthofSubmit</code>	2618
<code>\DefaultWidthofText</code>	2419
<code>destlabel</code> (hypersetup key)	<u>10</u>
dim commands:	
<code>\dim_to_decimal_in_bp:n</code>	9, 139
<code>\do</code>	2460, 2528
<code>\documentclass</code>	21
<code>draft</code> (hypersetup key)	<u>1177</u>
E	
<code>\edef</code> 2238, 2470, 2506, 2513, 2802, 3274, 3345	
<code>\EdefEscapeName</code>	2526, 2531
<code>\else</code>	2235, 2423, 2477,
2481, 2484, 2501, 2509, 2790, 2795,	
2812, 2818, 2830, 2834, 2851, 2855,	
2859, 2863, 2867, 2871, 2875, 2879,	
2883, 2887, 2904, 2908, 3009, 3010,	
3011, 3013, 3016, 3020, 3024, 3030,	
3047, 3051, 3055, 3061, 3065, 3133,	
3134, 3136, 3139, 3143, 3168, 3169,	
3171, 3174, 3178, 3245, 3312, 3333	
<code>\endcsname</code>	2488, 2786, 2809,
2820, 3300, 3310, 3352, 3353, 3354,	
3355, 3356, 3357, 3358, 3359, 3360	
<code>\endgroup</code>	2239, 2447, 2470, 2516,
2519, 2846, 3274, 3286, 3309, 3345	
<code>\endinput</code>	3246
exp commands:	
<code>\exp_args:Ne</code>	2894
<code>\exp_args:Nne</code>	1998
<code>\exp_args:Nno</code>	1946
<code>\exp_args:NV</code>	1657, 1688
<code>\exp_args:NnV</code>	392, 399
<code>\exp_args:Nnx</code>	2181
<code>\exp_args:No</code>	
.	421, 425, 2578, 2621, 2657, 2692
<code>\exp_args:NV</code>	375, 382
<code>\exp_args:Nx</code>	1939, 1987, 2230
<code>\exp_not:N</code>	616
<code>\exp_not:n</code>	
.	1092, 1126, 1237, 1297, 1318, 1355,
.	1375, 1398, 1442, 1475, 1500, 1551,
.	1576, 1603, 1652, 1729, 1757, 1784,
.	1804, 2019, 2126, 2154, 2164, 2175
<code>\expandafter</code>	2428, 2462,
2471, 2473, 2474, 2530, 2803, 2819,	
3246, 3270, 3271, 3309, 3310, 3332,	
3334, 3341, 3342, 3352, 3353, 3354,	
3355, 3356, 3357, 3358, 3359, 3360	
<code>\ExplSyntaxOff</code>	
.	10, 2267, 2383, 2410, 2716, 2784,
.	2900, 2964, 2999, 3071, 3122, 3221
<code>\ExplSyntaxOn</code>	
.	8, 11, 2268, 2388, 2411, 2719,
.	2891, 2912, 2965, 3034, 3072, 3158
<code>extension</code> (hypersetup key)	<u>11</u> , <u>1199</u>
F	
<code>\fbox</code>	2374, 2379, 2552
<code>\fi</code>	484, 2224,
2251, 2254, 2263, 2425, 2433, 2434,	
2465, 2483, 2486, 2487, 2493, 2496,	
2505, 2507, 2514, 2517, 2792, 2814,	
2824, 2826, 2832, 2836, 2843, 2844,	
2853, 2857, 2861, 2865, 2869, 2873,	
2877, 2881, 2885, 2889, 2906, 2910,	
3009, 3010, 3011, 3018, 3022, 3026,	
3028, 3030, 3049, 3053, 3057, 3061,	
3068, 3133, 3134, 3141, 3145, 3147,	
3168, 3169, 3176, 3180, 3182, 3247,	
3252, 3302, 3303, 3314, 3318, 3335	
<code>file</code> (hypersetup key)	8, <u>1245</u>

file commands:	
\file_input:n	9
fileborderstyle (hypersetup key)	11, 859
filecolor (hypersetup key)	749
final (hypersetup key)	1177
\fontencoding	2328, 2340, 2352, 2364
\fontfamily	2327, 2339, 2351, 2363
\fontseries	2329, 2341, 2353, 2365
\fontshape	2330, 2342, 2354, 2366
fp commands:	
\fp_eval:n	1827, 1855, 1871
G	
\GetDocumentProperties	2
\global	2393, 2782
group commands:	
\group_begin:	
. 331, 439, 453, 476, 519, 569, 629,	
721, 972, 989, 2105, 2326, 2338,	
2350, 2362, 2577, 2620, 2656, 2689	
\group_end:	
. 338, 443, 464, 506, 557, 610, 657,	
731, 992, 995, 2112, 2333, 2345,	
2357, 2369, 2613, 2649, 2682, 2714	
H	
\hbox	3256
hbox commands:	
\hbox_overlap_right:n	986
\hbox_set:Nn	2595, 2634, 2669
\hbox_set:Nw	973
\hbox_set_end:	982
\hbox_set_to_wd:Nnn	1853
hidefile (hypersetup key)	1129
hidelink (hypersetup key)	1129
hidelinks (hypersetup key)	1129
hidemenu (hypersetup key)	1129
hiderun (hypersetup key)	1129
hideurl (hypersetup key)	1129
hook commands:	
\hook_gput_code:nnn	
. 191, 715, 725, 965, 976	
\hook_new:n	182, 316
\hook_use:n	333, 440, 454
\hss	3261
Hy internal commands:	
\Hy_hyescapeform	2803, 2816, 2822
\Hy_hypSectionAnchor	3251, 3254
HyField internal commands:	
\HyField_hypAddToFields	2230, 2257
hyp commands:	
\l_hyp_annot_colorfile_bool	287
\l_hyp_annot_colorlink_bool	287
\l_hyp_annot_colormenu_bool	287
\l_hyp_annot_colorrund_bool	287
\l_hyp_annot_colorurl_bool	287
\l_hyp_annot_ocgcolorfile_bool	291
\l_hyp_annot_ocgcolorlink_bool	291
\l_hyp_annot_ocgcolormenu_bool	291
\l_hyp_annot_ocgcolorrund_bool	291
\l_hyp_annot_ocgcolorurl_bool	291
hyp internal commands:	
\g__hyp_AcroForm_CoFields_prop	
. 2191, 2212, 2286, 2288, 2303	
\g__hyp_AcroForm_Fields_prop	
. 2192, 2205, 2281	
\l__hyp_annot_GoTo_bool	436, 450, 461
\l__hyp_annot_GoTo_bool\l__-	
hyp_annot_URI_bool\l__-	
hyp_annot_GoToR_bool\l__-	
hyp_annot_Named_bool\l__-	
hyp_annot_Launch_bool	295
\l__hyp_annot_GoToR_bool	517
\l__hyp_annot_Launch_bool	567
\l__hyp_annot_Named_bool	627
\c__hyp_annot_types_seq	
. 250, 287, 291, 749, 1044, 1140	
\l__hyp_annot_URI_bool	474
\g__hyp_bordercolormodel_str	
. 286, 774, 790	
\l__hyp_CheckmarkOff_tl	
. 2271, 2272, 2928	
\l__hyp_CheckmarkYes_tl	
. 2269, 2270, 2927	
__hyp_clist_display:n	104, 111, 120
__hyp_color_export:nnN	
. 31, 662, 662, 678, 790	
__hyp_color_select:n	32, 679, 679, 695
__hyp_color_select:nN	32
__hyp_color_set:nN	32
__hyp_color_set:nn	
. 32, 187, 189, 696, 696, 712, 754	
__hyp_colormodel_export:nnnN	31, 662
__hyp_colormodel_export:wnnN	
. 666, 673	
__hyp_colormodel_select_aux:wn	
. 679, 683, 690	
__hyp_colormodel_set_aux:nwn	
. 696, 700, 707	
\c__hyp_colorscheme_daleif_prop	3393
\c__hyp_colorscheme_henryford_	
prop	3484
\c__hyp_colorscheme_julian_prop	3408
\c__hyp_colorscheme_original_	
prop	3378
\c__hyp_colorscheme_phetype_prop	
. 3469	

\c_hyp_colorscheme_szabolcsA-prop	3438	\l_hyp_text_enc_dest_print_tl	272, 529
\c_hyp_colorscheme_szabolcsB-prop	3453	\l_hyp_text_enc_dest_tl	24, 272, 371, 422
\c_hyp_colorscheme_tivv_prop	3423	\l_hyp_text_enc_file_print_tl	272, 524, 572
\l_hyp_dest_box	24, 300, 377, 378, 379, 1853, 1869, 1876	\l_hyp_text_enc_info_print_tl	272, 346
\l_hyp_dest_name_tmpa_tl	243, 410, 411, 417, 421, 423, 426, 530, 538	\l_hyp_text_enc_para_print_tl	272, 577
\l_hyp_dest_pdfremotestartview_tl	269, 535, 1660, 1664	\l_hyp_text_enc_uri_print_tl	272, 479, 1256
\g_hyp_dest_pdfstartpage_tl	269, 1671, 1673, 1680, 1698, 1705	_hyp_text_pdfstring:nnN	329, 329, 341, 343, 346, 369, 420, 477, 522, 527, 570, 575, 1256
\g_hyp_dest_pdfstartview_tl	269, 1673, 1680, 1691, 1695, 1698, 1705	_hyp_text_pdfstring_info:nN	344, 344, 1915, 1933, 1943
\l_hyp_dest_pdfview_tl	285, 393, 400, 1822, 1825, 1832, 1835, 1836, 1837, 1838, 1839, 1840, 1845, 1849, 1885	_hyp_text_purify:nN	317, 317, 334
\c_hyp_dest_startview_regex	301, 1658, 1689	_hyp_text_string_from_unicode:nN	325, 325, 336
\c_hyp_dest_undefined_tl	249, 416, 417	\g_hyp_text_tmpa_str	248, 337, 339
_hyp_destination:nn	24, 364, 364, 393, 400	\l_hyp_text_tmpa_str	247, 334, 335, 336, 337
\l_hyp_filename_tmpa_tl	243, 520, 521, 523, 525, 526, 573, 574, 590	\l_hyp_tmpa_box	238, 973, 986, 991
_hyp_link_goto_begin:nw	407, 441, 455	\l_hyp_tmpa_int	238, 1658, 1660, 1689, 1691, 1812, 1813, 1817, 1819, 1843, 1850, 1851, 1852, 1861, 1863, 1873, 1878, 2290, 2292, 2298
_hyp_link_goto_end:	429, 442, 463	\l_hyp_tmpa_str	238, 1915, 1916, 1918, 1933, 1934, 1936, 1943, 1944, 1948
\c_hyp_map_annot_hyp_prop	250	\l_hyp_tmpa_tl	238, 372, 376, 384, 790, 794, 1256, 1257, 1262, 1656, 1658, 1687, 1689, 1819, 1820, 1827
\c_hyp_map_hyp_annot_prop	250, 713, 777, 813, 840, 848, 859, 886, 894, 963, 1067, 1103, 1113, 1245	\l_hyp_uri_tmpa_tl	243, 480, 481
_hyp_ocg_init:	905, 905, 961, 971	hyp/annot/file (color name)	286
_hyp_PageLabels_gpush:	348, 348, 361	hyp/annot/link (color name)	286
\l_hyp_para_tmpa_tl	246, 578, 581, 590	hyp/annot/menu (color name)	286
\l_hyp_para_tmpa_tl\l_hyp_text_tmpa_str	243	hyp/annot/run (color name)	286
\l_hyp_RadioYes_tl	2521, 2522	hyp/annot/url (color name)	286
_hyp_ref_check:nn	210, 220, 232, 2398	hyp/text/pdfstring	316
_hyp_ref_if_exist:nn	210, 218, 219	\hypercalcbp	9, 15, 139
_hyp_ref_if_exist:nnTF	222, 2402	\HyperDestNameFilter	370, 421
_hyp_ref_label:nn	210, 211, 217, 2397	\hypersetup	1, 2, 6, 8, 10, 15, 59, 91, 162
_hyp_ref_value:nn	210, 233, 237, 2406	\hypersetup keys:	
_hyp_setup_info_date_key:nn	1960, 1992, 1993	allcolor	749
_hyp_setup_info_key:nn	1904, 1953, 1954, 1955, 1956, 1958, 1959	bookmarkstype	10
_hyp_store_metadata:nn	176, 181, 1901, 1921, 1939, 1973, 1987, 2008, 2080	bordercolormodel	10, 770
_hyp_text_cleanup:N	321, 321, 335	colorfile	749
		colorlink	749
		colorlinks	735
		colormenu	749
		colorrun	749

\keys_set:nn
 . 165, 193, 1159, 1166, 1999, 2026,
 2029, 2030, 2031, 2032, 2106, 2181
 \keys_set_known:nn 2184
 \kvsetkeys 164, 2277

L

\label 6, 11
 \LayoutCheckField 2702
 \LayoutChoiceField 2497
 \LayoutPushButtonField 2590, 2607
 \LayoutTextField 2435
 \leavevmode 2436, 2534, 3255
 legacy commands:

 \legacy_if:nTF 150, 202, 206,
 356, 2315, 2586, 2627, 2665, 2698, 2955
 \let 146,
 2194, 2195, 2197, 2198, 2225, 2276,
 2384, 2385, 2386, 2417, 2418, 2452,
 2453, 2495, 2782, 2789, 2791, 2796,
 2813, 2847, 3264, 3278, 3281, 3290,
 3301, 3308, 3311, 3313, 3321, 3329
 link (hypersetup key) 8, [1245](#)
 linkborder (hypersetup key) 11
 linkborderstyle (hypersetup key) . 11, [859](#)
 linkcolor (hypersetup key) [749](#)
 linkfileprefix (hypersetup key) ... [1199](#)
 linktoc (hypersetup key) [1199](#)
 linktocpage (hypersetup key) [1199](#)
 localanchorname (hypersetup key) . [1199](#)

M

\MakeButtonField
 2595, 2610, 2634, 2669, 2680
 \MakeChoiceField 2570
 \MakeFieldObject 2185,
 2324, 2336, 2348, 2360, 2372, 2377
 \MakeRadioField 2552
 \MakeTextField 2444
 \mbox 983
 menu (hypersetup key) 8, [1245](#)
 menuborder (hypersetup key) 11
 menuborderstyle (hypersetup key) . 11, [859](#)
 menucolor (hypersetup key) [749](#)
 mode commands:
 \mode_if_horizontal:TF 366, 388
 \mode_leave_vertical:
 409, 493, 545, 597,
 632, 2564, 2592, 2609, 2631, 2661, 2704
 msg commands:
 \msg_error:nn 201, 2606, 2679
 \msg_line_context: 69
 \g_msg_module_name_prop 12

\msg_new:nnn 37, 44, 49, 53,
 57, 64, 71, 78, 85, 92, 98, 105, 114, 124
 \msg_new:nnnn 13, 23
 \msg_warning:nn 152, 1173
 \msg_warning:nnn 161, 413, 654
 \msg_warning:nnnn
 1005, 1052, 1295, 1339,
 1373, 1396, 1458, 1487, 1521, 1536,
 1549, 1562, 1587, 1620, 1636, 1663,
 1694, 1727, 1741, 1768, 1802, 1884
 \msg_warning:nnnnn 172,
 1089, 1123, 1234, 1315, 1352, 1439,
 1472, 1497, 1573, 1600, 1649, 1754,
 1781, 2016, 2123, 2142, 2161, 2172

N

naturalnames (hypersetup key) [1199](#)
 nesting (hypersetup key) 11
 \newcommand 142, 3229, 3231, 3233
 \newcount 2389, 2560
 \NewDocumentCommand 2185
 \newlength 3248
 \noexpand ... 2471, 2472, 2473, 2474, 2800

O

ocgcolorfile (hypersetup key) [999](#)
 ocgcolorlink (hypersetup key) [999](#)
 ocgcolorlinks (hypersetup key) [999](#)
 ocgcolormenu (hypersetup key) [999](#)
 ocgcolorrun (hypersetup key) [999](#)
 ocgcolorurl (hypersetup key) [999](#)

P

page/Trans internal commands:
 l_hyp_page/Trans 314
 pageanchor (hypersetup key) [1199](#)
 \paperwidth 3
 \PassOptionsToPackage ... 208, 1191, 1196
 pdf commands:
 \pdf_bdcobject:nn 985, 988
 \pdf_destination:nn 19, 135, 383
 \pdf_destination:nnnn 376
 \pdf_emc: 987, 993
 \pdf_link_user:nnn 2567
 \pdf_name_from_unicode_e:n 634
 \pdf_object_new:nn
 . 907, 908, 909, 910, 2722, 2723, 2724
 \pdf_object_ref:n
 913, 915, 937, 938, 941,
 945, 950, 955, 960, 2308, 2310, 2780
 \pdf_object_ref_last: 2111
 \pdf_object_unnamed_write:nn . 2107
 \pdf_object_write:nn
 . 911, 917, 927, 939, 2725, 2767, 2774

\pdf_pageobject_ref:n	1680, 1705, 2404	\pdfmanagement_add:nnn
\pdf_string_from_unicode:nnN	.. 327	350, 937, 938,
\pdf_version:	960, 1262, 1287, 1303, 1307, 1327,
... 1343, 1462, 1491, 1525, 1540,		1335, 1364, 1419, 1429, 1453, 1467,
1566, 1591, 1624, 1640, 1745, 1772		1479, 1484, 1509, 1517, 1532, 1558,
\pdf_version_compare:NnTF	1583, 1615, 1632, 1678, 1703, 1717,
.....	999, 1046,	1737, 1764, 1793, 1899, 1918, 1936,
1333, 1424, 1446, 1482, 1515, 1530,		1947, 1971, 1985, 2003, 2092, 2111,
1556, 1581, 1607, 1630, 1735, 1762		2283, 2300, 2307, 2309, 2311, 2317
\pdf_version_compare_p:Nn 583	\pdfmanagement_if_active_p:	... 200
\pdf_version_major: .. 149, 1008, 1056		\pdfmanagement_remove:nn
\pdf_version_minor: .. 148, 1008, 1056		... 1259, 1283, 1291, 1311, 1323,	
pdfannot commands:		1348, 1360, 1368, 1382, 1391, 1435,	
\pdfannot_box:nnnn	1450, 1469, 1506, 1545, 1570, 1596,	
.. 2439, 2537, 2596, 2635, 2670, 2707		1611, 1645, 1675, 1700, 1713, 1722,	
\pdfannot_box_ref_last: .. 2232, 2248		1750, 1777, 1789, 1798, 1896, 1912,	
\pdfannot_dict_put:nnn	1930, 1968, 1982, 2089, 2102, 2321	
.. 791, 826, 850, 872, 896, 1074, 1105		\pdfmanagement_show:n 2284
\pdfannot_dict_remove:nn	pdfmeta commands:	
.. 785, 821, 842, 867, 888, 1082, 1115		\pdfmeta_standard_verify:nnTF	...
\pdfannot_link:nnn	.. 494, 546, 598, 640	630, 2583, 2662
\pdfannot_link_goto_begin:nw	.. 426	\pdfmeta_standard_verify:nTF	...
\pdfannot_link_goto_end:	431	2313, 2896
\pdfannot_link_margin:n 6, 1377	pdfmoddate (hypersetup key) 11, 1960
\c_pdfannot_link_types_seq 295	pdfproducer (hypersetup key) 1904
pdfauthor (hypersetup key) 1904	pdfremotestartview (hypersetup key)	... 9
\pdfbookmark 3234	pdfstartview (hypersetup key) 9
pdfborder (hypersetup key) 11	\pdfstringdef 6, 7, 15, 21, 2845
pdfborderstyle (hypersetup key)	.. 11, 859	pdfsubject (hypersetup key) 1904
pdfcreationdate (hypersetup key)	11, 1960	pdftitle (hypersetup key) 1904
pdfcreator (hypersetup key) 1904	pdftrapped (hypersetup key) 1994
\pdfdest 9	pdfversion (hypersetup key) 1167
pdfdict commands:		pdfview (hypersetup key) 9, 1808
\pdfdict_new:n	314, 468, 511, 561, 621	pdfxform commands:	
\pdfdict_put:nnn	\pdfxform_new:nnn 2187
.. 315, 469, 470, 481, 483, 488, 512,		\pdfxform_ref:n
513, 526, 534, 538, 543, 562, 563,		.. 2547, 2548, 2642, 2643, 2927, 2928	
574, 589, 595, 622, 623, 633, 638,		\phantom 2344, 2368
1403, 1404, 1408, 1409, 2120, 2130,		\phantomsection 3317
2134, 2137, 2139, 2158, 2167, 2169		plainpages (hypersetup key) 1199
\pdfdict_remove:nn	... 586, 1413, 1414	prg commands:	
\pdfdict_use:n	498, 550, 603, 644, 2109	\prg_generate_conditional_-	
pdfencoding (hypersetup key) 1167	variant:Nnn 219
\pdfescapestring 2789	\prg_new_eq_conditional:NNn	... 218
PDFForm internal commands:		prop commands:	
\PDFForm__hypName	\prop_const_from_keyval:Nn
..... 2828, 2831, 2835, 2838		252, 260, 3378, 3393,
pdfinfo (hypersetup key) 2022	3408, 3423, 3438, 3453, 3469, 3484
pdfkeywords (hypersetup key) 1904	\prop_gput:Nnn 12, 2205, 2212
pdflang (hypersetup key) 11, 1890	\prop_if_empty:Nn 2286
pdflinkmargin (hypersetup key) 11	\prop_item:Nn 2303
pdfmanagement commands:		\prop_map_inline:Nn	713, 777, 813,
\pdfmanagement_add:nn 1386	840, 848, 859, 886, 894, 963, 1067,
		1103, 1113, 1157, 1245, 2281, 2288

`\prop_new:N` 2191, 2192
`\protect` 224, 3250
`\providecommand`
 140, 141, 2219, 2222, 2449, 2785
`\ProvidesFile` 3

R

`\ReadBookmarks` 3239
 ref commands:
`\ref_if_exist:nn` 218
`\ref_label:nn` 214
`\ref_value:nn` 18, 235
 regex commands:
`\regex_const:Nn` 301
`\regex_extract_once:NnNTF` 1658, 1689
`\relax` 2452, 2453,
 2494, 2782, 2829, 2833, 2903, 2907,
 3010, 3011, 3019, 3023, 3050, 3054,
 3134, 3142, 3169, 3177, 3301, 3310
`\RemoveFromHook` 1267
`\RenewDocumentCommand` 155
`\RequirePackage` 7, 19, 204, 3237
`run` (hypersetup key) 8, 1245
`runborder` (hypersetup key) 11
`runborderstyle` (hypersetup key) .. 11, 859
`runcolor` (hypersetup key) 749

S

`\selectfont` 2331, 2343, 2355, 2367
 seq commands:
`\seq_const_from_clist:Nn` 250
`\seq_count:N` 1817, 1843
`\seq_get_right:NN` 1819
`\seq_item:Nn` 1660,
 1691, 1813, 1861, 1863, 1873, 1878
`\seq_map_inline:Nn`
 . 287, 291, 295, 749, 1044, 1140, 2298
`\seq_new:N` 240
`\seq_put_right:Nn`
 1850, 1851, 1852, 2290
`\seq_set_split:Nnn` 1812
`\seq_sort:Nn` 2292
`\setlength` 3267, 3338
`\setpdflinkmargin` 6
`\settowidth` 2464
`\show` 2525
 sort commands:
`\sort_return_same:` 2296
`\sort_return_swapped:` 2295
`\space` 228, 2533, 3012, 3029,
 3030, 3060, 3061, 3135, 3167, 3170
`\spacefactor` 366, 388
`\stockwidth` 3

str commands:

`\c_colon_str` 616
`\c_hash_str` 490
`\c_percent_str` 491
`\str_case:nnTF` 3192
`\str_case_e:nnTF` 1813
`\str_gset:Nn` 774
`\str_gset_eq:NN` 337
`\str_head:n` 2005, 2010
`\str_if_eq:nnTF`
 373, 1916, 1934, 1944, 2894
`\str_if_eq_p:nn` 581
`\str_lowercase:n`
 1813, 1939, 1987, 2006, 2011
`\str_new:N` 242, 247, 248, 286
`\str_remove_once:Nn` 487, 542, 594, 637
`\str_set:Nn` 319
`\str_set_convert:Nnnn` 6
`\str_set_eq:NN` 339
`\str_tail:n` 2006, 2011
`\str_uppercase:n` 1999, 2005, 2010
`\string` 2219,
 2222, 2243, 2261, 2805, 2809, 2820
`\subpdfbookmark` 3230
 sys commands:
`\c_sys_backend_str` 3192

T

TeX and LaTeX 2_ε commands:

`\@BOOKMARK` 3236
`\@CheckBox` 2685
`\@ChoiceMenu` 2450
`\@Form` 2274
`\@PushButton` 2575
`\@Reset` 2652
`\@Submit` 2616
`\@TextField` 2415
`\@bookmarksopenlevel` 1278
`\@bsphack` 213
`\@chapter` 3290, 3291
`\@currDisplay` 2464, 2533
`\@currValue` 2531, 2547
`\@currentHref`
 3271, 3284, 3295, 3325, 3342
`\@curropt` . 2449, 2459, 2462, 2527, 2530
`\@empty` 2418, 2434, 2850,
 2854, 2858, 2862, 2866, 2870, 2874,
 2878, 2882, 2886, 3030, 3061, 3062
`\@endForm` 2384
`\@esphack` 215
`\@firstofone` .. 2225, 2791, 2796, 2813
`\@firstoftwo` 3332
`\@for` 2460, 2528
`\@gobble` 2276

\@gobbletwo	3281	\Fld@flags	2920,
\@hyper@launch	30, 616	2973, 3006, 3042, 3081, 3130, 3165	
\@ifundefined	2276, 2805	\Fld@format@code	2854, 2856
\@latex@warning	225	\Fld@height	
\@mainaux	2218, 2221, 2241, 2260	2420, 2441, 2444, 2455, 2513, 2539,	
\@mkboth	3281	2552, 2570, 2619, 2655, 2691, 2709	
\@nil	2803, 2816, 2822	\Fld@keystroke@code	2850, 2852
\@part	3308, 3315	\Fld@listcount	2524, 2532, 2553, 2560
\@pdfauthor	16	\Fld@mappingname	2833, 2835, 2907, 2909
\@pdfborder	140	\Fld@maxlen	3117, 3119
\@pdfborderstyle	141	\Fld@menulength	2457, 2463, 2510
\@savsf	366, 388	\Fld@name	
\@schapter	3278, 3279	.. 2416, 2451, 2576, 2687, 2828, 2902	
\@secondoftwo	3334	\Fld@onblur@code	2870, 2872
\@sect	3329, 3330	\Fld@onclick@code	2995
\@spart	3321, 3322	\Fld@onenter@code	2882, 2884
\@ssect	3264, 3265	\Fld@onexit@code	2886, 2888
\@tempdima	2458, 2465, 2467, 2504,	\Fld@onfocus@code	2866, 2868
2505, 2506, 2510, 2511, 2512, 2513		\Fld@onmousedown@code	2874, 2876
\@tempdimb	2464, 2465	\Fld@onmouseup@code	2878, 2880
\@typeset@protect	3250	\Fld@pageobjref	
\c@secnumdepth	3298, 3316, 3331 2386, 2400, 2785, 2917,	
\calc@bm@number	3242	2970, 3003, 3039, 3078, 3127, 3162	
\check@bm@number	3241	\Fld@radiosymbol	3058
\Fld@additionalactions	2892, 2962,	\Fld@rotation	2932,
2996, 3032, 3069, 3116, 3156, 3185		2934, 2978, 2985, 2987, 3009, 3015,	
\Fld@align	2921, 3007, 3082	3017, 3046, 3048, 3086, 3095, 3097,	
\Fld@altname	2829, 2831, 2903, 2905	3133, 3138, 3140, 3168, 3173, 3175	
\Fld@annotflags	2916,	\Fld@submitflags	3154
2969, 3002, 3038, 3077, 3126, 3161		\Fld@validate@code	2858, 2860
\Fld@annotnames	2901, 2918,	\Fld@value	
2971, 3004, 3040, 3079, 3128, 3163		.. 2418, 2434, 2453, 2494, 2495, 3115	
\Fld@bcolor	2940, 2942, 3023,	\Fld@width	2419,
3025, 3054, 3056, 3090, 3103, 3105		2433, 2440, 2444, 2454, 2493, 2504,	
\fld@bcolor	3011	2506, 2538, 2552, 2570, 2588, 2618,	
\Fld@bordercolor	2936,	2629, 2654, 2666, 2690, 2700, 2708	
2938, 2980, 2989, 2991, 3010, 3019,		\Fld@X@additionalactions	
3021, 3050, 3052, 3088, 3099, 3101,	 2849, 2894, 2897	
3134, 3142, 3144, 3169, 3177, 3179		\Form@action	3152
\Fld@borderstyle	2922,	\G@refundefinedtrue	224
2975, 3008, 3044, 3083, 3132, 3183		\H@old@part	3308, 3319
\Fld@borderwidth	2511, 2512, 2922,	\H@old@schapter	3278, 3287
2975, 3008, 3044, 3083, 3132, 3183		\H@old@sect	3329, 3340, 3349
\Fld@calculate@code	2234, 2862, 2864	\H@old@spart	3321, 3327
\Fld@calculate@sortkey	2245	\H@old@ssect	3264, 3269
\Fld@cbsymbol	2944	\Hy@abspage	360
\Fld@charsize		\Hy@activeanchorfalse	405
.. 2510, 2948, 3029, 3060, 3111, 3167		\Hy@activeanchortrue	398
\Fld@checkequals	2462, 2530	\Hy@AtBeginDocument	2199, 2215
\Fld@choices	3031	\Hy@bookmarksnumberedfalse	1271
\Fld@color	2949, 2951, 3030, 3061, 3112	\Hy@bookmarksnumberedtrue	1272
\Fld@default		\Hy@bookmarksopenfalse	1275
... 2417, 2434, 2452, 2495, 2525,		\Hy@bookmarksopentruue	1276
2526, 2688, 3062, 3066, 3067, 3114		\Hy@bookmarkstype	1279

\Hy@chapapp	3282, 3293	\Hy@SectionHShift	
\Hy@CleanupFile	521		3248, 3257, 3267, 3338
\Hy@colorlink	26	\Hy@StepCount	2463, 2532
\Hy@currentbookmarklevel	3223	\Hy@temp	2238, 2253, 2802, 2803
\Hy@DisableOption	147, 154	\Hy@unicodetrue	2842
\Hy@drafttrue	1190	\Hy@unicodetrue	145
\Hy@escapeform		\Hy@VerboseAnchor	368
	2438, 2536, 2566, 2594,	\Hy@verbosefalse	1181
	2633, 2668, 2706, 2787, 2794, 2797	\Hy@VerboseLinkStart	438, 452
\Hy@escapestring	2789,	\Hy@VerboseLinkStop	
	2791, 2796, 2799, 2800, 2803, 2804,		444, 465, 504, 555, 608, 650
	2813, 2816, 2852, 2856, 2860, 2864,	\Hy@verbosetrue	1180
	2868, 2872, 2876, 2880, 2884, 2888,	\Hy@VersionChecked	136
	2944, 2995, 3058, 3114, 3115, 3152	\Hy@WrapperDef	364
\Hy@finaltrue	1195	\Hy@xspace@end	441, 503, 554, 607, 649
\Hy@FormObjects	2280, 2720, 2782	\HyAnn@AbsPageLabel	
\Hy@gtemp	2845, 2847		2385, 2391, 2437,
\Hy@href@nextactionraw			2535, 2565, 2593, 2632, 2667, 2705
	485, 487, 488, 540,	\HyAnn@Count	2389, 2390, 2393, 2394,
	542, 543, 592, 594, 595, 635, 637, 638		2395, 2396, 2397, 2398, 2402, 2406
\Hy@href@page	535	\HyField@AddToFields	2228, 2445,
\Hy@hypertextnamesfalse	1205		2554, 2571, 2602, 2646, 2675, 2712
\Hy@hypertextnamestrue	1204	\HyField@afields	2194
\Hy@linkfileprefix	1207	\HyField@AfterAuxOpen	2199, 2225, 2258
\Hy@linktoc	1230	\HyField@AuxAddToCoFields	
\Hy@localanchornamefalse	1210		2198, 2210, 2222, 2243
\Hy@localanchornamestrue	1209	\HyField@AuxAddToFields	
\Hy@MakeCurrentHrefAuto			2197, 2203, 2219, 2261
	3266, 3282, 3293, 3323, 3337	\HyField@cofields	2195
\Hy@MakeRemoteAction	531	\HyField@FlagsCheckBox	2697
\Hy@naturalnamesfalse	1214	\HyField@FlagsChoice	2503
\Hy@naturalnamestrue	1213	\HyField@FlagsPushButton	
\Hy@next	3292, 3301, 3304		2585, 2625, 2664
\Hy@numberline	143, 3224	\HyField@FlagsRadioButton	2499
\Hy@org@chapter	3290, 3305	\HyField@FlagsSubmit	2626
\Hy@OutlineName	3240	\HyField@FlagsText	2432
\Hy@OutlineRerunCheck	3238	\HyField@PDFChoices	2563
\Hy@pageanchorfalse	1218	\HyField@SetKeys	2428, 2469,
\Hy@pageanchortrue	1217		2472, 2491, 2578, 2621, 2657, 2692
\Hy@pdfmajorversion	149	\HyPat@ObjRef	3196, 3203, 3210, 3217
\Hy@pdfminorversion	148	\hyper@anchor	390
\Hy@pdfstringtrue	22, 332	\hyper@anchorend	
\Hy@pdfversion	2840		390, 3259, 3284, 3295, 3325
\Hy@plainpagesfalse	1222	\hyper@anchorstart	
\Hy@plainpagetrue	1221		390, 3259, 3284, 3295, 3325
\Hy@pstringdef	342, 2820	\hyper@link	26, 26, 434
\Hy@PutCatalog	348	\hyper@linkend	26, 26, 459
\Hy@raisedlink	3258, 3283, 3294, 3324	\hyper@linkfile	515
\Hy@RestoreLastskip	387	\hyper@linklaunch	29, 565, 619
\Hy@safe@activetrue	492, 2237	\hyper@linknamed	30, 157, 625
\Hy@SaveLastskip	367	\hyper@linkstart	26, 26, 448
\Hy@secnum@part	3311, 3313, 3316	\hyper@linkurl	472
\Hy@SectionAnchorHref	3249, 3271, 3342	\HyPL@Labels	350, 360
		\HyPL@StorePageLabel	348

	W		X
\write	2218, 2221, 2241, 2260	\x	2470, 2490, 3274, 3276, 3345, 3347