

The `hyperref-generic` module

A generic driver for `hyperref`

The L^AT_EX Project*

Version 0.95c, released 2021-03-17

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 9—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/graphicsx`, `geometry`, the KOMA-classes, memoir, ... all try to set this.

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

6 Commands to create “external” references

`hyperref` has three commands related to external references like URL and file: `\url`, `\nolinkurl` and `\href`. The first two take one argument, while the last has two: the url and some free text.

`\url` and `\href` create link annotations. `\url` creates always an URI type, `\href` creates URI, GoToR and Launch depending on the structure of the argument.

`\href` has to create a (in the PDF) valid url or file name from its first argument. `\url` has to create a (in the PDF) valid url from its only argument and has also to print this argument as url. `\nolinkurl` only prints the url.

For the printing `\url` and `\nolinkurl` rely on the `url` package and its `\Url` command.

(Expandable) commands are expanded and special chars can also be input by commands but beside this no conversion is done: for all input `hyperref` basically assumes that

the input is already a valid percent encoded url or a valid file name. `hyperref` also doesn't extend or add protocols.

As nowadays everyone is used to copy and paste links with all sorts of unicode into a browser and they work the `hyperref` input is clearly rather restricted.

So the new driver tries to extend the input and print options. Both `\href` and `\url` can now be told to accept non-ascii url's and to convert them internally to percent encoding. It is possible to define a standard protocol and so to avoid to have to type it all the time.

But extending the *print* options for `\url` and `\nolinkurl` while still using the `url`-package is hard to impossible in pdfL^AT_EX due to the way the `url` package works. Some chars can be added with the help of `\UrlSpecial` (at the cost of warnings) but it doesn't work for every input and documenting and explaining all the edge cases is no joy. So instead the new driver offers here the option to use different commands to format the printed output. It must be noted that this disable the special "hyphenation" method of url's.

6.1 Special problem: links to files

When a file is linked with `\href` than normally it is added as URI link. The exceptions are PDF's: for them PDF has the special type GoToR which allows also to link to a destination or a special page.

After a number of tests with various PDF viewer established that non-ascii files names don't work at all with a simple file name specification GoToR links now use a full filespec dictionary. This works better, but still no every PDF viewer support this correctly. on various system.

The following can be used to test viewers. It assumes that a `test.pdf`, a `grüßpdf.pdf` and a `grüße.txt` are in the current folder.

```
test-ascii
test grüßpdf.pdf
test grüße.txt
```

6.2 Splits

`\href` tries to be clever and to detect from the argument if a url or a file link or a launch command should be created.

The rules are not trivial, and they make the code complicated. This detection also makes it more difficult to handle special cases like non-ascii input for the link types.

For this reason three new commands have been create:

- `\hrefurl` for standard urls (and non-pdf files)
- `\hrefpdf` for references to pdf files
- `\hrefrun` for launch links

The new commands don't use prefixes like `\href`. Their argument should be the real content.

6.3 Options

All `\href` commands and `\url` have an option argument for keyval syntax. It accepts the following keys. Not all keys make sense for all keys, but they don't error, they are silently ignored.

key	applicable commands	note
<code>urlencode</code>	<code>\hrefurl</code>	if set the code will convert the argument to percent encoding. This allows non-ascii input.
<code>protocol</code>	<code>\hrefurl</code> , <code>\url</code>	This sets a prefix/protocol that is added to the url, see below.
<code>format</code>	<code>\url</code>	a command used to format the printed text. It replaces the standard <code>\Url</code> . This can improve non-ascii typesetting at the cost of losing the special line breaking.
<code>destination</code>	<code>\href</code> , <code>\hrefpdf</code>	A destination name in the PDF
<code>page</code>	<code>\href</code> , <code>\hrefpdf</code>	destination page, default: 1
<code>pdfremotestartview</code>	<code>\href</code> , <code>\hrefpdf</code>	start view, default: Fit
<code>ismap</code>	<code>\href</code> , <code>\hrefurl</code>	see PDF reference
<code>afrelationship</code>	<code>\href</code> , <code>\hrefpdf</code>	Changes the <code>/AFRelationship</code> key of the filespec dictionary. The value should be a PDF name without the starting slash.
<code>run-parameter</code>	<code>\hreflaunch</code>	run parameter (see the PDF reference)
<code>nextactionraw</code>	various	puts a <code>/Next</code> entry in the action dictionary (see the PDF reference)

The first four keys can be set also in `\hypersetup` for all following commands in the current group through the keys `href/urlencode`, `href/protocol`, `href/destination`, `href/format`.

It is possible to define own url commands with specific options e.g. with

```
\NewDocumentCommand\myurl{0{}}{\url[protocol=https://,format=\textsc,#1]}
```

7 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.

7.1 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.

7.2 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 `cmk`. `rgb` is the default. It does *not* change the model of text colors. Be aware that while the PDF format allows `cmk` (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`.

7.3 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 be possible to change this.

7.4 Changed behaviour

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

Color keys accept the following input syntax:

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

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

frenchlinks The option `frenchlinks` does nothing at all.

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

link margin The driver sets a default link margin—this is identical to `pdftex` and `luatex` driver, but a change for the `xetex` and `dvips` driver. The (undocumented) command `\setpdflinkmargin` does nothing. Use either the key `pdflinkmargin` or `\pdfannot_link_margin:n` to change the margin. See also the description in section 14 and in the `hyperref` manual.

8 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.

9 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 14. 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**

10 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.

url (*setup key*)

file (*setup key*)

run (*setup key*)

menu (*setup key*)

11 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.

12 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.

13 Destinations

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

From these instructions two (`Fit` and `FitB`) don't take an argument. All others take one (`FitH`, `FitV`, `FitBH`, `FitBV`) or more (`XYZ`, `FitR`) arguments. These arguments

are normally coordinates, `XYZ` takes also a zoom factor. The coordinates are absolute coordinates in `bp` relative to the lower left corner of the PDF.

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

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

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

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

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

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

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

- `pdfstartview` and `pdfremotestartview` will pass the optional numbers or keyword after expansion as absolute coordinates and zoom. Missing numbers will be filled up with `null`.

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

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

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

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

14 Assorted key descriptions

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

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

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

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

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

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

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

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

will create

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

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.

`nesting` (setup key)

`pdfborder` (setup key)

`linkborder` (setup key)

`urlborder` (setup key)

`runborder` (setup key)

`menuborder` (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 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]
```

`urlborderstyle` (setup key)

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

`fileborderstyle` (setup key)

`runborderstyle` (setup key)

`menuborderstyle` (setup key)

`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'`.

`pdfmoddate` (setup key)

`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 the set the language in `\DeclareDocumentMetadata` instead.

File I

hyperref-generic driver implementation

```
1 <*package>
2 <@@=hyp>
3 \ProvidesFile{hgeneric-testphase.def}[2021-03-17 v0.95c %
4   generic Hyperref driver for the LaTeX PDF management testphase bundle]
5
6 \RequirePackage{etoolbox} %why?
7 \ExplSyntaxOn
```

```

8 \file_input:n {hyperref-colorschemes.def}
9 \ExplSyntaxOff

```

1 messages

Redirect the message name:

```

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

```

At first a message for the testing of the resource management

```

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

```

The pdfversion should be set in \DeclareDocumentMetadata

```

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

```

A generic message for ignored keys.

```

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

```

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

```

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

```

pdf/A message for not allowed Named actions

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

A message if the destination name is empty.

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

A message if the destination check fails

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

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

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

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:

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

```

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

```

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

```

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

```

2 Variants

```

134 \cs_generate_variant:Nn\pdf_destination:nn {nf}
135 \cs_generate_variant:Nn\pdf_object_ref:n {e}

```

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 }
```

Allow non-ascii in href, and add more href versions. We add a few new keys: `urlencode` to force percent encoding (`\hrefurl`, `\href`) `protocol` to add a protocol (`\hrefurl`, `\href` doesn't work here as it needs the colon for the split and the guessing.) `destination` to add a destination (`\hrefpdf`)

```

176
177 \bool_new:N \l__hyp_href_url_encode_bool
178 \bool_new:N \l__hyp_href_url_ismap_bool
179 \tl_new:N \l__hyp_href_url_protocol_tl
180 \tl_new:N \l__hyp_href_pdf_destination_tl
181 \tl_new:N \l__hyp_href_pdf_page_tl
182 \tl_new:N \l__hyp_href_run_parameter_tl
183 \cs_new_protected:Npn \__hyp_href_url_format {\begingroup\Url}
184
185
186 \keys_define:nn { hyp / href }
187 {
188   ,urlencode .bool_set:N = \l__hyp_href_url_encode_bool
189   ,format .code:n = { \cs_set_eq:NN \__hyp_href_url_format #1 },
190   ,protocol .tl_set:N = \l__hyp_href_url_protocol_tl
191   ,destination .tl_set:N = \l__hyp_href_pdf_destination_tl
192   ,pdfremotestartview .code:n =
193     {
194       \keys_set:nn { hyp / setup }
195       { pdfremotestartview = #1 }
196     }
197   ,page .code:n =
198     {
199       \tl_set:Nn \l__hyp_href_pdf_page_tl {#1}
200       \tl_set:Nn \Hy@href@page {#1}
201     }
202   ,ismap .bool_set:N = \l__hyp_href_url_ismap_bool
203   ,run-parameter .tl_set:N = \l__hyp_href_run_parameter_tl
204   ,nextactionraw .code:n =
205     { %perhaps some safety match later, see hyperref code
206       \tl_if_empty:nTF
207         {
208           \pdfdict_remove:nn{l_hyp/annot/A}{Next}
209         }
210         {
211           \pdfdict_put:nnn{l_hyp/annot/A}{Next}{#1}
212           \tl_set:Nn \Hy@href@nextactionraw {/Next~#1}
213           \keys_set:nn {hyp / setup }{ pdfnewwindow = true}
214         }
215     }
216   ,afrelationship .code:n =
217     {
218       \pdfdict_put:nnx
219       { l_pdffile/Filespec}{AFRelationship}{ \pdf_name_from_unicode_e:n {#1}}
220     }
221 }
222
223
224 \keys_define:nn { hyp / setup }

```

```

225 {
226   ,href / urlencode .bool_set:N = \l__hyp_href_url_encode_bool
227   ,href / urlencode .default:n = {true}
228   ,href / urlencode .initial:n = {false}
229   ,href / protocol .tl_set:N = \l__hyp_href_url_protocol_tl
230   ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
231   ,href / format .code:n = { \cs_set_eq:NN \__hyp_href_url_format #1 }
232 }
233
234
235
236 \DeclareRobustCommand*{\href}[1][]{%
237   \begingroup
238   \keys_set:nn { hyp / href } {#1}
239   \let\protect\unexpandable@protect
240   \bool_if:NTF \l__hyp_href_url_encode_bool
241     {
242       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
243     }
244     {
245       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
246     }
247   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
248 }
249
250
251 \DeclareRobustCommand*{\hrefurl}[1][]{
252   \group_begin:
253   \keys_set:nn { hyp / href } {#1}
254   \let\protect\unexpandable@protect
255   \bool_if:NTF \l__hyp_href_url_encode_bool
256     {
257       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
258     }
259     {
260       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
261     }
262   \hyper@normalise\__hyp_href_url_aux:nn}
263
264 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
265 {
266   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
267   \group_end:
268 }
269
270
271 \DeclareRobustCommand*{\hrefpdf}[1]{}
272 {
273   \begingroup
274   \keys_set:nn { hyp / href } {#1}
275   \let\protect\unexpandable@protect
276   \hyper@normalise\__hyp_href_pdf_aux:nn
277 }
278

```

```

279 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
280 {
281   \exp_args:Nnno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}\endgroup
282 }
283
284 \DeclareRobustCommand*\hrefrun}[1] []
285 {
286   \group_begin:
287   \keys_set:nn { hyp / href } {#1}
288   \let\protect\@unexpandable@protect
289   \hyper@normalise\__hyp_href_run_aux:nn
290 }
291
292 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
293 {
294   \exp_args:Nnno\hyper@linklaunch{#2}{#1}{\l__hyp_href_run_parameter_tl}
295   \group_end:
296 }
297
298 \DeclareRobustCommand*\url}[1] []
299 {
300   \group_begin:
301   \keys_set:nn { hyp / href } {#1}
302   \let\protect\@unexpandable@protect
303   \bool_if:NTF \l__hyp_href_url_encode_bool
304   {
305     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
306   }
307   {
308     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
309   }
310   \hyper@normalise\__hyp_href_url_aux:n
311 }
312
313 \cs_new_protected:Npn \__hyp_href_url_aux:n #1
314 {
315   \exp_args:Nno
316   \hyper@linkurl{\__hyp_href_url_format {#1}}
317   {\l__hyp_href_url_protocol_tl#1}
318   \group_end:
319 }
320
321
322 \NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
323 \pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
324 \pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}
325 \pdfstringdefDisableCommands{\let\hrefrun\__hyp_secondoftwowithopt:wnn}

```

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.

```
326 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
327 {
328   %\tl_set:cn {@#1}{#2}
329   \AddToDocumentProperties[hyperref]{#1}{#2}
330 }
331 \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.

```
332 \hook_new:n{hyp/link/cite}
333 \color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
334 \color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
335 \keys_define:nn { hyp / setup }
336 {
337   ,citecolor .code:n = {\__hyp_color_set:nn {hyp/color/cite}{#1}}
338   ,citebordercolor
339     .code:n = {\__hyp_color_set:nn {hyp/color/citeborder}{#1}}
340 }
341 \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
342 {
343   \keys_set:nn { hyp / setup }
344   {
345     ,linkbordercolor= hyp/color/citeborder
346     ,linkcolor      = hyp/color/cite
347   }
348 }
```

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

```
349 \bool_lazy_and:nnF
350 { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
351 { \msg_error:nn { hyp}{ missing-resource-management } }
```

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

```
352 \legacy_if:nT { Hy@bookmarks }
353 {
354   \AddToHook{begindocument/before}[hyperref/bookmark]{\RequirePackage{bookmark}}
355 }
356 \legacy_if:nT { Hy@draft }
```

```

357 {
358   \PassOptionsToPackage{draft}{bookmark}
359 }

```

6 Reference and label commands

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

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

```

\__hyp_ref_label:nn
\__hyp_ref_if_exist:nn
\__hyp_ref_check:nn
\__hyp_ref_value:nn
360 %
361 \cs_new_protected:Npn \__hyp_ref_label:nn #1 #2 %label/attributes
362 {
363   \@bsphack
364   \ref_label:nn{#1}{#2}
365   \@esphack
366 }
367 \cs_generate_variant:Nn \__hyp_ref_label:nn {en}

```

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

```

368 \prg_new_eq_conditional:Nn \__hyp_ref_if_exist:nn \ref_if_exist:nn { p , T , F, TF }
369 \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.

```

370 \cs_new_protected:Npn \__hyp_ref_check:nn #1 #2 %label/attribute
371 {
372   \__hyp_ref_if_exist:nnF {#1}{#2}
373   {
374     \protect\G@refundefinedtrue
375     \@latex@warning
376     {
377       Reference~'\tl_to_str:n {#1}'~with~attribute~'\tl_to_str:n {#2}'~
378       on~page~\thepage~\space undefined
379     }
380   }
381 }
382 \cs_generate_variant:Nn \__hyp_ref_check:nn {en}

```

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

```

383 \cs_new:Npn \__hyp_ref_value:nn #1 #2
384 {
385   \ref_value:nn{#1}{#2}
386 }
387 \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 388 \box_new:N \l__hyp_tmpa_box
\l__hyp_tmpa_int 389 \tl_new:N \l__hyp_tmpa_tl
\l__hyp_tmpa_box 390 \seq_new:N \l__hyp_tmpa_seq
\l__hyp_tmpa_str 391 \int_new:N \l__hyp_tmpa_int
392 \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 393 \tl_new:N \l__hyp_dest_name_tmpa_tl
\l__hyp_filename_tmpa_tl 394 \tl_new:N \l__hyp_uri_tmpa_tl
\__hyp_text_tmpa_str\g__hyp_text_tmpa_str 395 \tl_new:N \l__hyp_filename_tmpa_tl
396 \tl_new:N \l__hyp_para_tmpa_tl
397 \str_new:N \l__hyp_text_tmpa_str
398 \str_new:N \g__hyp_text_tmpa_str
```

TODO: document and check use!

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

7.2 Constants

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

```
399 \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`

```
400 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
401 {url,link,file,menu,run}
402 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
403 {
404     URI    = url,
405     GoTo   = link,
406     GoToR  = file,
407     Named  = menu,
408     Launch = run
409 }
410 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
411 {
412     url    = URI,
413     link   = GoTo,
414     file   = GoToR,
415     menu   = Named,
416     run    = Launch
417 }
418
```

(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!

```
419 \tl_new:N \g__hyp_dest_pdfstartpage_tl
420 \tl_new:N \g__hyp_dest_pdfstartview_tl
421 \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

422 \tl_new:N \l__hyp_text_enc_uri_print_tl
423 \tl_new:N \l__hyp_text_enc_info_print_tl
424 \tl_new:N \l__hyp_text_enc_dest_tl
425 \tl_new:N \l__hyp_text_enc_dest_print_tl
426 \tl_new:N \l__hyp_text_enc_file_print_tl
427 \tl_new:N \l__hyp_text_enc_para_print_tl
428
429 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
430 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
431 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
432 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
433 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
434 \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.

```
435 \tl_new:N \l__hyp_dest_pdfview_tl
```

(End definition for `\l__hyp_dest_pdfview_tl`.)

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

`hyp/annot/url` (color name)

`hyp/annot/file` (color name)

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

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

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

```
436 \str_new:N \g__hyp_bordercolormodel_str
```

(End definition for `\g__hyp_bordercolormodel_str`.)

7.4 Booleans

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

```
\l_hyp_annot_colorlink_bool
\l_hyp_annot_colorurl_bool
\l_hyp_annot_colorfile_bool
\l_hyp_annot_colorrund_bool
\l_hyp_annot_colormenu_bool
437 \seq_map_inline:Nn \c__hyp_annot_types_seq
438 {
439   \bool_new:c {l_hyp_annot_color#1_bool}
440 }
```

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

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

```
\l_hyp_annot_ocgcolorlink_bool
\l_hyp_annot_ocgcolorurl_bool
\l_hyp_annot_ocgcolorfile_bool
\l_hyp_annot_ocgcolorrund_bool
\l_hyp_annot_ocgcolormenu_bool
441 \seq_map_inline:Nn \c__hyp_annot_types_seq
442 {
443   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
444 }
```

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

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

```
445 \seq_map_inline:Nn \c_pdfannot_link_types_seq
446 {
447   \bool_new:c {l__hyp_annot_#1_bool}
448   \bool_set_true:c {l__hyp_annot_#1_bool}
449 }
```

(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.

```
450 \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.

```
451 \regex_const:Nn \c__hyp_dest_startview_regex
452 {
453   \A\ *
454   (?
455     (?:XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
456     |
457     (?:Fit\b|FitB\b)
458     |
459     (?:\bFitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1}
460     |
461     (?:FitR (?:\ +\d+|\ +\d*\.\d+)){4}\ )
462   )
463 }
```

(End definition for `\c__hyp_dest_startview_regex`.)

7.7 PDF dictionaries

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

```
464 \pdfdict_new:n {l__hyp_page/Trans}
465 \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`

```
466 \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
467 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
468 {
469   \str_set:Nx #2 {\text_purify:n { #1 } }
470 }
```

(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
471 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
472 {
473
474 }
```

(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`

```

475 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
476 {
477   \pdf_string_from_unicode:nVN { #1 } #2 #2
478 }

```

(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`

```

479 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
480 {
481   \group_begin:
482   \Hy@pdfstringtrue
483   \hook_use:n {hyp/text/pdfstring}
484   \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
485   \__hyp_text_cleanup:N \l__hyp_text_tmpa_str
486   \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
487   \str_gset_eq:NN \g__hyp_text_tmpa_str\l__hyp_text_tmpa_str
488   \group_end:
489   \str_set_eq:NN #3 \g__hyp_text_tmpa_str
490 }
491 \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

```

492 \cs_new_protected:Npn \Hy@pstringdef #1 #2
493 { \__hyp_text_pdfstring:xnN {#2} {utf8/string-raw}#1 }

```

This is a special version for info keys:

`__hyp_text_pdfstring_info:nN`

```

494 \cs_new_protected:Npn \__hyp_text_pdfstring_info:nN #1 #2
495 {
496   \__hyp_text_pdfstring:noN { #1 } { \l__hyp_text_enc_info_print_tl } #2
497 }

```

(End definition for __hyp_text_pdfstring_info:nN.)

9 Pagelabels

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

The current implementation in hyperref/hyperref drivers:

xetex: hxtex.def, line 80-110

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

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

dvipdfm: identical to xetex

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

luatex identical to pdftex

The code in `hyperref` inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-28 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
498 \cs_new_protected:Npn \__hyp_PageLabels_gpush:
499   {
500     \pdfmanagement_add:nnx {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
501   }
502
503 \def\Hy@PutCatalog #1 {}
504
505
506 \legacy_if:nT { Hy@pdfpagelabels }
507   {
508     \cs_set_protected:Npn \HyPL@StorePageLabel #1
509       {
510         \tl_gput_right:Nx \HyPL@Labels { \the\Hy@abspage<<#1>> }
511         \__hyp_PageLabels_gpush:
512       }
513   }
```

(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
514 \Hy@WrapperDef \__hyp_destination:nn #1 #2
515 {
516   \mode_if_horizontal:T { \@savs\spacefactor }
517   \Hy@SaveLastskip      %defined in hyperref
518   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
519   \__hyp_text_pdfstring:xoN
520   { \HyperDestNameFilter{#1} }
521   { \l__hyp_text_enc_dest_tl }
522   \l__hyp_tmpa_tl
523   \str_if_eq:nnTF {#2} {fitrbx}
524   {
525     \exp_args:NV
526     \pdf_destination:nnnn \l__hyp_tmpa_tl
527     { \box_wd:N \l__hyp_dest_box }
528     { \box_ht:N \l__hyp_dest_box }
529     { \box_dp:N \l__hyp_dest_box }
530   }
531   {
532     \exp_args:NV
533     \pdf_destination:nf
534     { \l__hyp_tmpa_tl }
535     { #2 }
536   }
537   \Hy@RestoreLastskip %defined in hyperref
538   \mode_if_horizontal:T { \spacefactor\@savs }
539 }
```

(End definition for `__hyp_destination:nn`.)

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

```
\hyper@anchor
\hyper@anchorstart
\hyper@anchorend
540 \cs_new_protected:Npn \hyper@anchor #1
541 {
542   \exp_args:NnV
543   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
544 }
```

```

545
546 \cs_new_protected:Npn \hyper@anchorstart #1
547 {
548   \Hy@activeanchortrue
549   \exp_args:NnV
550   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
551 }
552
553 \cs_new_protected:Npn \hyper@anchorend
554 {
555   \Hy@activeanchorfalse
556 }

```

(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:

```

557 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
558 {
559   \mode_leave_vertical:
560   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
561   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
562   {
563     \msg_warning:nnx
564     { hyp }
565     { empty-destination-name }
566     { \c__hyp_dest_undefined_tl }
567     \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
568   }
569   {
570     \__hyp_text_pdfstring:xoN
571     { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
572     { \l__hyp_text_enc_dest_tl }
573     \l__hyp_dest_name_tmpa_tl
574   }
575   \exp_args:No
576   \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
577 }
578
579 \cs_new_protected:Npn \__hyp_link_goto_end:
580 {
581   \pdfannot_link_goto_end:
582 }

```

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.

```
583
584 \cs_new_protected:Npn \hyper@link #1 #2 #3 %#1 context, #2=destination name, #3 content
585 {
586   \bool_if:NTF \l__hyp_annot_GoTo_bool
587   {
588     \Hy@VerboseLinkStart{#1}{#2}
589     \group_begin:
590     \hook_use:n {hyp/link/#1}
591     \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
592     \__hyp_link_goto_end:
593     \group_end:
594     \Hy@VerboseLinkStop
595   }
596   {\let\protect\relax#3}}
597 }
598
599 \cs_new_protected:Npn \hyper@linkstart #1 #2 %#1 context, #2=destination name
600 {
601   \bool_if:NT \l__hyp_annot_GoTo_bool
602   {
603     \Hy@VerboseLinkStart{#1}{#2}% only for debug
604     \group_begin:
605     \hook_use:n {hyp/link/#1}
606     \__hyp_link_goto_begin:nw {#2}
607   }
608 }
609
610 \cs_new_protected:Npn \hyper@linkend
611 {
612   \bool_if:NT \l__hyp_annot_GoTo_bool
613   {
614     \__hyp_link_goto_end:
615     \group_end:
```

```

616         \Hy@VerboseLinkStop
617     }
618 }

```

10.3 URI links

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

```

619 \pdfdict_new:n {l_hyp/annot/A/URI}
620 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
621 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
622
623 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
624 {
625     \bool_if:NTF \l__hyp_annot_URI_bool
626     {
627         \group_begin:
628         \__hyp_text_pdfstring:xoN
629             { #2}
630             { \l__hyp_text_enc_uri_print_tl }
631             \l__hyp_uri_tmpa_tl
632         \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
633         \bool_if:NT \l__hyp_href_url_ismap_bool
634             {
635                 \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
636             }
637         \cs_set_eq:NN \# \c_hash_str
638         \cs_set_eq:NN \% \c_percent_str
639         \Hy@safe@activetrue
640         \mode_leave_vertical:
641         \pdfannot_dict_put:nnx {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
642         \pdfannot_link:nnx { URI }
643         {
644         }
645         {
646             \let\protect\relax
647             #1
648             \Hy@xspace@end
649             \Hy@VerboseLinkStop %where is the start??
650         }
651         \group_end:
652     }
653     {{\let\protect\relax#1}}
654 }
655

```

10.4 GoToR Links files

```

656 \pdfdict_new:n {l_hyp/annot/A/GoToR}
657 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
658 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
659
660
661 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname

```



```

662 {
663   \bool_if:NTF \l__hyp_annot_GoToR_bool
664   {
665     \group_begin:
666     \tl_set:Nx \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
667     \exp_args:Nx
668     \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
669     {
670       \pdfdict_put:nnx { l_pdffile/Filespec } { Subtype } { \pdf_name_from_unicode_e:n {ap
671       \exp_args:Nnox
672       \pdffile_embed_file:nnn
673       {}
674       {\l__hyp_filename_tmpa_tl }
675       {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
676     }
677     \pdfdict_put:nnx
678     {l_hyp/annot/A/GoToR}
679     {F}
680     {\pdf_object_ref:e {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
681     \__hyp_text_pdfstring:nnN
682     { #3 }
683     { \l__hyp_text_enc_dest_print_tl }
684     \l__hyp_dest_name_tmpa_tl
685     \tl_if_blank:eTF {#3}
686     {
687       \pdfdict_put:nnx {l_hyp/annot/A/GoToR} {D}
688       {
689         [
690           \int_eval:n
691           { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
692           /\l__hyp_dest_pdfremotestartview_tl
693         ]
694       }
695     }
696     {
697       \pdfdict_put:nno {l_hyp/annot/A/GoToR} {D} { \l__hyp_dest_name_tmpa_tl }
698     }
699     \mode_leave_vertical:
700     \pdf_object_unnamed_write:nx {dict} { \pdfdict_use:n {l_hyp/annot/A/GoToR} }
701     \pdfannot_dict_put:nnx {link/GoToR} {A} { \pdf_object_ref_last: }
702     \pdfannot_link:nxn %expansion??
703     { GoToR }
704     {
705     }
706     {
707       \let\protect\relax
708       #1\Hy@xspace@end
709       \Hy@VerboseLinkStop %where is the start??
710     }
711     \group_end:
712   }
713   {{\let\protect\relax#1}}
714 }

```

10.5 Launch links

We define `\hyper@linklaunch` for naming consistency

```

715 \pdfdict_new:n {l_hyp/annot/A/Launch}
716 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
717 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
718
719 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
720 {
721   \bool_if:NTF \l__hyp_annot_Launch_bool
722   {
723     \group_begin:
724     \__hyp_text_pdfstring:nnN
725       { #1 }
726       { \l__hyp_text_enc_file_print_tl }
727       \l__hyp_filename_tmpa_tl
728     \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
729     \__hyp_text_pdfstring:noN
730       { #3 }
731       { \l__hyp_text_enc_para_print_tl }
732       \l__hyp_para_tmpa_tl
733     \bool_if:nTF
734     {
735       \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
736       ||
737       \pdf_version_compare_p:Nn > {1.9}
738     }
739     {
740       \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
741     }
742     {
743       \pdfdict_put:nnx {l_hyp/annot/A/Launch}{Win}
744       {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
745     }
746     \mode_leave_vertical:
747     \pdfannot_dict_put:nnx {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}>>}
748     \pdfannot_link:nnx
749       { Launch }
750       {
751         % /A
752         % <<
753         % \pdfdict_use:n {l_hyp/annot/A/Launch}
754         % >>
755       }
756       {
757         \let\protect\relax
758         #2\Hy@xspace@end
759         \Hy@VerboseLinkStop %where is the start??
760       }
761     \group_end:
762   }
763   {{\let\protect\relax#2}}
764 }

```

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

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

```

765 \use:x
766 { % filename, anchor text, linkname
767   \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str ##1 \exp_not:N \ \ ##2 #
768 }
769 {
770   \hyper@linklaunch {#1}{#2}{#3}
771 }

```

10.6 Named links (menu)

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

```

772 \pdfdict_new:n {l_hyp/annot/A/Named}
773 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
774 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
775
776 \cs_new_protected:Npn \hyper@linknamed #1 #2 % #1 action, #2 link text
777 {
778   \bool_if:NTF \l__hyp_annot_Named_bool
779   {
780     \group_begin:
781     \pdfmeta_standard_verify:nnTF {named_actions}{#1}
782     {
783       \mode_leave_vertical:
784       \pdfdict_put:nnx {l_hyp/annot/A/Named}{N}
785       {\pdf_name_from_unicode_e:n{#1}}
786       \pdfannot_dict_put:nnx {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}>>}
787       \pdfannot_link:nxn { Named }
788       {
789         /A
790         <<
791         \pdfdict_use:n { l_hyp/annot/A/Named }
792         >>
793       }
794       {
795         #2
796         \Hy@xspace@end
797         \Hy@VerboseLinkStop
798       }
799     }
800     {
801       \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
802       #2
803     }
804     \group_end:
805   }
806   {\let\protect\relax#2}}
807 }
808

```

11 Link decorations

11.1 Functions to export and select colors

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

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

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

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

```

809 \cs_new_protected:Npn \_hyp_color_export:nnN #1 #2 #3
810 {
811   \tl_if_head_eq_charcode:nNTF {#1}[ %]
812   {
813     \_hyp_colormodel_export:wnnN #1 {#2} #3
814   }
815   {
816     \color_export:nnN {#1} {#2} #3
817   }
818 }
819
820 \cs_new_protected:Npn \_hyp_colormodel_export:wnnN [#1] #2 #3 #4
821 {
822   \color_export:nnnnN {#1}{#2}{#3}#4
823 }
824
825 \cs_generate_variant:Nn \_hyp_color_export:nnN {xVN}

```

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

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

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

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

```

826 \cs_new_protected:Npn \_hyp_color_select:n #1
827 {
828   \tl_if_head_eq_charcode:nNTF {#1}[ %]
829   {
830     \_hyp_colormodel_select_aux:wn #1
831   }
832   {
833     \color_select:n {#1}
834   }

```

```

835 }
836
837 \cs_new_protected:Npn \__hyp_colormodel_select_aux:wn [#1] #2
838 {
839   \color_select:nn {#1}{#2}
840 }
841
842 \cs_generate_variant:Nn \__hyp_color_select:n {e}

```

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

```

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

```

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

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

`__hyp_colormodel_set_aux:nwn`

```

843 \cs_new_protected:Npn \__hyp_color_set:nn #1 #2
844 {
845   \tl_if_head_eq_charcode:nNTF {#2}[ %]
846   {
847     \__hyp_colormodel_set_aux:nwn { #1 } #2
848   }
849   {
850     \color_set:nn {#1} {#2}
851   }
852 }
853
854 \cs_new_protected:Npn \__hyp_colormodel_set_aux:nwn #1 [#2] #3
855 {
856   \color_set:nnn {#1}{#2}{#3}
857 }
858
859 \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.

```

860 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
861 {
862   \hook_gput_code:nnn
863   {pdfannot/link/#2/begin}
864   {hyp/color}
865   {
866     \bool_if:cT { l_hyp_annot_color#1_bool }
867     {
868       \group_begin:
869       \color_select:n { hyp/color/#1}
870     }
871   }

```

```

871     }
872     \hook_gput_code:nnn
873     {pdfannot/link/#2/end}
874     {hyp/color}
875     {
876       \bool_if:cT { l_hyp_annot_color#1_bool }
877       {
878         \group_end:
879       }
880     }
881   }

```

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

```

882 \keys_define:nn { hyp / setup }
883 {
884   ,colorlinks .meta:n =
885   {
886     ,pdfborder={0~0~0}
887     ,pdfborderstyle=
888     ,colorurl  =#1
889     ,colorlink =#1
890     ,colorrun  =#1
891     ,colormenu =#1
892     ,colorfile =#1
893   }
894   ,colorlinks .default:n = {true}
895 }

```

`colorurl` (*setup key*)

`colorlink` (*setup key*)

`colorrun` (*setup key*)

`colormenu` (*setup key*)

`colorfile` (*setup key*)

`urlcolor` (*setup key*)

`linkcolor` (*setup key*)

`runcolor` (*setup key*)

`menucolor` (*setup key*)

`filecolor` (*setup key*)

`allcolor` (*setup key*)

```

896 \seq_map_inline:Nn \c__hyp_annot_types_seq
897 {
898   \keys_define:nn { hyp / setup }
899   {
900     ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
901     ,#1color .code:n =      { \__hyp_color_set:ne {hyp/color/#1}{##1} }
902   }
903 }
904
905 \keys_define:nn { hyp / setup }
906 {
907   ,allcolors .meta:n =
908   {
909     ,urlcolor=#1
910     ,linkcolor=#1
911     ,runcolor=#1
912     ,filecolor=#1
913     ,menucolor=#1
914   }
915   ,allcolors .value_required:n = true
916 }

```

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*)

```
917 \keys_define:nn { hyp / setup }
918 {
919   ,bordercolormodel .choices:nn =
920     {rgb,cmyk}
921     { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
922   ,bordercolormodel .initial:n = {rgb}
923 }

924 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
925 {
926   \keys_define:nn { hyp / setup }
927   {
928     #1bordercolor .code:n =
929     {
930       \tl_if_empty:nTF { ##1 }
931       {
932         \pdfannot_dict_remove:nn
933         {link/#2}
934         { C }
935       }
936       {
937         \__hyp_color_export:xVN {##1}\g__hyp_bordercolormodel_str \l__hyp_tmpa_tl
938         \pdfannot_dict_put:nnx
939         {link/#2}
940         { C }
941         { [\l__hyp_tmpa_tl] }
942       }
943     }
944   }
945 }

946
947 \keys_define:nn { hyp / setup }
948 {
949   ,allbordercolors .meta:n =
950   {
951     ,linkbordercolor=#1
952     ,urlbordercolor =#1
953     ,filebordercolor=#1
954     ,menubordercolor=#1
955     ,runbordercolor =#1
956   }
957   ,allbordercolors .value_required:n = true
958 }
959
```

11.3.2 Borderwidth and -arc

```

960 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
961 {
962   \keys_define:nn { hyp / setup }
963   {
964     #1border .code:n =
965     {
966       \tl_if_empty:nTF { ##1 }
967       {
968         \pdfannot_dict_remove:nn
969         {link/#2}
970         { Border }
971       }
972       {
973         \pdfannot_dict_put:nnn
974         {link/#2}
975         { Border }
976         { [##1] }
977       }
978     }
979   }
980 }
981 \keys_define:nn { hyp / setup }
982 {
983   ,pdfborder .code:n =
984   {
985     \tl_if_empty:nTF { #1 }
986     {
987       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
988       {
989         \pdfannot_dict_remove:nn
990         {link/##2}
991         { Border }
992       }
993     }
994     {
995       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
996       {
997         \pdfannot_dict_put:nnn
998         {link/##2}
999         { Border }
1000         { [#1] }
1001       }
1002     }
1003   }
1004   ,pdfborder .initial:n = {0~0~1},
1005 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1006 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1007 {
runborderstyle (setup key) 1008   \keys_define:nn { hyp / setup }
fileborderstyle (setup key)
menuborderstyle (setup key)

```



```

1009     {
1010         #1borderstyle .code:n =
1011         {
1012             \tl_if_empty:nTF { ##1 }
1013             {
1014                 \pdfannot_dict_remove:nn
1015                 {link/#2}
1016                 { BS }
1017             }
1018             {
1019                 \pdfannot_dict_put:nnn
1020                 {link/#2}
1021                 { BS }
1022                 { <<##1>> }
1023             }
1024         }
1025     }
1026 }
1027 \keys_define:nn { hyp / setup }
1028 {
1029     ,pdfborderstyle .code:n =
1030     {
1031         \tl_if_empty:nTF { #1 }
1032         {
1033             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1034             {
1035                 \pdfannot_dict_remove:nn
1036                 {link/##2}
1037                 { BS }
1038             }
1039         }
1040         {
1041             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1042             {
1043                 \pdfannot_dict_put:nnn
1044                 {link/##2}
1045                 { BS }
1046                 { <<##1>> }
1047             }
1048         }
1049     }
1050     ,pdfborderstyle .initial:n = {},
1051 }

```

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.

```

1052 \cs_new_protected:Npn \__hyp_ocg_init:
1053 {
1054     \pdf_object_new:nn { l__hyp_ocg_view_dict_obj } { dict }

```

```

1055 \pdf_object_new:nn { l__hyp_ocg_print_dict_obj } { dict }
1056 \pdf_object_new:nn { l__hyp_ocg_config_dict_obj } { dict }
1057 \pdf_object_new:nn { l__hyp_ocg_ref_array_obj } { array }
1058 \pdf_object_write:nx { l__hyp_ocg_ref_array_obj }
1059 {
1060   \pdf_object_ref:n { l__hyp_ocg_view_dict_obj }
1061   \c_space_tl
1062   \pdf_object_ref:n { l__hyp_ocg_print_dict_obj }
1063 }
1064 \pdf_object_write:nn { l__hyp_ocg_view_dict_obj }
1065 {
1066   /Type/OCG
1067   /Name(View)
1068   /Usage
1069   <<
1070     /Print <</PrintState/OFF>>~
1071     /View <</ViewState/ON >>~
1072   >>
1073 }
1074 \pdf_object_write:nn { l__hyp_ocg_print_dict_obj }
1075 {
1076   /Type/OCG
1077   /Name(Print)
1078   /Usage
1079   <<
1080     /Print <</PrintState/ON>>~
1081     /View <</ViewState/OFF>>~
1082   >>
1083 }
1084 \pdfmanagement_add:nxx { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {l__hyp_ocg_ref_array_obj} }
1085 \pdfmanagement_add:nxx { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {l__hyp_ocg_print_dict_obj} }
1086 \pdf_object_write:nx { l__hyp_ocg_config_dict_obj }
1087 {
1088   /OFF[\pdf_object_ref:n { l__hyp_ocg_print_dict_obj }]
1089   /AS[
1090     <<
1091       /Event/View
1092       /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
1093       /Category[/View]
1094     >>
1095     <<
1096       /Event/Print
1097       /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
1098       /Category[/Print]
1099     >>
1100     <<
1101       /Event/Export
1102       /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
1103       /Category[/Print]
1104     >>
1105   ]
1106 }
1107 \pdfmanagement_add:nxx { Catalog / OCProperties }{ D }{ \pdf_object_ref:n { l__hyp_ocg_ref_array_obj} }
1108 \cs_gset:Npn \__hyp_ocg_init: {}

```

```

1109 }
(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.
1110 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1111 {
1112   \hook_gput_code:nnn
1113     {pdfannot/link/#2/begin}
1114     {hyp/ocg}
1115     {
1116       \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1117       {
1118         \_hyp_ocg_init:
1119         \group_begin:
1120         \hbox_set:Nw \l__hyp_tmpa_box
1121       }
1122     }
1123   \hook_gput_code:nnn
1124     {pdfannot/link/#2/end}
1125     {hyp/ocg}
1126     {
1127       \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1128       {
1129         \hbox_set_end:
1130         \mbox
1131         {
1132           \pdf_bdcobject:nn {0C}{l__hyp_ocg_print_dict_obj}
1133           \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1134           \pdf_emc:
1135           \pdf_bdcobject:nn {0C}{l__hyp_ocg_view_dict_obj}
1136           \group_begin:
1137           \color_select:n { hyp/color/#1 }
1138           \box_use_drop:N \l__hyp_tmpa_box
1139           \group_end:
1140           \pdf_emc:
1141         }
1142         \group_end:
1143       }
1144     }
1145 }

```

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

```

ocgcolorlink (setup key) 1146 \bool_lazy_or:nnTF
ocgcolorurl (setup key) 1147 { \pdf_version_compare_p:Nn > {1.4} }
ocgcolorfile (setup key) 1148 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
ocgcolormenu (setup key) 1149 {
ocgcolorrun (setup key) 1150   \keys_define:nn { hyp / setup }
1151   {
1152     ,_ocgcolorlinks .meta:n =
1153     {
1154       ocgcolorlink=#1,
1155       ocgcolorurl=#1,
1156       ocgcolorfile=#1,

```

```

1157         ocgcolorrun=#1,
1158         ocgcolormenu=#1
1159     }
1160     ,_ocgcolorlinks .default:n = true
1161 }
1162 }
1163 {
1164     \keys_define:nn { hyp / setup }
1165     {
1166         ,_ocgcolorlinks .code:n =
1167         {
1168             \msg_warning:nnxx
1169             { hyp }
1170             { ignore-deprecated-or-unknown-option-in-pdf-version }
1171             { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1172         }
1173     }
1174 }
1175
1176 \keys_define:nn { hyp / setup }
1177 {
1178     ,ocgcolorlinks .choice:
1179     ,ocgcolorlinks / true .meta:n =
1180     {
1181         pdfborder      = {0~0~0},
1182         pdfborderstyle = {},
1183         colorlinks     = false,
1184         _ocgcolorlinks = true
1185     }
1186     ,ocgcolorlinks / false .meta:n =
1187     {
1188         _ocgcolorlinks = false
1189     }
1190     ,ocgcolorlinks .default:n = {true}
1191 }
1192
1193 \seq_map_inline:Nn \c__hyp_annot_types_seq
1194 {
1195     \bool_lazy_or:nnTF
1196     { \pdf_version_compare_p:Nn > {1.4} }
1197     { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1198     {
1199         \keys_define:nn { hyp / setup }
1200         {
1201             ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1202         }
1203     }
1204     {
1205         \keys_define:nn { hyp / setup }
1206         {
1207             ,ocgcolor#1 .code:n=
1208             {
1209                 \msg_warning:nnxx
1210                 { hyp }

```

```

1211         { ignore-deprecated-or-unknown-option-in-pdf-version }
1212         { ocgcolor#1 }
1213         { \pdf_version_major:.\pdf_version_minor: }
1214     }
1215 }
1216 }
1217 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1218 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1219 {
1220   \keys_define:nn { hyp / setup }
1221   {
1222     ,#1highlight .choices:nn =
1223     { /I, /N, /O, /P}
1224     {
1225       \pdfannot_dict_put:nnn
1226       {link/#2}
1227       { H }
1228       { ##1 }
1229     }
1230     ,#1highlight / .code:n =
1231     {
1232       \pdfannot_dict_remove:nn
1233       {link/#2}
1234       { H }
1235     }
1236     }
1237     ,#1highlight / unknown .code:n =
1238     {
1239       \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1240       { pdfhighlight }
1241       { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1242       { \exp_not:n {##1} }
1243     }
1244   }
1245 }
1246 }
1247
1248
1249 \keys_define:nn { hyp / setup }
1250 {
1251   ,pdfhighlight .choices:nn =
1252   { /I, /N, /O, /P}
1253   {
1254     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1255     {
1256       \pdfannot_dict_put:nnn
1257       {link/####2}
1258       { H }
1259       { #1 }
1260     }

```

```

1261     }
1262     ,pdfhighlight / .code:n =
1263     {
1264         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1265         {
1266             \pdfannot_dict_remove:nn
1267             {link/##2}
1268             { H }
1269         }
1270     }
1271     ,pdfhighlight .initial:n = {/I},
1272     ,pdfhighlight / unknown .code:n =
1273     {
1274         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1275         { pdfhighlight }
1276         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1277         { \exp_not:n {#1} }
1278     }
1279 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1280 \keys_define:nn { hyp / setup }
hideurl (setup key) 1281 {
hidefile (setup key) 1282     hidelinks .meta:n =
hiderun (setup key) 1283     {
hidemenu (setup key) 1284         ,colorlinks      = false
1285         ,ocgcolorlinks = false
1286         ,pdfborder      = { 0~0~0 }
1287         ,pdfborderstyle=
1288     }
1289 }
1290
1291 \seq_map_inline:Nn \c__hyp_annot_types_seq
1292 {
1293     \keys_define:nn { hyp / setup }
1294     {
1295         hide#1 .meta:n =
1296         {
1297             ,color#1      = false
1298             ,ocgcolor#1   = false
1299             ,#1border     = { 0~0~0 }
1300             ,#1borderstyle =
1301         }
1302     }
1303 }

```

11.7 color schemes and settings

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

```

colorscheme (setup key)
1304 \keys_define:nn { hyp / setup }
1305 {
1306   colorscheme .code:n =
1307   {
1308     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1309     {
1310       \keys_set:nn { hyp /setup }
1311       {
1312         ##1 = ##2
1313       }
1314     }
1315   }
1316 }
1317 \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)
1318 \keys_define:nn { hyp / setup }
1319 {
1320   ,unicode .code:n = {}
1321   ,pdfencoding .code:n = {}
1322   ,pdfversion .code:n =
1323   {
1324     \msg_warning:nn { hyp }{ pdfversion-disabled }
1325   }
1326 }
1327 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
debug (setup key)
draft (setup key)
final (setup key)
1328 \keys_define:nn { hyp / setup }
1329 {
1330   ,verbose .choice:
1331   ,verbose / true .code:n = { \Hy@verbosetrue}
1332   ,verbose / false .code:n = { \Hy@verbosefalse}
1333   ,verbose .default:n = {true}
1334   ,debug .meta:n = {verbose=#1}
1335   ,debug .default:n = {true}
1336 }
1337 \keys_define:nn { hyp / setup }
1338 {
1339   ,draft .code:n =
1340   {
1341     \Hy@drafttrue

```

```

1342         \PassOptionsToPackage{draft}{bookmark}
1343     }
1344     ,final .code:n =
1345     {
1346         \Hy@finaltrue
1347         \PassOptionsToPackage{final}{bookmark}
1348     }
1349 }

extension (setup key)
hypertextnames (setup key) 1350 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1351 {
    pageanchor (setup key) 1352 ,extension .tl_set:N = \XR@ext
    linktoc (setup key) 1353 ,extension .initial:n= pdf
    linktocpage (setup key) 1354 ,hypertextnames .choice:
    plainpages (setup key) 1355 ,hypertextnames / true .code:n = { \Hy@hypertextnamestrue}
localanchorname (setup key) 1356 ,hypertextnames / false .code:n = { \Hy@hypertextnamesfalse}
linkfileprefix (setup key) 1357 ,hypertextnames .default:n = {true}
1358 ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1359 ,localanchorname .choice:
1360 ,localanchorname / true .code:n = { \Hy@localanchornamestrue }
1361 ,localanchorname / false .code:n = { \Hy@localanchornamefalse }
1362 ,localanchorname .default:n = {true}
1363 ,naturalnames .choice:
1364 ,naturalnames / true .code:n = { \Hy@naturalnamestrue}
1365 ,naturalnames / false .code:n = { \Hy@naturalnamesfalse}
1366 ,naturalnames .default:n = {true}
1367 ,pageanchor .choice:
1368 ,pageanchor / true .code:n = { \Hy@pageanchortrue}
1369 ,pageanchor / false .code:n = { \Hy@pageanchorfalse}
1370 ,pageanchor .default:n = {true}
1371 ,plainpages .choice:
1372 ,plainpages / true .code:n = { \Hy@plainpagestrue}
1373 ,plainpages / false .code:n = { \Hy@plainpagesfalse}
1374 ,plainpages .default:n = {true}
1375 }
1376
1377 \keys_define:nn { hyp / setup }
1378 {
1379 ,linktoc .choices:nn = { none, section, all, page }
1380 {
1381     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1382 }
1383 ,linktoc / unknown .code:n =
1384 {
1385     \msg_warning:nxxx { hyp } { unknown-choice }
1386     { linktoc }
1387     { none, section, all, page }
1388     { \exp_not:n {#1} }
1389 }
1390 ,linktocpage .choice:
1391 ,linktocpage / true .meta:n = {linktoc=page}
1392 ,linktocpage / false .meta:n = {linktoc=section}
1393 ,linktocpage .default:n = true

```



```

1394 }
1395
link (setup key) This booleans allow to disable the link types.
url (setup key) 1396 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1397 {
menu (setup key) 1398 \keys_define:nn { hyp / setup }
run (setup key) 1399 {
1400 ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1401 }
1402 }

1403 \keys_define:nn { hyp / setup }
1404 {
1405 ,baseurl .code:n =
1406 {
1407 \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1408 \tl_if_empty:NTF \l__hyp_tmpa_tl
1409 {
1410 \pdfmanagement_remove:nn {Catalog} { URI }
1411 }
1412 {
1413 \pdfmanagement_add:nxx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1414 }
1415 }
1416 %only false does something ...
1417 ,bookmarks .choice:
1418 ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1419 ,bookmarks / true .code:n = {}
1420 ,bookmarks .default:n = {true}
1421 ,bookmarksnumbered .choice:
1422 ,bookmarksnumbered / false .code:n = { \Hy@bookmarksnumberedfalse }
1423 ,bookmarksnumbered / true .code:n = { \Hy@bookmarksnumberedtrue }
1424 ,bookmarksnumbered .default:n = {true}
1425 ,bookmarksopen .choice:
1426 ,bookmarksopen / false .code:n = { \Hy@bookmarksopenfalse }
1427 ,bookmarksopen / true .code:n = { \Hy@bookmarksopentruetrue }
1428 ,bookmarksopen .default:n = {true}
1429 ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1430 ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1431 ,pdfcenterwindow .choice:
1432 ,pdfcenterwindow / false .code:n =
1433 {
1434 \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1435 }
1436 ,pdfcenterwindow / true .code:n =
1437 {
1438 \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1439 }
1440 ,pdfcenterwindow / .code:n =
1441 {
1442 \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1443 }
1444 ,pdfcenterwindow / unknown .code:n =

```

```

1445     {
1446         \msg_warning:nnxx { hyp } { no-bool }
1447         { pdfcenterwindow }
1448         { \exp_not:n {#1} }
1449     }
1450 ,pdfcenterwindow .default:n = true
1451 ,pdfdirection .choice:
1452 ,pdfdirection / L2R .code:n =
1453 {
1454     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1455 }
1456 ,pdfdirection / R2L .code:n =
1457 {
1458     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1459 }
1460 ,pdfdirection / .code:n =
1461 {
1462     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1463 }
1464 ,pdfdirection / unknown .code:n =
1465 {
1466     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1467     { pdfdirection }
1468     { L2R , R2L }
1469     { \exp_not:n {#1} }
1470 }
1471 ,pdfdisplaydoctitle .choice:
1472 ,pdfdisplaydoctitle / false .code:n =
1473 {
1474     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1475 }
1476 ,pdfdisplaydoctitle / true .code:n =
1477 {
1478     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1479 }
1480 ,pdfdisplaydoctitle .default:n = true
1481 ,pdfduplex .choices:nn =
1482 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1483 {
1484     \pdf_version_compare:NnTF > {1.6}
1485     {
1486         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1487         { PrintDuplex } { /#1 }
1488     }
1489     {
1490         \msg_warning:nnxx
1491         {hyp}
1492         {ignore-deprecated-or-unknown-option-in-pdf-version}
1493         {pdfduplex}
1494         {\pdf_version:}
1495     }
1496 }%
1497 ,pdfduplex / .code:n =
1498 {

```

```

1499     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1500   }
1501 ,pdfduplex / unknown .code:n =
1502   {
1503     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1504     { pdfduplex }
1505     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1506     { \exp_not:n {#1} }
1507   }
1508 ,pdffitwindow .choice:
1509 ,pdffitwindow / false .code:n =
1510   {
1511     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1512   }
1513 ,pdffitwindow / true .code:n =
1514   {
1515     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1516   }
1517 ,pdffitwindow / .code:n =
1518   {
1519     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1520   }
1521 ,pdffitwindow .default:n = true
1522 ,pdffitwindow / unknown .code:n =
1523   {
1524     \msg_warning:nnxx { hyp } { no-bool }
1525     { pdffitwindow }
1526     { \exp_not:n {#1} }
1527   }
1528 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1529 ,pdflinkmargin .initial:n = {1pt}
1530 ,pdfmenubar .choice:
1531 ,pdfmenubar / true .code:n =
1532   {
1533     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1534   }
1535 ,pdfmenubar / false .code:n =
1536   {
1537     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1538     { HideMenubar } { true }
1539   }
1540 ,pdfmenubar / .code:n =
1541   {
1542     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1543   }
1544 ,pdfmenubar .default:n = true
1545 ,pdfmenubar / unknown .code:n =
1546   {
1547     \msg_warning:nnxx { hyp } { no-bool }
1548     { pdfmenubar }
1549     { \exp_not:n {#1} }
1550   }
1551 ,pdfnewwindow .choice:
1552 ,pdfnewwindow / true .code:n =

```

```

1553     {
1554         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1555         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1556     }
1557 ,pdfnewwindow / false .code:n =
1558     {
1559         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1560         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1561     }
1562 ,pdfnewwindow / .code:n =
1563     {
1564         \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1565         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1566     }
1567 ,pdfnonfullscreenpagemode .choices:nn =
1568     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1569     {
1570         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1571         { NonFullScreenPageMode } {/#1}
1572     }
1573 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1574     {
1575         \pdf_version_compare:NnTF < {1.6}
1576         {
1577             %message
1578         }
1579         {
1580             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1581             {NonFullScreenPageMode}{/UseAttachments}
1582         }
1583     }
1584 ,pdfnonfullscreenpagemode / .code:n =
1585     {
1586         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1587     }
1588 ,pdfnonfullscreenpagemode / unknown .code:n =
1589     {
1590         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1591         { pdfnonfullscreenpagemode }
1592         { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1593         { \exp_not:n {#1} }
1594     }
1595 ,pdfnumcopies .code:n =
1596     {
1597         \pdf_version_compare:NnTF > {1.6}
1598         {
1599             \tl_if_empty:nTF {#1}
1600             {
1601                 \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1602             }
1603             {
1604                 \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1605                 {NumCopies}{#1}
1606             }

```

```

1607     }
1608     {
1609         \msg_warning:nnxx
1610         {hyp}
1611         {ignore-deprecated-or-unknown-option-in-pdf-version}
1612         {pdfnumcopies}
1613         {\pdf_version:}
1614     }
1615 }
1616 ,pdfpagelayout .choices:nn =
1617 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1618 { \pdfmanagement_add:nnx {Catalog} { PageLayout }{ /#1 } }
1619 ,pdfpagelayout / .code:n =
1620 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1621 ,pdfpagelayout / unknown .code:n =
1622 {
1623     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1624     { pdfpagelayout }
1625     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1626     { \exp_not:n {#1} }
1627 }
1628 ,pdfpagemode .choices:nn =
1629 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1630 { \pdfmanagement_add:nnx {Catalog} { PageMode }{ /#1 } }
1631 ,pdfpagemode / UseAttachments .code:n =
1632 {
1633     \pdf_version_compare:NnTF > {1.5}
1634     {
1635         \pdfmanagement_add:nnx {Catalog} { PageMode }{ /UseAttachments }
1636     }
1637     {
1638         \msg_warning:nnxx
1639         {hyp}
1640         {ignore-deprecated-or-unknown-value-in-pdf-version}
1641         {UseAttachments}
1642         {\pdf_version:}
1643     }
1644 }
1645 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1646 ,pdfpagemode / unknown .code:n =
1647 {
1648     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1649     { pdfpagemode }
1650     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1651     { \exp_not:n {#1} }
1652 }
1653 ,pdfpagescrop .code:n =
1654 {
1655     \tl_if_empty:nTF %or blank?
1656     {
1657         \pdfmanagement_remove:nn {Pages} { CropBox }
1658     }
1659     {
1660         \pdfmanagement_add:nnx {Pages} { CropBox } { [#1] }

```

```

1661     }
1662   }
1663   ,pdfpicktraybypdfsize .choice:
1664   ,pdfpicktraybypdfsize / true .code:n =
1665   {
1666     \pdf_version_compare:NnTF > {1.6}
1667     {
1668       \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1669       { PickTrayByPDFSize } { true }
1670     }
1671     {
1672       \msg_warning:nnxx
1673       {hyp}
1674       {ignore-deprecated-or-unknown-option-in-pdf-version}
1675       {pdfpicktraybypdfsize}
1676       {\pdf_version:}
1677     }
1678   }
1679   ,pdfpicktraybypdfsize / false .code:n =
1680   {
1681     \pdf_version_compare:NnTF > {1.6}
1682     {
1683       \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1684       { PickTrayByPDFSize } { false }
1685     }
1686     {
1687       \msg_warning:nnxx
1688       {hyp}
1689       {ignore-deprecated-or-unknown-option-in-pdf-version}
1690       {pdfpicktraybypdfsize}
1691       {\pdf_version:}
1692     }
1693   }
1694   ,pdfpicktraybypdfsize / .code:n =
1695   {
1696     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1697   }
1698   ,pdfpicktraybypdfsize / unknown .code:n =
1699   {
1700     \msg_warning:nnxx { hyp } { no-bool }
1701     { picktraybypdfsize }
1702     { \exp_not:n {#1} }
1703   }
1704   ,pdfprintarea .choices:nn =
1705   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1706   {
1707     \pdf_version_compare:NnTF < {2.0}
1708     {
1709       \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1710       { PrintArea } { /#1 }
1711     }
1712     {
1713       \msg_warning:nnxx
1714       {hyp}

```

```

1715         {ignore-deprecated-or-unknown-option-in-pdf-version}
1716         {pdfprintarea}
1717         {\pdf_version:}
1718     }
1719 }%
1720 ,pdfprintarea / .code:n =
1721 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1722 ,pdfprintarea / unknown .code:n =
1723 {
1724     \msg_warning:nxxx { hyp } { unknown-choice+empty }
1725     { pdfprintarea }
1726     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1727     { \exp_not:n {#1} }
1728 }
1729 ,pdfprintclip .choices:nn =
1730 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1731 {
1732     \pdf_version_compare:NnTF < {2.0}
1733     {
1734         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1735         { PrintClip } { /#1 }
1736     }
1737     {
1738         \msg_warning:nxxx
1739         {hyp}
1740         {ignore-deprecated-or-unknown-option-in-pdf-version}
1741         {pdfprintclip}
1742         {\pdf_version:}
1743     }
1744 }%
1745 ,pdfprintclip / .code:n =
1746 {
1747     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1748 }
1749 ,pdfprintclip / unknown .code:n =
1750 {
1751     \msg_warning:nxxx { hyp } { unknown-choice+empty }
1752     { pdfprintclip }
1753     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1754     { \exp_not:n {#1} }
1755 }
1756 ,pdfprintpagerange .code:n =
1757 {
1758     \pdf_version_compare:NnTF > {1.6}
1759     {
1760         \tl_if_empty:nTF { #1}
1761         {
1762             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1763             { PrintPageRange }
1764         }
1765         {
1766             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1767             {PrintPageRange}{[#1]}
1768         }
1769     }

```

```

1769     }
1770     {
1771         \msg_warning:nxxx
1772         {hyp}
1773         {ignore-deprecated-or-unknown-option-in-pdf-version}
1774         {pdfprintpagerange}
1775         {\pdf_version:}
1776     }
1777 }
1778 ,pdfprintscaling .choices:nn =
1779 { None, AppDefault }
1780 {
1781     \pdf_version_compare:NnTF > {1.5}
1782     {
1783         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1784         { PrintScaling } { /#1 }
1785     }
1786     {
1787         \msg_warning:nxxx
1788         {hyp}
1789         {ignore-deprecated-or-unknown-option-in-pdf-version}
1790         {pdfprintscaling}
1791         {\pdf_version:}
1792     }
1793 }%
1794 ,pdfprintscaling / .code:n =
1795 {
1796     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1797 }
1798 ,pdfprintscaling / unknown .code:n =
1799 {
1800     \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1801     { pdfprintarea }
1802     { None, AppDefault }
1803     { \exp_not:n {#1} }
1804 }
1805 ,pdfremotestartview .code:n =
1806 {
1807     \tl_set:Nx \l__hyp_tmpa_tl {#1-null-null-null~}
1808     \exp_args:NNV
1809     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_seq
1810     {
1811         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1812     }
1813     {
1814         \msg_warning:nxxx {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1815         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1816     }
1817 }
1818 ,pdfremotestartview .initial:n = {Fit}
1819 % pdfstartpage is special as it shares code with pdfstartview
1820 ,pdfstartpage .code:n =
1821 {
1822     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }

```



```

1823     \bool_if:nTF
1824     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1825     {
1826         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1827     }
1828     {
1829         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1830         {
1831             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1832         }
1833     }
1834 }
1835 ,pdfstartpage .initial:n =1
1836 ,pdfstartview .code:n =
1837 {
1838     \tl_set:Nx \l__hyp_tmpa_tl {#1-null-null-null~}
1839     \exp_args:NNV
1840     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1841     {
1842         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1843     }
1844     {
1845         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1846         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1847     }
1848     \bool_if:nTF
1849     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1850     {
1851         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1852     }
1853     {
1854         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1855         {
1856             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1857         }
1858     }
1859 }
1860 ,pdfstartview .initial:n = Fit
1861 ,pdftoolbar .choice:
1862 ,pdftoolbar / true .code:n =
1863 {
1864     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1865 }
1866 ,pdftoolbar / false .code:n =
1867 {
1868     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
1869     { HideToolbar } { true }
1870 }
1871 ,pdftoolbar / true .code:n =
1872 {
1873     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1874 }
1875 ,pdftoolbar .default:n = true
1876 ,pdftoolbar / unknown .code:n =

```

```

1877     {
1878         \msg_warning:nxxx { hyp } { no-bool }
1879         { pdftoolbar }
1880         { \exp_not:n {#1} }
1881     }
1882 % pdfview see below.
1883 ,pdfviewarea .choices:nn =
1884 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1885 {
1886     \pdf_version_compare:NnTF < {2.0}
1887     {
1888         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1889         { ViewArea } { /#1 }
1890     }
1891     {
1892         \msg_warning:nxxx
1893         {hyp}
1894         {ignore-deprecated-or-unknown-option-in-pdf-version}
1895         {pdfviewarea}
1896         {\pdf_version:}
1897     }
1898 }%
1899 ,pdfviewarea / .code:n =
1900 {
1901     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1902 }
1903 ,pdfviewarea / unknown .code:n =
1904 {
1905     \msg_warning:nxxx { hyp } { unknown-choice+empty }
1906     { pdfviewarea }
1907     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1908     { \exp_not:n {#1} }
1909 }
1910 ,pdfviewclip .choices:nn =
1911 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1912 {
1913     \pdf_version_compare:NnTF < {2.0}
1914     {
1915         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1916         { ViewClip } { /#1 }
1917     }
1918     {
1919         \msg_warning:nxxx
1920         {hyp}
1921         {ignore-deprecated-or-unknown-option-in-pdf-version}
1922         {pdfviewclip}
1923         {\pdf_version:}
1924     }
1925 }%
1926 ,pdfviewclip / .code:n =
1927 {
1928     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
1929 }
1930 ,pdfviewclip / unknown .code:n =

```

```

1931 {
1932     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1933     { pdfviewclip }
1934     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1935     { \exp_not:n {#1} }
1936 }
1937 ,pdfwindowui .choice:
1938 ,pdfwindowui / true .code:n =
1939 {
1940     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
1941 }
1942 ,pdfwindowui / false .code:n =
1943 {
1944     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1945     { HideWindowUI } { true }
1946 }
1947 ,pdfwindowui / .code:n =
1948 {
1949     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
1950 }
1951 ,pdfwindowui / unknown .code:n =
1952 {
1953     \msg_warning:nnxx { hyp } { no-bool }
1954     { pdfwindowui }
1955     { \exp_not:n {#1} }
1956 }
1957 ,pdfwindowui .default:n = true
1958 }

```

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

```

1959 \keys_define:nn { hyp / setup }
1960 {
1961     ,pdfview .code:n =
1962     {
1963         \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
1964         \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } } {
1965             { xyz }
1966             {
1967                 \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
1968                 {
1969                     \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
1970                     \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
1971                     {
1972                         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1973                     }
1974                     {
1975                         \tl_set:Nx \l__hyp_dest_pdfview_tl
1976                         {
1977                             \fp_eval:n { \l__hyp_tmpa_tl * 100 }
1978                         }
1979                     }
1980                 }
1981             }
1982         }

```

```

1983         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1984     }
1985 }
1986 { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
1987 { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
1988 { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
1989 { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
1990 { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
1991 { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
1992 { fitr }
1993 {
1994     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
1995     {
1996         \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
1997     }
1998     {
1999         %ensure 4 values ...
2000         \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2001         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2002         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2003         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2004         \hbox_set_to_wd:Nnn \l__hyp_dest_box
2005         {
2006             \fp_eval:n
2007             {
2008                 round
2009                 (
2010                     abs
2011                     (
2012                         \seq_item:Nn\l__hyp_tmpa_seq{4}
2013                         -
2014                         (\seq_item:Nn\l__hyp_tmpa_seq{2})
2015                     ),
2016                     3
2017                 )
2018             }bp
2019         }{}
2020         \box_set_dp:Nn \l__hyp_dest_box
2021         {
2022             \fp_eval:n
2023             {
2024                 round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2025             }bp
2026         }
2027         \box_set_ht:Nn \l__hyp_dest_box
2028         {
2029             \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2030         }
2031     }
2032 }
2033 }
2034 {
2035     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2036     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}

```

```

2037     }
2038   }
2039   ,pdfview .initial:n = {xyz}
2040 }

```

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.

```

2041 \keys_define:nn { hyp / setup }
2042 {
2043   ,pdflang .code:n =
2044   {
2045     \tl_if_empty:nTF { #1 }
2046     {
2047       \pdfmanagement_remove:nn {Catalog} { Lang }
2048     }
2049     {
2050       \pdfmanagement_add:nx {Catalog} { Lang } { (#1) }
2051     }
2052     \__hyp_store_metadata:nn {pdflang}{#1}
2053   }
2054 }

```

12.3.1 “info keys”

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

```

pdfcreator (setup key) 2055 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
pdfsubject (setup key) 2056 {
pdfproducer (setup key) 2057   \keys_define:nn { hyp / setup }
pdfkeywords (setup key) 2058   {
2059     pdf#1 .code:n =
2060     {
2061       \tl_if_blank:nTF {##1}
2062       {
2063         \pdfmanagement_remove:nn {Info}{#2}
2064       }
2065       {
2066         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2067         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2068         {
2069           \pdfmanagement_add:nx {Info}{#2}{\l__hyp_tmpa_str}
2070         }
2071       }
2072       \__hyp_store_metadata:nn {pdf#1}{##1}
2073     }
2074   }
2075   \keys_define:nn { hyp / info }
2076   {
2077     #2 .code:n =

```

```

2078         {
2079             \tl_if_blank:nTF {##1}
2080             {
2081                 \pdfmanagement_remove:nn {Info}{#2}
2082             }
2083             {
2084                 \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2085                 \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2086                 {
2087                     \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2088                 }
2089             }
2090             \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2091         }
2092     ,unknown .code:n =
2093     {
2094         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2095         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2096         {
2097             \exp_args:Nno
2098             \pdfmanagement_add:nnx {Info}
2099             { \l_keys_key_str } {\l__hyp_tmpa_str}
2100         }
2101     }
2102 }
2103 }
2104 \__hyp_setup_info_key:nn {author} {Author}
2105 \__hyp_setup_info_key:nn {title} {Title}
2106 \__hyp_setup_info_key:nn {producer} {Producer}
2107 \__hyp_setup_info_key:nn {creator} {Creator}
2108 % ignored key: addtopdfcreator
2109 \__hyp_setup_info_key:nn {subject} {Subject}
2110 \__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.

```

2111 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2112 {
2113     \keys_define:nn { hyp / setup }
2114     {
2115         pdf#1 .code:n =
2116         {
2117             \tl_if_blank:nTF {##1}
2118             {
2119                 \pdfmanagement_remove:nn {Info}{#2}
2120             }
2121             {
2122                 \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
2123             }
2124             \__hyp_store_metadata:nn {pdf#1}{##1}
2125         }
2126     }
2127     \keys_define:nn { hyp / info }
2128     {

```

```

2129         #2 .code:n =
2130         {
2131             \tl_if_blank:nTF {##1}
2132             {
2133                 \pdfmanagement_remove:nn {Info}{#2}
2134             }
2135             {
2136                 \pdfmanagement_add:nxx {Info}{#2}{(\tl_to_str:n {##1})}
2137             }
2138             \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2139         }
2140     }
2141 }
2142
2143 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2144 \__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 ...

```

2145 \keys_define:nn { hyp / setup }
2146 {
2147     ,pdftrapped .code:n =
2148     {
2149         \exp_args:Nne
2150         \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2151     }
2152     ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2153     {
2154         \pdfmanagement_add:nxx {Info}{Trapped}
2155         {/
2156             \str_uppercase:f { \str_head:n { #1 } }
2157             \str_lowercase:f { \str_tail:n { #1 } }
2158         }
2159         \__hyp_store_metadata:nx {pdftrapped}
2160         {
2161             \str_uppercase:f { \str_head:n { #1 } }
2162             \str_lowercase:f { \str_tail:n { #1 } }
2163         }
2164     }
2165     ,_pdftrapped / unknown .code:n =
2166     {
2167         \msg_warning:nnxxx { hyp } { unknown-choice }
2168         { pdftrapped }
2169         { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2170         { \exp_not:n {#1} }
2171     }
2172 }

```

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

```

2173 \keys_define:nn { hyp / setup }
2174 {
2175     pdfinfo .code:n =
2176     {
2177         \keys_set:nn { hyp / info } { #1 }
2178     }

```

```
2179 }
```

Now we set some default values

```
2180 \keys_set:nn { hyp / setup} {pdfcreator = LaTeX-with-hyperref}
2181 \keys_set:nn { hyp / setup} {pdfauthor  = }
2182 \keys_set:nn { hyp / setup} {pdftitle   = }
2183 \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:

```
2184 \clist_map_inline:nn
2185 {
2186   ,pdfcopyright
2187   ,pdftype
2188   ,pdflicenseurl
2189   ,pdfauthortitle
2190   ,pdfcaptionwriter
2191   ,pdfmetalang
2192   ,pdfapart
2193   ,pdfaconformance
2194   ,pdfuapart
2195   ,pdfxstandard
2196   ,pdfsource
2197   ,pdfdocumentid
2198   ,pdfinstanceid
2199   ,pdfversionid
2200   ,pdfrendition
2201   ,pdfpublication
2202   ,pdfpubtype
2203   ,pdfbytes
2204   ,pdfnumpages
2205   ,pdfissn
2206   ,pdfeissn
2207   ,pdfisbn
2208   ,pdfbookedition
2209   ,pdfpublisher
2210   ,pdfvolumenum
2211   ,pdfissuenum
2212   ,pdfpagerange
2213   ,pdfdoi
2214   ,pdfurl
2215   ,pdfidentifier
2216   ,pdfsubtitle
2217   ,pdfpubstatus
2218   ,pdfcontactaddress
2219   ,pdfcontactcity
2220   ,pdfcontactregion
2221   ,pdfcontactpostcode
2222   ,pdfcontactcountry
2223   ,pdfcontactphone
2224   ,pdfcontactemail
```



```

2225     ,pdfcontacturl
2226     ,pdfdate
2227   }
2228   {
2229     \keys_define:nn { hyp / setup }
2230     {
2231       #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2232     }
2233   }

```

12.5 Transitions

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

```

2234 \keys_define:nn { hyp / setup }
2235 {
2236   pdfpageduration .code:n =
2237   {
2238     \tl_if_blank:nTF { #1 }
2239     {
2240       \pdfmanagement_remove:nn {Page}{Dur}
2241     }
2242     {
2243       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2244     }
2245   }
2246 }

```

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

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

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

The keys allowed in the argument of `pdfpagetransition` are

```

2247 \keys_define:nn { hyp / setup }
2248 {
2249   pdfpagetransition .code:n =
2250   {
2251     \tl_if_blank:nTF {#1}
2252     {
2253       \pdfmanagement_remove:nn {Page}{Trans}
2254     }
2255     {
2256       \group_begin:
2257       \keys_set:nn { hyp / trans }{style=R,#1}
2258       \pdf_object_unnamed_write:nx { dict }
2259       {
2260         \pdfdict_use:n {l__hyp_page/Trans}
2261       }
2262       \pdfmanagement_add:nnx {Page}{Trans}{\pdf_object_ref_last:}
2263       \group_end:
2264     }
2265   }
2266 }
2267 \keys_define:nn { hyp / trans }
2268 {
2269   ,style .choices:nn =
2270   {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2271   { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2272   ,style / unknown .code:n =
2273   {
2274     \msg_warning:nnxxx { hyp } { unknown-choice }
2275     { trans / style }
2276     { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2277     { \exp_not:n {#1} }
2278   }
2279   ,duration .code:n =
2280   {
2281     \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}

```

```

2282     }
2283     ,direction .choices:nn =
2284     {H,V}
2285     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2286     ,direction .choices:nn =
2287     {0,90,180,270,315}
2288     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ #1 } }
2289     ,direction / None .code:n =
2290     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ /None } }
2291     ,direction / unknown .code:n =
2292     {
2293         \msg_warning:nnxxx { hyp } { unknown-choice }
2294         { trans / direction }
2295         {
2296             H~(horizontal,~only~Split,~Blinds),
2297             V~(vertical,~only~Split,~Blinds),
2298             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2299             90~(bottom~to~top,~only~Wipe),
2300             180~(right~to~left,~only~Wipe),
2301             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2302             315~(top~left~to~bottom,~only~Glitter),
2303             None~(only~Fly)
2304         }
2305         { \exp_not:n {#1} }
2306     }
2307     ,motion .choices:nn =
2308     {I,O}
2309     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2310     ,motion / unknown .code:n =
2311     {
2312         \msg_warning:nnxxx { hyp } { unknown-choice }
2313         { trans / motion }
2314         { I~(inwards) , O~(outwards) }
2315         { \exp_not:n {#1} }
2316     }
2317     ,scale .code:n =
2318     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2319     ,opaque .choices:nn = {true,false}
2320     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2321     ,opaque / unknown .code:n =
2322     {
2323         \msg_warning:nnxxx { hyp } { unknown-choice }
2324         { trans / B }
2325         { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2326         { \exp_not:n {#1} }
2327     }
2328     % try to set unknown keys as style
2329     ,unknown .code:n =
2330     {
2331         % warning ...
2332         \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2333     }
2334 }

```

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

```

2335 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}
    Unfinished Form field code
2336 \NewDocumentCommand \MakeFieldObject { m m }
2337 {
2338   \pdfxform_new:nnn { #2 }{} { #1 }
2339 }
2340
2341
2342 \prop_new:N \g__hyp_AcroForm_CoFields_prop
2343 \prop_new:N \g__hyp_AcroForm_Fields_prop
2344
2345 \let\HyField@afields\ltx@empty
2346 \let\HyField@cofields\ltx@empty
2347 %% UF test for old pdftex removed
2348 %\let\HyField@AuxAddToFields\ltx@gobble
2349 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2350 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2351
2352 % the value doesn't matter, but with a prop we avoid duplicates and it is
2353 % clearly faster than removing them from a sequence
2354 \def\HyField@AuxAddToFields#1
2355 {
2356   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2357 }%
2358
2359 %fields with empty key get a value too -- lets hope that
2360 %this give the expected behaviour
2361 \def\HyField@AuxAddToCoFields #1 #2
2362 {
2363   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2364 }
2365
2366 \Hy@AtBeginDocument
2367 {
2368   \if@filesw
2369     \immediate\write\@mainaux{%
2370       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2371     }%
2372     \immediate\write\@mainaux{%
2373       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2374     }%
2375   \fi
2376   \let\HyField@AfterAuxOpen\@firstofone
2377 }%
2378
2379 \def\HyField@AddToFields
2380 {
2381   \exp_args:Nx\HyField__hypAddToFields
2382   {
2383     \pdfannot_box_ref_last:
2384   }
2385   \ifx\Fld@calculate@code\ltx@empty

```

```

2386 \else
2387 \begingroup
2388 \Hy@safe@activetrue
2389 \edef\Hy@temp{%
2390 \endgroup
2391 \if@filesw
2392 \write\@mainaux
2393 {
2394 \string\HyField@AuxAddToCoFields
2395 {
2396 \Fld@calculate@sortkey
2397 }
2398 {
2399 \pdfannot_box_ref_last:
2400 }
2401 }
2402 \fi
2403 }%
2404 \Hy@temp
2405 \fi
2406 }%
2407
2408 \def\HyField__hypAddToFields#1{
2409 \HyField@AfterAuxOpen{%
2410 \if@filesw
2411 \write\@mainaux{%
2412 \string\HyField@AuxAddToFields{#1}%
2413 }%
2414 \fi
2415 }%
2416 }%
2417
2418 \ExplSyntaxOff
2419 \ExplSyntaxOn
2420 \tl_new:N \l__hyp_CheckmarkYes_tl
2421 \tl_set:Nn \l__hyp_CheckmarkYes_tl { __hyp_xform_CheckMarkYes }
2422 \tl_new:N \l__hyp_CheckmarkOff_tl
2423 \tl_set:Nn \l__hyp_CheckmarkOff_tl { __hyp_xform_CheckMarkOff }
2424
2425 \def\@Form[#1]
2426 {
2427 \@ifundefined{textcolor}{\let\textcolor\@gobble}{ }
2428 \kvsetkeys{Form}{#1}
2429 \pdf@ifdraftmode{ }
2430 {
2431 \Hy@FormObjects
2432 \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2433 {
2434 \pdfmanagement_add:nxn { Catalog / AcroForm } { Fields }{##1}
2435 %\pdfmanagement_show:n { Catalog / AcroForm }
2436 }
2437 \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2438 {
2439 \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop

```

```

2440         {
2441             \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2442         }
2443     \seq_sort:Nn \l__hyp_tmpa_seq
2444     {
2445         \int_compare:nNnTF { \pdf@strcmp{##1}{##2} } > { 0 }
2446         { \sort_return_swapped: }
2447         { \sort_return_same: }
2448     }
2449     \seq_map_inline:Nn \l__hyp_tmpa_seq
2450     {
2451         \pdfmanagement_add:nxx { Catalog / AcroForm }
2452         { CO }
2453         {
2454             \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2455         }
2456     }
2457 }
2458 \pdfmanagement_add:nxx {Catalog / AcroForm/DR/Font }
2459 {ZaDb} {\pdf_object_ref:n {l__hyp_font_zapfdingbats_obj} }
2460 \pdfmanagement_add:nxx {Catalog / AcroForm/DR/Font }
2461 {Helv} {\pdf_object_ref:n {l__hyp_font_helvetica_obj} }
2462 \pdfmanagement_add:nxx {Catalog /AcroForm}
2463 {DA}{(/Helv~10~Tf~0~g)}
2464 \pdfmeta_standard_verify:nTF {form_no_NeedAppearance}
2465 {
2466     \legacy_if:nT { HyField@NeedAppearances }
2467     {
2468         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2469     }
2470 }
2471 {
2472     \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2473 }
2474 }
2475 \MakeFieldObject
2476 {
2477     \group_begin:
2478     \fontfamily{pzd}
2479     \fontencoding{U}
2480     \fontseries{m}
2481     \fontshape{n}
2482     \selectfont
2483     \char123
2484     \group_end:
2485 }
2486 {__hyp_xform_Ding}
2487 \MakeFieldObject
2488 {
2489     \group_begin:
2490     \fontfamily{pzd}
2491     \fontencoding{U}
2492     \fontseries{m}
2493     \fontshape{n}

```

```

2494         \selectfont
2495         \phantom{\char123}
2496         \group_end:
2497     }
2498     {__hyp_xform_DingOff}
2499 \MakeFieldObject
2500 {
2501     \group_begin:
2502     \fontfamily{pzd}
2503     \fontencoding{U}
2504     \fontseries{m}
2505     \fontshape{n}
2506     \selectfont
2507     \char51
2508     \group_end:
2509 }
2510 {__hyp_xform_CheckMarkYes}
2511 \MakeFieldObject
2512 {
2513     \group_begin:
2514     \fontfamily{pzd}
2515     \fontencoding{U}
2516     \fontseries{m}
2517     \fontshape{n}
2518     \selectfont
2519     \phantom{\char51} %perhaps xetex needs some small glyph ..
2520     \group_end:
2521 }
2522 {__hyp_xform_CheckMarkOff}
2523 \MakeFieldObject
2524 {
2525     \fbox{\textcolor{yellow}{\textsf{Submit}}}} %color?
2526 }
2527 {__hyp_xform_Submit}
2528 \MakeFieldObject
2529 {
2530     \fbox{\textcolor{yellow}{\textsf{SubmitP}}}} %color?
2531 }
2532 {__hyp_xform_SubmitP}
2533 }
2534 \ExplSyntaxOff
2535 \let\@endForm\ltx@empty
2536 \let\HyAnn@AbsPageLabel\ltx@empty
2537 \let\Fld@pageobjref\ltx@empty
2538
2539 \ExplSyntaxOn
2540 \newcount\HyAnn@Count
2541 \HyAnn@Count=\ltx@zero
2542 \def\HyAnn@AbsPageLabel
2543 {
2544     \global\advance\HyAnn@Count by\ltx@one
2545     %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2546     %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2547     %\zref@refused{HyAnn@\the\HyAnn@Count}%

```

```

2548 \__hyp_ref_label:en {HyAnn@the\HyAnn@Count}{abspage}
2549 \__hyp_ref_check:en {HyAnn@the\HyAnn@Count}{abspage}
2550 }%
2551 \def\Fld@pageobjref
2552 {
2553 \__hyp_ref_if_exist:enT {HyAnn@the\HyAnn@Count}{abspage}
2554 {
2555 /P~\pdf_pageobject_ref:n
2556 {
2557 \__hyp_ref_value:en{HyAnn@the\HyAnn@Count}{abspage}
2558 }
2559 }
2560 }
2561 \ExplSyntaxOff
2562 \ExplSyntaxOn
2563 %% check if the attr should be set through
2564 %% hooks.
2565 %% check if options are missing.
2566 \def\@TextField[#1]#2{% parameters, label
2567 \def\Fld@name{#2}%
2568 \let\Fld@default\ltx@empty
2569 \let\Fld@value\@empty
2570 \def\Fld@width{\DefaultWidthofText}%
2571 \def\Fld@height{%
2572 \ifFld@multiline
2573 \DefaultHeightofTextMultiline
2574 \else
2575 \DefaultHeightofText
2576 \fi
2577 }%
2578 \begingroup
2579 \expandafter\HyField@SetKeys\expandafter{%
2580 \DefaultOptionsofText,#1%
2581 }%
2582 \PDFForm@Name
2583 \HyField@FlagsText
2584 \ifFld@hidden\def\Fld@width{1sp}\fi
2585 \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2586 \LayoutTextField{#2}{%
2587 \leavevmode
2588 \HyAnn@AbsPageLabel
2589 \Hy@escapeform\PDFForm@Text
2590 \pdfannot_box:nnnn
2591 {\Fld@width}
2592 {\Fld@height}
2593 {Opt} %is this correct?
2594 {\PDFForm@Text}
2595 \MakeTextField{\Fld@width}{\Fld@height}
2596 \HyField@AddToFields
2597 }%
2598 \endgroup
2599 }
2600 \providecommand\@curropt{}
2601 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices

```



```

2602 \def\Fld@name{#2}
2603 \let\Fld@default\relax
2604 \let\Fld@value\relax
2605 \def\Fld@width{\DefaultWidthofChoiceMenu}
2606 \def\Fld@height{\DefaultHeightofChoiceMenu}
2607 \begingroup
2608   \Fld@menulength=0 %
2609   \@tempdima\z@
2610   \clist_map_variable:nNn { #3 } \@curropt
2611   %\@for\@curropt:=#3\do
2612   {%
2613     \expandafter\Fld@checkequals\@curropt==\\%
2614     \Hy@StepCount\Fld@menulength
2615     \settowidth{\@tempdimb}{\@currDisplay}%
2616     \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2617   }%
2618   \advance\@tempdima by~15\p@
2619   \begingroup
2620     \HyField@SetKeys{#1}
2621   \edef\x{\endgroup
2622     \noexpand\expandafter
2623     \noexpand\HyField@SetKeys
2624     \noexpand\expandafter{%
2625       \expandafter\noexpand\curname DefaultOptionsof%
2626       \ifFld@radio
2627         Radio%
2628       \else
2629         \ifFld@combo
2630           \ifFld@popdown
2631             PopdownBox%
2632           \else
2633             ComboBox%
2634           \fi
2635         \else
2636           ListBox%
2637         \fi
2638       \fi
2639     \endcurname
2640   }%
2641   }\x
2642   \HyField@SetKeys{#1}%
2643   \PDFForm@Name
2644   \ifFld@hidden\def\Fld@width{1sp}\fi
2645   \ifx\Fld@value\relax
2646     \let\Fld@value\Fld@default
2647   \fi
2648   \LayoutChoiceField{#2}{%
2649     \ifFld@radio
2650       \HyField@FlagsRadioButton
2651       \__hypRadio{#3}%
2652     \else
2653       \begingroup
2654         \HyField@FlagsChoice
2655         \ifdim\Fld@width<\@tempdima

```

```

2656         \ifdim\@tempdima<1cm\@tempdima1cm\fi
2657         \edef\Fld@width{\the\@tempdima}%
2658     \fi
2659     \ifFld@combo
2660     \else
2661         \@tempdima=\the\Fld@menulength\Fld@charsize
2662         \advance\@tempdima by~\Fld@borderwidth bp %
2663         \advance\@tempdima by~\Fld@borderwidth bp %
2664         \edef\Fld@height{\the\@tempdima}%
2665     \fi
2666     \__hypListBox{#3}%
2667 \endgroup
2668 \fi
2669 }%
2670 \endgroup
2671 }
2672 \tl_new:N \l__hyp_RadioYes_tl
2673 \tl_set:Nn \l__hyp_RadioYes_tl { __hyp_xform_Ding }
2674 \def\__hypRadio#1{%
2675     \Fld@listcount=0-%
2676     %\show\Fld@default
2677     \EdefEscapeName\Fld@default{\Fld@default}%
2678     \clist_map_variable:nNn { #1 } \@curropt
2679     %\@for\@curropt:=#1\do
2680     {%
2681         \expandafter\Fld@checkequals\@curropt==\\%
2682         \EdefEscapeName\@currValue{\@currValue}%
2683         \Hy@StepCount\Fld@listcount
2684         \@currDisplay\space
2685         \leavevmode
2686         \HyAnn@AbsPageLabel
2687         \Hy@escapeform\PDFForm@Radio
2688         \pdfannot_box:nnnn
2689             {\Fld@width}
2690             {\Fld@height}
2691             {Opt} %is this correct?
2692             {
2693                 \PDFForm@Radio
2694                 /AP
2695                 <<
2696                 /N
2697                 <<
2698                 /\@currValue\c_space_tl \pdfxform_ref:o {__hyp_xform_Ding}
2699                 %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2700                 >>
2701                 >>
2702             }
2703         {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2704         \int_compare:nNnT { \Fld@listcount } = { 1 }
2705         { \HyField@AddToFields }
2706         \c_space_tl % deliberate space between radio buttons
2707             % to do: --> should be configurable
2708     }%
2709 }

```

```

2710
2711 \newcount\Fld@listcount
2712 \def\__hypListBox#1
2713 {
2714   \HyField@PDFChoices{#1}
2715   \mode_leave_vertical:
2716   \HyAnn@AbsPageLabel
2717   \Hy@escapeform\PDFForm@List
2718   \pdf_link_user:nnn
2719     {widget} %perhaps we need more types??
2720     {\PDFForm@List}
2721     {\MakeChoiceField{\Fld@width}{\Fld@height}}
2722   \HyField@AddToFields
2723 }
2724
2725
2726 \def\@PushButton[#1]#2{% parameters, label
2727   \def\Fld@name{#2}%
2728   \group_begin:
2729     \exp_args:No\HyField@SetKeys
2730     {
2731       \DefaultOptionsofPushButton,#1
2732     }
2733     \PDFForm@Name
2734     \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2735     {
2736       \HyField@FlagsPushButton
2737       \legacy_if:nT {Fld@hidden}
2738       {
2739         \def\Fld@width{1sp}
2740       }
2741       \LayoutPushButtonField
2742       {
2743         \mode_leave_vertical:
2744         \HyAnn@AbsPageLabel
2745         \Hy@escapeform\PDFForm@Push
2746         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2747         \pdfannot_box:nnnn
2748           {\box_wd:N\l_tmpa_box}
2749           {\box_ht:N\l_tmpa_box}
2750           {\box_dp:N\l_tmpa_box} %is this correct?
2751           {\PDFForm@Push}
2752           {\box_use:N\l_tmpa_box}
2753         \HyField@AddToFields
2754       }
2755     }
2756     {
2757       \msg_error:nn { hyp }{ pdfa-no-push-button }
2758       \LayoutPushButtonField
2759       {
2760         \mode_leave_vertical:
2761         \MakeButtonField{#2}
2762       }
2763     }

```

```

2764 \group_end:
2765 }
2766
2767 \def\@Submit[#1]#2
2768 {
2769   \def\Fld@width {\DefaultWidthofSubmit}
2770   \def\Fld@height{\DefaultHeightofSubmit}
2771   \group_begin:
2772     \exp_args:No\HyField@SetKeys
2773     {
2774       \DefaultOptionsofSubmit,#1
2775     }
2776     \HyField@FlagsPushButton
2777     \HyField@FlagsSubmit
2778     \legacy_if:nT { Fld@hidden }
2779     {
2780       \def\Fld@width{1sp}
2781     }
2782     \mode_leave_vertical:
2783     \HyAnn@AbsPageLabel
2784     \Hy@escapeform\PDFForm@Submit
2785     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2786     \pdfannot_box:nnnn
2787       {\box_wd:N\l_tmpa_box}
2788       {\box_ht:N\l_tmpa_box}
2789       {\box_dp:N\l_tmpa_box} %is this correct?
2790     {
2791       \PDFForm@Submit
2792       /AP<<
2793         /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2794         /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2795       >>
2796     }
2797     \HyField@AddToFields
2798     \box_use:N\l_tmpa_box
2799
2800   \group_end:
2801 }
2802
2803 \def\@Reset[#1]#2
2804 {
2805   \def\Fld@width {\DefaultWidthofReset}
2806   \def\Fld@height{\DefaultHeightofReset}
2807   \group_begin:
2808     \exp_args:No\HyField@SetKeys
2809     {
2810       \DefaultOptionsofReset,#1
2811     }
2812     \mode_leave_vertical:
2813     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2814     {
2815       \HyField@FlagsPushButton
2816       \legacy_if:nT { Fld@hidden }
2817       { \def\Fld@width{1sp} }

```

```

2818 \HyAnn@AbsPageLabel
2819 \Hy@escapeform\PDFForm@Reset
2820 \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2821 \pdfannot_box:nnnn
2822   {\box_wd:N\l_tmpa_box}
2823   {\box_ht:N\l_tmpa_box}
2824   {\box_dp:N\l_tmpa_box} %is this correct?
2825   { \PDFForm@Reset }
2826 \HyField@AddToFields
2827 \box_use:N \l_tmpa_box
2828 }
2829 {
2830   \msg_error:nn { hyp }{ pdfa-no-reset-button }
2831   \MakeButtonField{#2}
2832 }
2833 \group_end:
2834 }
2835
2836 \def\@CheckBox[#1]#2
2837   {% parameters, label
2838   \def\Fld@name{#2}
2839   \def\Fld@default{0}
2840   \group_begin:
2841     \def\Fld@width {\DefaultWidthofCheckBox}
2842     \def\Fld@height{\DefaultHeightofCheckBox}
2843     \exp_args:No\HyField@SetKeys
2844     {
2845       \DefaultOptionsofCheckBox,#1
2846     }
2847     \PDFForm@Name
2848     \HyField@FlagsCheckBox
2849     \legacy_if:nT { Fld@hidden }
2850     {
2851       \def\Fld@width{1sp}
2852     }
2853     \LayoutCheckField{#2}
2854     {
2855       \mode_leave_vertical:
2856       \HyAnn@AbsPageLabel
2857       \Hy@escapeform\PDFForm@Check
2858       \pdfannot_box:nnnn
2859       {\Fld@width}
2860       {\Fld@height}
2861       {Opt} %is this correct?
2862       {\PDFForm@Check}
2863       \HyField@AddToFields %check if this works with xelatex ...
2864     }
2865   \group_end:
2866 }
2867 \ExplSyntaxOff
2868
2869 %hm. Should a luatex driver use type1 fonts in fields????
2870 \ExplSyntaxOn
2871 \def\Hy@FormObjects

```

```

2872 {
2873   \pdf_object_new:nn {l__hyp_encoding_pdfdoc_obj } { dict }
2874   \pdf_object_new:nn {l__hyp_font_zapfdingbats_obj } { dict }
2875   \pdf_object_new:nn {l__hyp_font_helvetica_obj } { dict }
2876   \pdf_object_write:nx {l__hyp_encoding_pdfdoc_obj }
2877   {
2878     /Type/Encoding
2879     /Differences[
2880       24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2881       /ring/tilde
2882       \c_space_tl
2883       39/quotesingle
2884       \c_space_tl
2885       96/grave %
2886       \iow_newline:
2887       128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2888       /fraction/guilsinglleft/guilsinglright/minus/perthousand
2889       /quotedblbase/quotedblleft/quotedblright/quoteleft
2890       /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2891       /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2892       \iow_newline:
2893       164/currency
2894       \c_space_tl
2895       166/brokenbar
2896       \c_space_tl
2897       168/dieresis/copyright/ordfeminine
2898       \c_space_tl
2899       172/logicalnot/.notdef/registered/macron/degree/plusminus
2900       /twosuperior/threesuperior/acute/mu
2901       \c_space_tl
2902       183/periodcentered/cedilla/onesuperior/ordmasculine
2903       \c_space_tl
2904       188/onequarter/onehalf/threequarters
2905       \iow_newline:
2906       192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
2907       /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
2908       /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
2909       /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
2910       /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
2911       /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
2912       /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
2913       /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
2914       /ocircumflex/otilde/odieresis/divide/oshlash/ugrave
2915       /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
2916     ]
2917   }
2918   \pdf_object_write:nn {l__hyp_font_zapfdingbats_obj }
2919   {
2920     /Type/Font
2921     /Subtype/Type1
2922     /Name/ZaDb
2923     /BaseFont/ZapfDingbats
2924   }
2925   \pdf_object_write:nx {l__hyp_font_helvetica_obj }

```

```

2926     {
2927         /Type/Font
2928         /Subtype/Type1
2929         /Name/Helv
2930         /BaseFont/Helvetica
2931         /Encoding~\pdf_object_ref:n { l__hyp_encoding_pdfdoc_obj }
2932     }
2933     \global\let\Hy@FormObjects\relax
2934 }
2935 \ExplSyntaxOff
2936 \providecommand*{\Fld@pageobjref}{}
2937 \ifcsname pdf@escapestring\endcsname
2938     \def\Hy@escapeform#1{%
2939         \ifHy@pdfescapeform
2940             \let\Hy@escapestring\pdfescapestring
2941         \else
2942             \let\Hy@escapestring\@firstofone
2943         \fi
2944     }%
2945     \Hy@escapeform{}%
2946 \else
2947     \let\Hy@escapestring\@firstofone
2948     \def\Hy@escapeform#1{%
2949         \ifHy@pdfescapeform
2950             \def\Hy@escapestring##1{%
2951                 \noexpand\Hy@escapestring{\noexpand##1}%
2952             }%
2953             \edef\Hy@temp{#1}%
2954             \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{}\@nil
2955             \def\Hy@escapestring##1{%
2956                 \@ifundefined{Hy@esc@\string##1}{%
2957                     ##1%
2958                     \ThisShouldNotHappen
2959                 }{%
2960                     \csname Hy@esc@\string##1\endcsname
2961                 }%
2962             }%
2963         \else
2964             \let\Hy@escapestring\@firstofone
2965         \fi
2966     }%
2967     \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
2968         \ifx\#3\%
2969             \else
2970                 \expandafter
2971                 \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
2972                 \ltx@ReturnAfterFi{%
2973                     \Hy__hypescapeform#3\@nil
2974                 }%
2975             \fi
2976         }%
2977     \fi
2978     \def\PDFForm@Name{%
2979         \PDFForm__hypName\Fld@name

```

```

2980 \ifx\Fld@altname\relax
2981 \else
2982   \PDFForm__hypName\Fld@altname
2983 \fi
2984 \ifx\Fld@mappingname\relax
2985 \else
2986   \PDFForm__hypName\Fld@mappingname
2987 \fi
2988 }
2989 \def\PDFForm__hypName#1{%
2990   \begingroup
2991     \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
2992       \ifHy@unicode
2993         \Hy@unicodedefalse
2994       \fi
2995     \fi
2996     \pdfstringdef\Hy@gtemp#1%
2997   \endgroup
2998   \let#1\Hy@gtemp
2999 }
3000 \def\Fld@X@additionalactions{%
3001   \ifx\Fld@keystroke@code\@empty
3002   \else
3003     /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3004   \fi
3005   \ifx\Fld@format@code\@empty
3006   \else
3007     /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3008   \fi
3009   \ifx\Fld@validate@code\@empty
3010   \else
3011     /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3012   \fi
3013   \ifx\Fld@calculate@code\@empty
3014   \else
3015     /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3016   \fi
3017   \ifx\Fld@onfocus@code\@empty
3018   \else
3019     /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3020   \fi
3021   \ifx\Fld@onblur@code\@empty
3022   \else
3023     /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3024   \fi
3025   \ifx\Fld@onmousedown@code\@empty
3026   \else
3027     /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3028   \fi
3029   \ifx\Fld@onmouseup@code\@empty
3030   \else
3031     /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3032   \fi
3033   \ifx\Fld@onenter@code\@empty

```



```

3034 \else
3035 /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3036 \fi
3037 \ifx\Fld@onexit@code\@empty
3038 \else
3039 /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3040 \fi
3041 }
3042 \ExplSyntaxOn
3043 \def\Fld@additionalactions
3044 {%
3045 \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{ }
3046 {
3047 \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3048 {/AA<<\Fld@X@additionalactions>>}
3049 }
3050 }
3051 \ExplSyntaxOff
3052 \def\Fld@annotnames{%
3053 /T(\Fld@name)%
3054 \ifx\Fld@altname\relax
3055 \else
3056 /TU(\Fld@altname)%
3057 \fi
3058 \ifx\Fld@mappingname\relax
3059 \else
3060 /TM(\Fld@mappingname)%
3061 \fi
3062 }
3063 \ExplSyntaxOn
3064 \def\PDFForm@Check
3065 {
3066 /Subtype/Widget
3067 ~\Fld@annotflags
3068 ~\Fld@pageobjref
3069 ~\Fld@annotnames
3070 /FT/Btn
3071 \Fld@flags
3072 /Q~\Fld@align
3073 /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3074 /AP
3075 <<
3076 /N
3077 <<
3078 /Yes~\pdfxform_ref:o{\l__hyp_CheckmarkYes_tl}
3079 /Off~\pdfxform_ref:o{\l__hyp_CheckmarkOff_tl}
3080 >>
3081 >>
3082 /MK<<
3083 \int_compare:nNnF {\Fld@rotation}={0}
3084 {
3085 /R~\Fld@rotation
3086 }
3087 \tl_if_empty:NF\Fld@bordercolor

```

```

3088     {
3089         /BC[\Fld@bordercolor]
3090     }
3091     \tl_if_empty:NF\Fld@bcolor
3092     {
3093         /BG[\Fld@bcolor]
3094     }
3095     /CA(\Hy@escapestring{\Fld@cbsymbol})%
3096     >>
3097     /DA
3098     (
3099         /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3100         \tl_if_empty:NF \Fld@color
3101         {
3102             \c_space_tl \Fld@color
3103         }
3104     )
3105     /H/P
3106     \legacy_if:nTF {Fld@checked}
3107     {
3108         /V/Yes /AS/Yes
3109     }
3110     {
3111         /V/Off /AS/Off
3112     }
3113     \Fld@additionalactions
3114 }
3115 \ExplSyntaxOff
3116 \ExplSyntaxOn
3117 \def\PDFForm@Push
3118 {
3119     /Subtype/Widget
3120     ~\Fld@annotflags
3121     ~\Fld@pageobjref
3122     ~\Fld@annotnames
3123     /FT/Btn
3124     ~\Fld@flags
3125     /H/P
3126     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3127     \bool_if:nT
3128     {
3129         !\int_compare_p:nNn {\Fld@rotation} = {0}
3130         ||
3131         \tl_if_exist_p:N \Fld@bordercolor
3132     }
3133     {
3134         /MK
3135         <<
3136         \int_compare:nNnF {\Fld@rotation} = {0}
3137         {
3138             /R~\Fld@rotation
3139         }
3140         \tl_if_exist:NT \Fld@bordercolor
3141         {

```

```

3142         /BC[\Fld@bordercolor]
3143     }
3144     >>
3145 }
3146 /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3147 \Fld@additionalactions
3148 }
3149
3150 \ExplSyntaxOff
3151 \def\PDFForm@List{%
3152     /Subtype/Widget%
3153     \Fld@annotflags
3154     \Fld@pageobjref
3155     \Fld@annotnames
3156     /FT/Ch%
3157     \Fld@flags
3158     /Q \Fld@align
3159     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3160     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3161         \ifx\Fld@bordercolor\relax\else 1\fi
3162         \ifx\Fld@bcolor\relax \else 1\fi
3163         \space
3164     \else
3165         /MK<<%
3166         \ifnum\Fld@rotation=\z@
3167             \else
3168                 /R \Fld@rotation
3169             \fi
3170         \ifx\Fld@bordercolor\relax
3171             \else
3172                 /BC[\Fld@bordercolor]%
3173             \fi
3174         \ifx\Fld@bcolor\relax
3175             \else
3176                 /BG[\Fld@bcolor]%
3177             \fi
3178         >>%
3179     \fi
3180     /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3181         \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3182     \Fld@choices
3183     \Fld@additionalactions
3184 }
3185 \ExplSyntaxOn
3186 \def\PDFForm@Radio
3187 {
3188     /Subtype/Widget
3189     ~\Fld@annotflags
3190     ~\Fld@pageobjref
3191     ~\Fld@annotnames
3192     /FT/Btn
3193     \Fld@flags
3194     /H/P
3195     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>

```

```

3196 /MK<<
3197 \ifnum\Fld@rotation=\z@
3198 \else
3199 /R~\Fld@rotation
3200 \fi
3201 \ifx\Fld@bordercolor\relax
3202 \else
3203 /BC[\Fld@bordercolor]%
3204 \fi
3205 \ifx\Fld@bcolor\relax
3206 \else
3207 /BG[\Fld@bcolor]%
3208 \fi
3209 /CA(\Hy@escapestring{\Fld@radiosymbol})%
3210 >>
3211 /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3212 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3213 \ifx\Fld@default\@empty
3214 /V/Off%
3215 /DV/Off%
3216 \else
3217 /V/\Fld@default
3218 /DV/\Fld@default
3219 \fi
3220 \Fld@additionalactions
3221 }
3222 \ExplSyntaxOff
3223 \ExplSyntaxOn
3224 % Does an appeareance dict make sense here?
3225 \def\PDFForm@Text
3226 {
3227 /Subtype/Widget
3228 ~\Fld@annotflags
3229 ~\Fld@pageobjref
3230 ~\Fld@annotnames
3231 /FT/Tx
3232 ~\Fld@flags
3233 /Q~\Fld@align
3234 /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3235 \bool_if:nT
3236 {
3237 !\int_compare_p:nNn {\Fld@rotation} = {0}
3238 ||
3239 \tl_if_exist_p:N \Fld@bordercolor
3240 ||
3241 \tl_if_exist_p:N \Fld@bcolor
3242 }
3243 {
3244 /MK
3245 <<
3246 \int_compare:nNnF {\Fld@rotation} = {0}
3247 {
3248 /R~\Fld@rotation
3249 }

```

```

3250         \tl_if_exist:NT \Fld@bordercolor
3251         {
3252             /BC[\Fld@bordercolor]
3253         }
3254         \tl_if_exist:NT \Fld@bcolor
3255         {
3256             /BG[\Fld@bcolor]
3257         }
3258     >>
3259 }
3260 /DA
3261 (
3262     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3263     \tl_if_empty:NF {\c_space_tl\Fld@color}
3264 )
3265 /DV(\Hy@escapestring{\Fld@default})
3266 /V(\Hy@escapestring{\Fld@value})
3267 ~\Fld@additionalactions
3268 \int_compare:nNnT { \Fld@maxlen}>{0}
3269 {
3270     /MaxLen~\Fld@maxlen
3271 }
3272 }
3273 \ExplSyntaxOff
3274
3275 \def\PDFForm@Submit{%
3276     /Subtype/Widget%
3277     \Fld@annotflags
3278     \Fld@pageobjref
3279     \Fld@annotnames
3280     /FT/Btn%
3281     \Fld@flags
3282     /H/P%
3283     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3284     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3285         \ifx\Fld@bordercolor\relax\else 1\fi
3286         \space
3287     \else
3288         /MK<<%
3289         \ifnum\Fld@rotation=\z@
3290         \else
3291             /R \Fld@rotation
3292         \fi
3293         \ifx\Fld@bordercolor\relax
3294         \else
3295             /BC[\Fld@bordercolor]%
3296         \fi
3297     >>%
3298     \fi
3299     /A<<%
3300     /S/SubmitForm%
3301     /F<<%
3302     /FS/URL%
3303     /F(\Hy@escapestring{\Form@action})%

```

```

3304     >>%
3305     \Fld@submitflags
3306     >>%
3307     \Fld@additionalactions
3308 }
3309 \ExplSyntaxOn
3310 \def\PDFForm@Reset{%
3311     /Subtype/Widget%
3312     \Fld@annotflags
3313     \Fld@pageobjref
3314     \Fld@annotnames
3315     /FT/Btn%
3316     \Fld@flags
3317     /H/P%
3318     /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3319     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3320         \ifx\Fld@bordercolor\relax\else 1\fi
3321         \space
3322     \else
3323         /MK<<%
3324         \ifnum\Fld@rotation=\z@
3325         \else
3326             /R~\Fld@rotation
3327         \fi
3328         \ifx\Fld@bordercolor\relax
3329         \else
3330             /BC[\Fld@bordercolor]%
3331         \fi
3332         >>%
3333     \fi
3334     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3335     /A<</S/ResetForm>>%
3336     \Fld@additionalactions
3337 }%
3338
3339
3340 %these patterns are used in hyperref checks.
3341 %it is unclear if they are really useful and if a backend support is
3342 %needed.
3343 \str_case:VnF \c_sys_backend_str
3344 {
3345     { pdfmode }
3346     {
3347         \def\HyPat@ObjRef
3348         {
3349             [0-9]*[1-9][0-9]*~0~R
3350         }
3351     }
3352     { dvipdfmx }
3353     {
3354         \def\HyPat@ObjRef
3355         {
3356             @[\~]+
3357         }

```

```

3358 }
3359 { xdvipdfmx }
3360 {
3361   \def\HyPat@ObjRef
3362   {
3363     @[\~]+
3364   }
3365 }
3366 }
3367 { %also set in hyperref sty, so probably not needed.
3368   \def\HyPat@ObjRef/{.+}
3369 }
3370
3371
3372 \ExplSyntaxOff
3373 % UF: removed Hy@writebookmark
3374 %   \Hy@currentbookmarklevel{0}
3375 %   \Hy@numberline
3376 %   \__hypwritetorep
3377 %   counter{bookmark@seq@number}
3378 % removed \HyPsd@SanitizeForOutFile, not needed
3379 % removed \currentpdfbookmark, defined by bookmark,
3380 % should use \newcommand there
3381 % removed \subpdfbookmark, defined by bookmark,
3382 % should use \newcommand there
3383 % removed \belowpdfbookmark, defined by bookmark,
3384 % should use \newcommand there
3385 % removed \pdfbookmark, defined by bookmark,
3386 % \BOOKMARK
3387 % \@BOOKMARK
3388 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3389 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3390 %% removed \ReadBookmarks / unneeded with bookmark.
3391 %% removed \Hy@OutlineName
3392 %% removed \check@bm@number
3393 %% removed \calc@bm@number
3394
3395 \ifHy@implicit
3396 \else
3397   \expandafter\endinput
3398 \fi
3399 \newlength\Hy@SectionHShift
3400 \def\Hy@SectionAnchorHref#1{%
3401   \ifx\protect\@typeset@protect
3402     \Hy__hypSectionAnchor{#1}%
3403   \fi
3404 }
3405 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%
3406   \leavevmode
3407   \hbox to Opt{%
3408     \kern-\Hy@SectionHShift
3409     \Hy@raisedlink{%
3410       \hyper@anchorstart{#1}\hyper@anchorend
3411     }%

```

```

3412     \hss
3413   }%
3414 }
3415 \let\H@old@ssect\@ssect
3416 \def\@ssect#1#2#3#4#5{%
3417   \Hy@MakeCurrentHrefAuto{section*}%
3418   \setlength{\Hy@SectionHShift}{#1}%
3419   \begingroup
3420     \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3421     \toks\tw@expandafter{%
3422       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3423       #5%
3424     }%
3425   \edef\x{\endgroup
3426     \the\toks@{\the\toks\tw@}%
3427   }\x
3428 }
3429 \let\H@old@schapter\@schapter
3430 \def\@schapter#1{%
3431   \begingroup
3432     \let\@mkboth\@gobbletwo
3433     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3434     \Hy@raisedlink{%
3435       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3436     }%
3437   \endgroup
3438   \H@old@schapter{#1}%
3439 }
3440 \ltx@ifundefined{@chapter}{}{%
3441   \let\Hy@org@chapter\@chapter
3442   \def\@chapter{%
3443     \def\Hy@next{%
3444       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3445       \Hy@raisedlink{%
3446         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3447       }%
3448     }%
3449     \ifnum\c@secnumdepth>\m@ne
3450       \ltx@ifundefined{if@mainmatter}%
3451       \iftrue{\csname if@mainmatter\endcsname}%
3452       \let\Hy@next\relax
3453     \fi
3454     \fi
3455     \Hy@next
3456     \Hy@org@chapter
3457   }%
3458 }
3459 \let\H@old@part\@part
3460 \begingroup\expandafter\expandafter\expandafter\endgroup
3461 \expandafter\ifx\csname chapter\endcsname\relax
3462   \let\Hy@secnum@part\z@
3463 \else
3464   \let\Hy@secnum@part\m@ne
3465 \fi

```



```

3466 \def\@part{%
3467   \ifnum\Hy@secnum@part>\c@secnumdepth
3468     \phantomsection
3469   \fi
3470   \H@old@part
3471 }
3472 \let\H@old@spart\@spart
3473 \def\@spart#1{%
3474   \Hy@MakeCurrentHrefAuto{part*}%
3475   \Hy@raisedlink{%
3476     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3477   }%
3478   \H@old@spart{#1}%
3479 }
3480 \let\H@old@sect\@sect
3481 \def\@sect#1#2#3#4#5#6[#7]#8{%
3482   \ifnum #2>\c@secnumdepth
3483     \expandafter\@firstoftwo
3484   \else
3485     \expandafter\@secondoftwo
3486   \fi
3487   {%
3488     \Hy@MakeCurrentHrefAuto{section*}%
3489     \setlength{\Hy@SectionHShift}{#3}%
3490     \begingroup
3491       \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[\@currentHref]}%
3492       \toks\tw@{\expandafter\@firstoftwo\expandafter\@secondoftwo}%
3493       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3494       #8%
3495     }%
3496     \edef\x{\endgroup
3497       \the\toks@{\the\toks\tw@}}%
3498     }\x
3499   }{%
3500     \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[\@currentHref]{#8}%
3501   }%
3502 }
3503 \expandafter\def\csname Parent-4\endcsname{}
3504 \expandafter\def\csname Parent-3\endcsname{}
3505 \expandafter\def\csname Parent-2\endcsname{}
3506 \expandafter\def\csname Parent-1\endcsname{}
3507 \expandafter\def\csname Parent0\endcsname{}
3508 \expandafter\def\csname Parent1\endcsname{}
3509 \expandafter\def\csname Parent2\endcsname{}
3510 \expandafter\def\csname Parent3\endcsname{}
3511 \expandafter\def\csname Parent4\endcsname{}
3512 %%
3513 %% End of file 'hgeneric-testphase.def'.
3514 \endpackage
3515 \def\color@{
3516   % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3517   % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3518   % linkcolor=
3519   %,filecolor=

```

```

3520 %,urlcolor=
3521 %,menucolor=
3522 %,runcolor=
3523 %,linkbordercolor=
3524 %,filebordercolor=
3525 %,urlbordercolor=
3526 %,menubordercolor=
3527 %,runbordercolor=
3528
3529 \prop_const_from_keyval:Nn \c__hyp_colorscheme_original_prop
3530 {
3531     linkcolor      = [rgb]{1,0,0}, %red
3532     filecolor      = [rgb]{0,1,1}, %cyan
3533     urlcolor       = [rgb]{1,0,1}, %magenta
3534     menucolor      = [rgb]{1, 0, 0}, %red
3535     runcolor       = [rgb]{0,1,1}, %cyan
3536 %-----
3537     linkbordercolor = [rgb]{1, 0 ,0 },
3538     filebordercolor = [rgb]{0, .5, .5},
3539     urlbordercolor  = [rgb]{0, 1, 1},
3540     menubordercolor = [rgb]{1, 0, 0},
3541     runbordercolor  = [rgb]{0, .7, .7}
3542 }
3543
3544 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3545 {
3546     linkcolor      = [rgb]{0,0.2,0.6},
3547     filecolor      = [rgb]{0.8,0,0.8},
3548     urlcolor       = [rgb]{0.8,0,0.8},
3549     menucolor      = [rgb]{0,0.2,0.6},
3550     runcolor       = [rgb]{0.8,0,0.8},
3551 %----- %-----
3552     linkbordercolor = [rgb]{0,0.2,0.6},
3553     filebordercolor = [rgb]{0.8,0,0.8},
3554     urlbordercolor  = [rgb]{0.8,0,0.8},
3555     menubordercolor = [rgb]{0,0.2,0.6},
3556     runbordercolor  = [rgb]{0.8,0,0.8}
3557 }
3558
3559 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3560 { %two colors: intern/extern
3561     linkcolor      = [rgb]{0.79216, 0, 0.12549},
3562     filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3563     urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3564     menucolor      = [rgb]{0.79216, 0, 0.12549 },
3565     runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3566 %----- %-----
3567     linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3568     filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3569     urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3570     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3571     runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3572 }
3573

```

```

3574 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3575 { %all darkgray
3576   linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3577   filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3578   urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3579   menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3580   runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3581   %----- %-----
3582   linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3583   filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3584   urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3585   menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3586   runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3587 }
3588
3589 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3590 { %dvipsnam.def
3591   linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3592   filecolor      = [rgb]{1, 0, 0}, %Red
3593   urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3594   menucolor      = [rgb]{1, 0, 0}, %Red
3595   runcolor       = [rgb]{1, 0, 0}, %Red
3596   %----- %-----
3597   linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3598   filebordercolor = [rgb]{1, 0, 0}, %Red
3599   urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3600   menubordercolor = [rgb]{1, 0, 0}, %Red
3601   runbordercolor  = [rgb]{1, 0, 0} %Red
3602 }
3603
3604 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3605 { %dvipsnam.def
3606   linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3607   filecolor      = [rgb]{0, 1, 0}, %Green
3608   urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3609   menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3610   runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3611   %----- %-----
3612   linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3613   filebordercolor = [rgb]{0, 1, 0}, %Green
3614   urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3615   menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3616   runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3617 }
3618
3619
3620 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3621 {
3622   linkcolor      = [rgb]{0.50196, 0, 0.02353},
3623   filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3624   urlcolor       = [rgb]{0.54118, 0, 0.52941},
3625   menucolor      = [rgb]{0.44706, 0.45882, 0},
3626   runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3627   %----- %-----

```

```

3628     linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3629     filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3630     urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3631     menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3632     runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3633 }
3634
3635 \prop_const_from_keyval:Nn \c_hyp_colorscheme_henryford_prop
3636 {
3637     linkcolor      = [rgb]{0,0,0},
3638     filecolor      = [rgb]{0,0,0},
3639     urlcolor       = [rgb]{0,0,0},
3640     menucolor      = [rgb]{0,0,0},
3641     runcolor       = [rgb]{0,0,0},
3642     %----- %-----
3643     linkbordercolor = [rgb]{0,0,0},
3644     filebordercolor = [rgb]{0,0,0},
3645     urlbordercolor  = [rgb]{0,0,0},
3646     menubordercolor = [rgb]{0,0,0},
3647     runbordercolor  = [rgb]{0,0,0}
3648 }
3649 </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	
\#	637
\%	638
\.	455, 459, 461
@curropt commands:	
\@curropt:	2611, 2679
\\	17, 18, 19, 26, 30, 40, 60, 67, 74, 81, 88, 103, 108, 109, 117, 118, 127, 128, 129, 130, 131, 138, 767, 2613, 2681, 2968
_	453, 455, 459, 461
A	
\A	453
\Acrobatmenu	17, 155
\addcontentsline	12
\AddToDocumentProperties	329
\AddToHook	354
\advance	2544, 2618, 2662, 2663
allcolor (hypersetup key)	896
\author	2
B	
\b	457
\begingroup	183, 237, 273, 2387, 2578, 2607, 2619, 2653, 2990, 3419, 3431, 3460, 3490
\belowpdfbookmark	3383
\bgroup	247
\BOOKMARK	3386
bookmarkstype (hypersetup key)	12
bool commands:	
\bool_if:N	240, 255, 303, 586, 601, 612, 625, 633, 663, 721, 778, 866, 876, 1116, 1127
\bool_if:N	733, 1823, 1848, 3127, 3235
\bool_lazy_and:nnTF	349
\bool_lazy_or:nnTF	1146, 1195
\bool_new:N	177, 178, 439, 443, 447
\bool_set_true:N	448
bordercolormodel (hypersetup key)	12, 917
box commands:	
\box_dp:N	529, 2750, 2789, 2824
\box_ht:N	528, 2749, 2788, 2823
\box_new:N	388, 450
\box_set_dp:Nn	2020
\box_set_ht:Nn	2027
\box_use:N	1133, 2752, 2798, 2827
\box_use_drop:N	1138
\box_wd:N	527, 2748, 2787, 2822
\l_tmpa_box	2746, 2748, 2749, 2750, 2752, 2785, 2787, 2788, 2789, 2798, 2820, 2822, 2823, 2824, 2827
C	
\char	2483, 2495, 2507, 2519
\chardef	136
\cite	30
clist commands:	
\clist_map_function:nN	110, 119
\clist_map_inline:nn	2184
\clist_map_variable:nNn	2610, 2678
color commands:	
\color_export:nnN	816
\color_export:nnnN	822
\color_select:n	833, 869, 1137
\color_select:nn	839
\color_set:nn	2, 334, 850
\color_set:nnn	2, 333, 856
color names:	
hyp/annot/file	436
hyp/annot/link	436
hyp/annot/menu	436
hyp/annot/run	436
hyp/annot/url	436
colorfile (hypersetup key)	896
colorlink (hypersetup key)	896
colorlinks (hypersetup key)	882
colormenu (hypersetup key)	896
colorrund (hypersetup key)	896
colorscheme (hypersetup key)	1, 1304
colorurl (hypersetup key)	896
cs commands:	
\cs_generate_variant:Nn	134, 135, 331, 367, 382, 387, 491, 825, 842, 859
\cs_gset:Npn	1108
\cs_if_exist_p:N	350
\cs_new:Npn	103, 383
\cs_new_protected:Npn	183, 264, 279, 292, 313, 326, 361, 370, 467, 471, 475, 479, 492, 494, 498, 540, 546, 553, 557, 579, 584, 599, 610, 623, 661, 719, 776, 809, 820, 826, 837, 843, 854, 1052, 2055, 2111
\cs_set_eq:NN	
\cs_set_eq:NN	139, 148, 149, 189, 231, 637, 638, 1381

<code>\cs_set_protected:Npn</code>	137, 162, 508, 767	<code>\else</code>	2386, 2574, 2628, 2632, 2635, 2652, 2660, 2941, 2946, 2963, 2969, 2981, 2985, 3002, 3006, 3010, 3014, 3018, 3022, 3026, 3030, 3034, 3038, 3055, 3059, 3160, 3161, 3162, 3164, 3167, 3171, 3175, 3181, 3198, 3202, 3206, 3212, 3216, 3284, 3285, 3287, 3290, 3294, 3319, 3320, 3322, 3325, 3329, 3396, 3463, 3484
<code>\csname</code>	2625, 2960, 2971, 3451, 3461, 3503, 3504, 3505, 3506, 3507, 3508, 3509, 3510, 3511	<code>\endcsname</code>	2639, 2937, 2960, 2971, 3451, 3461, 3503, 3504, 3505, 3506, 3507, 3508, 3509, 3510, 3511
<code>\currentpdfbookmark</code>	3379	<code>\endgroup</code> ..	281, 2390, 2598, 2621, 2667, 2670, 2997, 3425, 3437, 3460, 3496
D			
<code>\d</code>	455, 459, 461	<code>\endinput</code>	3397
<code>debug</code> (hypersetup key)	<u>1328</u>	exp commands:	
<code>\DeclareDocumentMetadata</code>	3, 10, 13, 14, 17, 19, 27, 33	<code>\exp_args:Ne</code>	3045
<code>\DeclareRobustCommand</code>	236, 251, 271, 284, 298, 3405	<code>\exp_args:Nne</code>	2149
<code>\def</code>	143, 503, 2350, 2354, 2361, 2379, 2408, 2425, 2542, 2551, 2566, 2567, 2570, 2571, 2584, 2585, 2601, 2602, 2605, 2606, 2644, 2674, 2712, 2726, 2727, 2739, 2767, 2769, 2770, 2780, 2803, 2805, 2806, 2817, 2836, 2838, 2839, 2841, 2842, 2851, 2871, 2938, 2948, 2950, 2955, 2967, 2978, 2989, 3000, 3043, 3052, 3064, 3117, 3151, 3186, 3225, 3275, 3310, 3347, 3354, 3361, 3368, 3400, 3416, 3430, 3442, 3443, 3466, 3473, 3481, 3503, 3504, 3505, 3506, 3507, 3508, 3509, 3510, 3511	<code>\exp_args:Nno</code>	281, 294
<code>\DefaultHeightofCheckBox</code>	2842	<code>\exp_args:Nno</code>	266, 315, 2097
<code>\DefaultHeightofChoiceMenu</code>	2606	<code>\exp_args:Nnox</code>	671
<code>\DefaultHeightofReset</code>	2806	<code>\exp_args:NNV</code>	1808, 1839
<code>\DefaultHeightofSubmit</code>	2770	<code>\exp_args:NnV</code>	542, 549
<code>\DefaultHeightofText</code>	2575	<code>\exp_args:Nnx</code>	2332
<code>\DefaultHeightofTextMultiline</code> ...	2573	<code>\exp_args:No</code>	571, 575, 2729, 2772, 2808, 2843
<code>\DefaultOptionsofCheckBox</code>	2845	<code>\exp_args:NV</code>	525, 532
<code>\DefaultOptionsofPushButton</code>	2731	<code>\exp_args:Nx</code> ...	667, 2090, 2138, 2381
<code>\DefaultOptionsofReset</code>	2810	<code>\exp_not:N</code>	767
<code>\DefaultOptionsofSubmit</code>	2774	<code>\exp_not:n</code>	1243, 1277, 1388, 1448, 1469, 1506, 1526, 1549, 1593, 1626, 1651, 1702, 1727, 1754, 1803, 1880, 1908, 1935, 1955, 2170, 2277, 2305, 2315, 2326
<code>\DefaultOptionsofText</code>	2580	<code>\expandafter</code>	2579, 2613, 2622, 2624, 2625, 2681, 2954, 2970, 3397, 3421, 3422, 3460, 3461, 3483, 3485, 3492, 3493, 3503, 3504, 3505, 3506, 3507, 3508, 3509, 3510, 3511
<code>\DefaultWidthofCheckBox</code>	2841	<code>\ExplSyntaxOff</code>	9, 2418, 2534, 2561, 2867, 2935, 3051, 3115, 3150, 3222, 3273, 3372
<code>\DefaultWidthofChoiceMenu</code>	2605	<code>\ExplSyntaxOn</code>	7, 10, 2419, 2539, 2562, 2870, 3042, 3063, 3116, 3185, 3223, 3309
<code>\DefaultWidthofReset</code>	2805	extension (hypersetup key)	
<code>\DefaultWidthofSubmit</code>	2769	F	
<code>\DefaultWidthofText</code>	2570	<code>\fbox</code>	2525, 2530, 2703
<code>destlabel</code> (hypersetup key)	<u>12</u>	<code>\fi</code>	2375, 2402, 2405, 2414, 2576, 2584, 2585, 2616, 2634, 2637, 2638, 2644, 2647, 2656, 2658, 2665, 2668, 2943, 2965, 2975, 2977, 2983, 2987, 2994, 2995,
dim commands:			
<code>\dim_to_decimal_in_bp:n</code>	11, 139		
<code>\do</code>	2611, 2679		
<code>\documentclass</code>	20		
<code>draft</code> (hypersetup key)	<u>1328</u>		
E			
<code>\edef</code>	2389, 2621, 2657, 2664, 2953, 3425, 3496		
<code>\EdefEscapeName</code>	2677, 2682		

__hyp_colormodel_select_aux:wn .	\l__hyp_href_url_protocol_tl . . .
..... 826, 830, 837 179, 190, 229, 266, 317
__hyp_colormodel_set_aux:nwn . . .	__hyp_link_goto_begin:nw
..... 843, 847, 854 557, 591, 606
\c__hyp_colorscheme_daleif_prop 3544	__hyp_link_goto_end: . . 579, 592, 614
\c__hyp_colorscheme_henryford_-	\c__hyp_map_annot_hyp_prop 400
prop 3635	\c__hyp_map_hyp_annot_prop . . 400,
\c__hyp_colorscheme_julian_prop 3559	860, 924, 960, 987, 995, 1006, 1033,
\c__hyp_colorscheme_original_-	1041, 1110, 1218, 1254, 1264, 1396
prop 3529	__hyp_ocg_init: 1052, 1052, 1108, 1118
\c__hyp_colorscheme_phelype_prop	__hyp_PageLabels_gpsh: 498, 498, 511
..... 3620	\l__hyp_para_tmpa_tl 396, 732, 735, 744
\c__hyp_colorscheme_szabolcsA_-	\l__hyp_para_tmpa_tl_uuu\l__hyp_-
prop 3589	text_tmpa_str_uuu\g__hyp_text_-
\c__hyp_colorscheme_szabolcsB_-	tmpa_str 393
prop 3604	\l__hyp_RadioYes_tl 2672, 2673
\c__hyp_colorscheme_tivv_prop . . 3574	__hyp_ref_check:nn 360, 370, 382, 2549
\l__hyp_dest_box 29,	__hyp_ref_if_exist:nn . 360, 368, 369
450, 527, 528, 529, 2004, 2020, 2027	__hyp_ref_if_exist:nnTF . . 372, 2553
\l__hyp_dest_name_tmpa_tl . . 393,	__hyp_ref_label:nn 360, 361, 367, 2548
560, 561, 567, 571, 573, 576, 684, 697	__hyp_ref_value:nn 360, 383, 387, 2557
\l__hyp_dest_pdfremotestartview_-	__hyp_secondoftwoewithopt:wnn . . .
tl 419, 692, 1811, 1815 322, 323, 324, 325
\g__hyp_dest_pdfstartpage_tl . . .	__hyp_setup_info_date_key:nn . . .
..... 419, 1822, 1824, 1831, 1849, 1856 2111, 2143, 2144
\g__hyp_dest_pdfstartview_tl . . .	__hyp_setup_info_key:nn . . 2055,
..... 419, 1824, 1831, 1842, 1846, 1849, 1856	2104, 2105, 2106, 2107, 2109, 2110
\l__hyp_dest_pdfview_tl	__hyp_store_metadata:nn
..... 435, 543, 550, 326, 331, 2052,
1973, 1976, 1983, 1986, 1987, 1988,	2072, 2090, 2124, 2138, 2159, 2231
1989, 1990, 1991, 1996, 2000, 2036	__hyp_text_cleanup:N . . 471, 471, 485
\c__hyp_dest_startview_regex . . .	\l__hyp_text_enc_dest_print_tl . .
..... 451, 1809, 1840 422, 683
\c__hyp_dest_undefined_tl	\l__hyp_text_enc_dest_tl
..... 399, 566, 567 29, 422, 521, 572
__hyp_destination:nn	\l__hyp_text_enc_file_print_tl . .
..... 29, 514, 514, 543, 550 422, 726
\l__hyp_filename_tmpa_tl . . . 393,	\l__hyp_text_enc_info_print_tl . .
666, 668, 674, 675, 680, 727, 728, 744 422, 496
__hyp_href_pdf_aux:nn 276, 279	\l__hyp_text_enc_para_print_tl . .
\l__hyp_href_pdf_destination_tl 422, 731
..... 180, 191, 230, 281	\l__hyp_text_enc_uri_print_tl . . .
\l__hyp_href_pdf_page_tl 181, 199, 691 242,
__hyp_href_run_aux:nn 289, 292	245, 257, 260, 305, 308, 422, 630, 1407
\l__hyp_href_run_parameter_tl . . .	__hyp_text_pdfstring:nnN
..... 182, 203, 294 479, 479, 491, 493,
__hyp_href_url_aux:n 310, 313	496, 519, 570, 628, 681, 724, 729, 1407
__hyp_href_url_aux:nn 262, 264	__hyp_text_pdfstring_info:nN . . .
\l__hyp_href_url_encode_bool 494, 494, 2066, 2084, 2094
..... 177, 188, 226, 240, 255, 303	__hyp_text_purify:nN . . 467, 467, 484
__hyp_href_url_format	__hyp_text_string_from_unicode:nN
..... 183, 189, 231, 316 475, 475, 486
\l__hyp_href_url_ismap_bool	\g__hyp_text_tmpa_str . . 398, 487, 489
..... 178, 202, 633	

\l__hyp_text_tmpa_str	linkfileprefix	1350
..... 397, 484, 485, 486, 487	linktoc	1350
\l__hyp_tmpa_box 388, 1120, 1133, 1138	linktocpage	1350
\l__hyp_tmpa_int	localanchorname	1350
..... 388	menu	9, 1396
\l__hyp_tmpa_seq	menuborder	13
..... 388, 1809,	menuborderstyle	13, 1006
1811, 1840, 1842, 1963, 1964, 1968,	menucolor	896
1970, 1994, 2001, 2002, 2003, 2012,	naturalnames	1350
2014, 2024, 2029, 2441, 2443, 2449	nesting	13
\l__hyp_tmpa_str	ocgcolorfile	1146
..... 388, 2066, 2067, 2069,	ocgcolorlink	1146
2084, 2085, 2087, 2094, 2095, 2099	ocgcolorlinks	1146
\l__hyp_tmpa_tl 388, 522, 526, 534,	ocgcolormenu	1146
937, 941, 1407, 1408, 1413, 1807,	ocgcolorrun	1146
1809, 1838, 1840, 1970, 1971, 1978	ocgcolorurl	1146
\l__hyp_uri_tmpa_tl ... 393, 631, 632	pageanchor	1350
hyp/annot/file (color name)	pdfauthor	2055
..... 436	pdfborder	13
hyp/annot/link (color name)	pdfborderstyle	13, 1006
..... 436	pdfcreationdate	13, 2111
hyp/annot/menu (color name)	pdfcreator	2055
..... 436	pdfencoding	1318
hyp/annot/run (color name)	pdfinfo	2173
..... 436	pdfkeywords	2055
hyp/annot/url (color name)	pdflang	13, 2041
..... 436	pdflinkmargin	13
hyp/text/pdfstring	pdfmoddate	13, 2111
..... 466	pdfproducer	2055
\hypercalcbp	pdfremotestartview	11
..... 11, 17, 139	pdfstartview	11
\HyperDestNameFilter	pdfsubject	2055
..... 520, 571	pdftitle	2055
\hypersetup 1, 2, 5, 8, 9, 12, 17, 64, 97, 162	pdftrapped	2145
\hypersetup keys:	pdfversion	1318
allcolor	pdfview	11, 1959
..... 896	plainpages	1350
bookmarkstyle	run	9, 1396
..... 12	runborder	13
bordercolormodel	runborderstyle	13, 1006
..... 12, 917	runcolor	896
colorfile	unicode	1318
..... 896	url	9, 1396
colorlink	urlborder	13
..... 896	urlborderstyle	13, 1006
colorlinks	urlcolor	896
..... 882	verbose	1328
colormenu	hypertextnames (hypersetup key) ...	1350
..... 896		
colorrun		
..... 896		
colorscheme		
..... 1, 1304		
colorurl		
..... 896		
debug		
..... 1328		
destlabel		
..... 12		
draft		
..... 1328		
extension		
..... 12, 1350		
file		
..... 9, 1396		
fileborderstyle		
..... 13, 1006		
filecolor		
..... 896		
final		
..... 1328		
hidefile		
..... 1280		
hidelink		
..... 1280		
hidelinks		
..... 1280		
hidemenu		
..... 1280		
hiderun		
..... 1280		
hideurl		
..... 1280		
hypertextnames		
..... 1350		
link		
..... 9, 1396		
linkborder		
..... 13		
linkborderstyle		
..... 13, 1006		
linkcolor		
..... 896		

I

\ifcase	3160, 3284, 3319
\ifcsname	2937
\ifdim	2616, 2655, 2656

`\ifnum` . . . 2991, 3160, 3166, 3197, 3284,
 3289, 3319, 3324, 3449, 3467, 3482
`\iftrue` 3451
`\ifx` 2385, 2585, 2645,
 2968, 2980, 2984, 3001, 3005, 3009,
 3013, 3017, 3021, 3025, 3029, 3033,
 3037, 3054, 3058, 3161, 3162, 3170,
 3174, 3181, 3201, 3205, 3212, 3213,
 3285, 3293, 3320, 3328, 3401, 3461
`\immediate` 2369, 2372
int commands:
 `\int_compare:nNnTF` . . . 1968, 1994,
 2445, 2704, 3083, 3136, 3246, 3268
 `\int_compare_p:nNn` 3129, 3237
 `\int_eval:n` 690
 `\int_max:nn` 691
 `\int_new:N` 391
int internal commands:
 `__hypListbox` 2666, 2712
 `__hypRadio` 2651, 2674
 `__hypwritetorep` 3376
iow commands:
 `\iow_newline:` 2886, 2892, 2905

K

`\kern` 3408
keys commands:
 `\keys_define:nn`
 168, 186, 224, 335, 882, 898,
 905, 917, 926, 947, 962, 981, 1008,
 1027, 1150, 1164, 1176, 1199, 1205,
 1220, 1249, 1280, 1293, 1304, 1318,
 1328, 1337, 1350, 1377, 1398, 1403,
 1959, 2041, 2057, 2075, 2113, 2127,
 2145, 2173, 2229, 2234, 2247, 2267
 `\l_keys_key_str` 173, 2099, 2332
 `\keys_set:nn`
 . 165, 194, 213, 238, 253, 274, 287,
 301, 343, 1310, 1317, 2150, 2177,
 2180, 2181, 2182, 2183, 2257, 2332
 `\keys_set_known:nn` 2335
`\kvsetkeys` 164, 2428

L

`\label` 8, 12
`\LayoutCheckField` 2853
`\LayoutChoiceField` 2648
`\LayoutPushButtonField` 2741, 2758
`\LayoutTextField` 2586
`\leavevmode` 2587, 2685, 3406
legacy commands:
 `\legacy_if:nTF` 150, 352, 356,
 506, 2466, 2737, 2778, 2816, 2849, 3106

`\let` 146, 239, 254,
 275, 288, 302, 323, 324, 325, 596,
 646, 653, 707, 713, 757, 763, 806,
 2345, 2346, 2348, 2349, 2376, 2427,
 2535, 2536, 2537, 2568, 2569, 2603,
 2604, 2646, 2933, 2940, 2942, 2947,
 2964, 2998, 3415, 3429, 3432, 3441,
 3452, 3459, 3462, 3464, 3472, 3480
`link` (hypersetup key) 9, 1396
`linkborder` (hypersetup key) 13
`linkborderstyle` (hypersetup key) 13, 1006
`linkcolor` (hypersetup key) 896
`linkfileprefix` (hypersetup key) 1350
`linktoc` (hypersetup key) 1350
`linktocpage` (hypersetup key) 1350
`localanchorname` (hypersetup key) 1350

M

`\MakeButtonField`
 2746, 2761, 2785, 2820, 2831
`\MakeChoiceField` 2721
`\MakeFieldObject` 2336,
 2475, 2487, 2499, 2511, 2523, 2528
`\MakeRadioField` 2703
`\MakeTextField` 2595
`\mbox` 1130
`menu` (hypersetup key) 9, 1396
`menuborder` (hypersetup key) 13
`menuborderstyle` (hypersetup key) 13, 1006
`menucolor` (hypersetup key) 896
mode commands:
 `\mode_if_horizontal:TF` 516, 538
 `\mode_leave_vertical:`
 559, 640, 699, 746,
 783, 2715, 2743, 2760, 2782, 2812, 2855
msg commands:
 `\msg_error:nn` 351, 2757, 2830
 `\msg_line_context:` 68
 `\g_msg_module_name_prop` 11
 `\msg_new:nnn` 36, 43, 48, 52,
 56, 63, 70, 77, 84, 91, 97, 104, 113, 123
 `\msg_new:nnnn` 12, 22
 `\msg_warning:nn` 152, 1324
 `\msg_warning:nnn` 161, 563, 801
 `\msg_warning:nnnn`
 1168, 1209, 1446, 1490,
 1524, 1547, 1609, 1638, 1672, 1687,
 1700, 1713, 1738, 1771, 1787, 1814,
 1845, 1878, 1892, 1919, 1953, 2035
 `\msg_warning:nnnnn` 172,
 1240, 1274, 1385, 1466, 1503, 1590,
 1623, 1648, 1724, 1751, 1800, 1905,
 1932, 2167, 2274, 2293, 2312, 2323

N	
naturalnames (hypersetup key)	1350
nesting (hypersetup key)	13
\newcommand	142 , 3380 , 3382 , 3384
\newcount	2540 , 2711
\NewDocumentCommand	2336
\NewExpandableDocumentCommand	322
\newlength	3399
\noexpand	2622 , 2623 , 2624 , 2625 , 2951
\nolinkurl	3 , 4
O	
ocgcolorfile (hypersetup key)	1146
ocgcolorlink (hypersetup key)	1146
ocgcolorlinks (hypersetup key)	1146
ocgcolormenu (hypersetup key)	1146
ocgcolorrun (hypersetup key)	1146
ocgcolorurl (hypersetup key)	1146
P	
page/Trans internal commands:	
l_hyp_page/Trans	464
pageanchor (hypersetup key)	1350
\paperwidth	3
\PassOptionsToPackage	358 , 1342 , 1347
pdf commands:	
\pdf_bdcobject:nn	1132 , 1135
\pdf_destination:nn	24 , 134 , 533
\pdf_destination:nnnn	526
\pdf_emc:	1134 , 1140
\pdf_link_user:nnn	2718
\pdf_name_from_unicode_e:n	219 , 670 , 785
\pdf_object_if_exist:nTF	668
\pdf_object_new:nn	1054 , 1055 , 1056 , 1057 , 2873 , 2874 , 2875
\pdf_object_ref:n	135 , 680 , 1060 , 1062 , 1084 , 1085 , 1088 , 1092 , 1097 , 1102 , 1107 , 2459 , 2461 , 2931
\pdf_object_ref_last:	701 , 2262
\pdf_object_unnamed_write:nn	700 , 2258
\pdf_object_write:nn	1058 , 1064 , 1074 , 1086 , 2876 , 2918 , 2925
\pdf_pageobject_ref:n	1831 , 1856 , 2555
\pdf_string_from_unicode:nnN	477
\pdf_version:	1494 , 1613 , 1642 , 1676 , 1691 , 1717 , 1742 , 1775 , 1791 , 1896 , 1923
\pdf_version_compare:NnTF	1484 , 1575 , 1597 , 1633 , 1666 , 1681 , 1707 , 1732 , 1758 , 1781 , 1886 , 1913
\pdf_version_compare_p:Nn	737 , 1147 , 1196
\pdf_version_major:	149 , 1148 , 1171 , 1197 , 1213
\pdf_version_minor:	148 , 1171 , 1213
pdfannot commands:	
\pdfannot_box:nnnn	2590 , 2688 , 2747 , 2786 , 2821 , 2858
\pdfannot_box_ref_last:	2383 , 2399
\pdfannot_dict_put:nnn	641 , 701 , 747 , 786 , 938 , 973 , 997 , 1019 , 1043 , 1225 , 1256
\pdfannot_dict_remove:nn	932 , 968 , 989 , 1014 , 1035 , 1233 , 1266
\pdfannot_link:nnn	642 , 702 , 748 , 787
\pdfannot_link_goto_begin:nw	576
\pdfannot_link_goto_end:	581
\pdfannot_link_margin:n	8 , 1528
\c_pdfannot_link_types_seq	445
pdfauthor (hypersetup key)	2055
\pdfbookmark	3385
pdfborder (hypersetup key)	13
pdfborderstyle (hypersetup key)	13 , 1006
pdfcreationdate (hypersetup key)	13 , 2111
pdfcreator (hypersetup key)	2055
\pdfdest	11
pdfdict commands:	
\pdfdict_new:n	464 , 619 , 656 , 715 , 772
\pdfdict_put:nnn	211 , 218 , 465 , 620 , 621 , 632 , 635 , 657 , 658 , 670 , 677 , 687 , 697 , 716 , 717 , 728 , 743 , 773 , 774 , 784 , 1554 , 1555 , 1559 , 1560 , 2271 , 2281 , 2285 , 2288 , 2290 , 2309 , 2318 , 2320
\pdfdict_remove:nn	208 , 740 , 1564 , 1565
\pdfdict_use:n	641 , 700 , 747 , 753 , 786 , 791 , 2260
pdfencoding (hypersetup key)	1318
\pdfescapestring	2940
pdffile commands:	
\pdffile_embed_file:nnn	672
PDFForm internal commands:	
\PDFForm_hypName	2979 , 2982 , 2986 , 2989
pdfinfo (hypersetup key)	2173
pdfkeywords (hypersetup key)	2055
pdflang (hypersetup key)	13 , 2041
pdflinkmargin (hypersetup key)	13
pdfmanagement commands:	
\pdfmanagement_add:nn	1537
\pdfmanagement_add:nnn	500 , 1084 , 1085 , 1107 , 1413 , 1438 , 1454 , 1458 , 1478 , 1486 , 1515 , 1570 , 1580 , 1604 , 1618 , 1630 , 1635 , 1660 , 1668 , 1683 , 1709 , 1734 , 1766 , 1783 , 1829 , 1854 , 1868 ,

1888, 1915, 1944, 2050, 2069, 2087, 2098, 2122, 2136, 2154, 2243, 2262, 2434, 2451, 2458, 2460, 2462, 2468	\providecommand
\pdfmanagement_if_active_p: . . . 350	140, 141, 2370, 2373, 2600, 2936
\pdfmanagement_remove:nn	\ProvidesFile 3
. . . 1410, 1434, 1442, 1462, 1474, 1499, 1511, 1519, 1533, 1542, 1586, 1601, 1620, 1657, 1696, 1721, 1747, 1762, 1796, 1826, 1851, 1864, 1873, 1901, 1928, 1940, 1949, 2047, 2063, 2081, 2119, 2133, 2240, 2253, 2472	R
\pdfmanagement_show:n 2435	\ReadBookmarks 3390
pdfmeta commands:	ref commands:
\pdfmeta_standard_verify:nnTF . . .	\ref_if_exist:nn 368
. 781, 2734, 2813	\ref_label:nn 364
\pdfmeta_standard_verify:nTF . . .	\ref_value:nn 22, 385
. 2464, 3047	regex commands:
pdfmoddate (hypersetup key) 13, 2111	\regex_const:Nn 451
pdfproducer (hypersetup key) 2055	\regex_extract_once:NnNTF 1809, 1840
pdfremotestartview (hypersetup key) . . 11	\relax 596, 646, 653, 707, 713, 757, 763, 806, 2603, 2604, 2645, 2933, 2980, 2984, 3054, 3058, 3161, 3162, 3170, 3174, 3201, 3205, 3285, 3293, 3320, 3328, 3452, 3461
pdfstartview (hypersetup key) 11	\RemoveFromHook 1418
\pdfstringdef 8, 9, 17, 26, 2996	\RenewDocumentCommand 155
\pdfstringdefDisableCommands	\RequirePackage 6, 18, 354, 3388
. 323, 324, 325	run (hypersetup key) 9, 1396
pdfsubject (hypersetup key) 2055	runborder (hypersetup key) 13
pdftitle (hypersetup key) 2055	runborderstyle (hypersetup key) . 13, 1006
pdftrapped (hypersetup key) 2145	runcolor (hypersetup key) 896
pdfversion (hypersetup key) 1318	S
pdfview (hypersetup key) 11, 1959	\selectfont 2482, 2494, 2506, 2518
pdfxform commands:	seq commands:
\pdfxform_new:nnn 2338	\seq_const_from_clist:Nn 400
\pdfxform_ref:n	\seq_count:N 1968, 1994
. . . 2698, 2699, 2793, 2794, 3078, 3079	\seq_get_right:NN 1970
\phantom 2495, 2519	\seq_item:Nn 1811, 1842, 1964, 2012, 2014, 2024, 2029
\phantomsection 3468	\seq_map_inline:Nn
plainpages (hypersetup key) 1350	. 437, 441, 445, 896, 1193, 1291, 2449
prg commands:	\seq_new:N 390
\prg_generate_conditional_-	\seq_put_right:Nn
variant:Nnn 369 2001, 2002, 2003, 2441
\prg_new_eq_conditional:NNn . . . 368	\seq_set_split:Nnn 1963
prop commands:	\seq_sort:Nn 2443
\prop_const_from_keyval:Nn	\setlength 3418, 3489
. 402, 410, 3529, 3544, 3559, 3574, 3589, 3604, 3620, 3635	\setpdflinkmargin 8
\prop_gput:Nnn 11, 2356, 2363	\settowidth 2615
\prop_if_empty:Nn 2437	\show 2676
\prop_item:Nn 2454	sort commands:
\prop_map_inline:Nn	\sort_return_same: 2447
. 860, 924, 960, 987, 995, 1006, 1033, 1041, 1110, 1218, 1254, 1264, 1308, 1396, 2432, 2439	\sort_return_swapped: 2446
\prop_new:N 2342, 2343	\space 378, 2684, 3163, 3180, 3181, 3211, 3212, 3286, 3318, 3321
\protect 239, 254, 275, 288, 302, 374, 596, 646, 653, 707, 713, 757, 763, 806, 3401	\spacefactor 516, 538
	\stockwidth 3
	str commands:
	\c_colon_str 767

<code>\c_hash_str</code>	637	<code>\ifundefined</code>	2427, 2956
<code>\c_percent_str</code>	638	<code>\@latex@warning</code>	375
<code>\str_case:nnTF</code>	3343	<code>\@mainaux</code>	2369, 2372, 2392, 2411
<code>\str_case_e:nnTF</code>	1964	<code>\@mkboth</code>	3432
<code>\str_gset:Nn</code>	921	<code>\@nil</code>	2954, 2967, 2973
<code>\str_gset_eq:NN</code>	487	<code>\@part</code>	3459, 3466
<code>\str_head:n</code>	2156, 2161	<code>\@pdfauthor</code>	21
<code>\str_if_eq:nnTF</code>	523, 2067, 2085, 2095, 3045	<code>\@pdfborder</code>	140
<code>\str_if_eq_p:nn</code>	735, 1148, 1197	<code>\@pdfborderstyle</code>	141
<code>\str_lowercase:n</code>	1964, 2090, 2138, 2157, 2162	<code>\@savsf</code>	516, 538
<code>\str_new:N</code>	392, 397, 398, 436	<code>\@schapter</code>	3429, 3430
<code>\str_set:Nn</code>	469	<code>\@secondoftwo</code>	3485
<code>\str_set_convert:Nnnn</code>	8	<code>\@sect</code>	3480, 3481
<code>\str_set_eq:NN</code>	489	<code>\@spart</code>	3472, 3473
<code>\str_tail:n</code>	2157, 2162	<code>\@ssect</code>	3415, 3416
<code>\str_uppercase:n</code>	2150, 2156, 2161	<code>\@tempdima</code>	2609, 2616, 2618, 2655, 2656, 2657, 2661, 2662, 2663, 2664
<code>\string</code>	2370, 2373, 2394, 2412, 2956, 2960, 2971	<code>\@tempdimb</code>	2615, 2616
<code>\subpdfbookmark</code>	3381	<code>\@typeset@protect</code>	3401
sys commands:		<code>\@unexpandable@protect</code>	239, 254, 275, 288, 302
<code>\c_sys_backend_str</code>	3343	<code>\c@secnumdepth</code>	3449, 3467, 3482
T			
TeX and L ^A T _E X 2 _ε commands:			
<code>\@BOOKMARK</code>	3387	<code>\Fld@align</code>	3072, 3158, 3233
<code>\@CheckBox</code>	2836	<code>\Fld@altname</code> ..	2980, 2982, 3054, 3056
<code>\@ChoiceMenu</code>	2601	<code>\Fld@annotflags</code>	3067, 3120, 3153, 3189, 3228, 3277, 3312
<code>\@Form</code>	2425	<code>\Fld@annotnames</code>	3052, 3069, 3122, 3155, 3191, 3230, 3279, 3314
<code>\@PushButton</code>	2726	<code>\Fld@bcolor</code>	3091, 3093, 3174, 3176, 3205, 3207, 3241, 3254, 3256
<code>\@Reset</code>	2803	<code>\fld@bcolor</code>	3162
<code>\@Submit</code>	2767	<code>\Fld@bordercolor</code>	3087, 3089, 3131, 3140, 3142, 3161, 3170, 3172, 3201, 3203, 3239, 3250, 3252, 3285, 3293, 3295, 3320, 3328, 3330
<code>\@TextField</code>	2566	<code>\Fld@borderstyle</code>	3073, 3126, 3159, 3195, 3234, 3283, 3334
<code>\@bookmarksopenlevel</code>	1429	<code>\Fld@borderwidth</code> ..	2662, 2663, 3073, 3126, 3159, 3195, 3234, 3283, 3334
<code>\@bsphack</code>	363	<code>\Fld@calculate@code</code> ..	2385, 3013, 3015
<code>\@chapter</code>	3441, 3442	<code>\Fld@calculate@sortkey</code>	2396
<code>\@currDisplay</code>	2615, 2684	<code>\Fld@cbsymbol</code>	3095
<code>\@currValue</code>	2682, 2698	<code>\Fld@charsize</code>	2661, 3099, 3180, 3211, 3262, 3318
<code>\@currentHref</code>	3422, 3435, 3446, 3476, 3493	<code>\Fld@checkequals</code>	2613, 2681
<code>\@curropt</code> ..	2600, 2610, 2613, 2678, 2681	<code>\Fld@choices</code>	3182
<code>\@empty</code>	2569, 2585, 3001, 3005, 3009, 3013, 3017, 3021, 3025, 3029, 3033, 3037, 3181, 3212, 3213	<code>\Fld@color</code> ..	3100, 3102, 3181, 3212, 3263
<code>\@endForm</code>	2535	<code>\Fld@default</code>	2568, 2585, 2603, 2646, 2676, 2677, 2839, 3213, 3217, 3218, 3265
<code>\@esphack</code>	365		
<code>\@firstofone</code> ..	2376, 2942, 2947, 2964		
<code>\@firstoftwo</code>	3483		
<code>\@for</code>	2611, 2679		
<code>\@gobble</code>	2427		
<code>\@gobbletwo</code>	3432		
<code>\@hyper@launch</code>	34, 767		
<code>\@ifnextchar</code>	247		

\Fld@flags	3071,	\Hy@bookmarkstype	1430
3124, 3157, 3193, 3232, 3281, 3316		\Hy@chapapp	3433, 3444
\Fld@format@code	3005, 3007	\Hy@colorlink	31
\Fld@height	2571, 2592, 2595, 2606, 2664, 2690,	\Hy@currentbookmarklevel	3374
2703, 2721, 2770, 2806, 2842, 2860		\Hy@DisableOption	147, 154
\Fld@keystroke@code	3001, 3003	\Hy@drafttrue	1341
\Fld@listcount	2675, 2683, 2704, 2711	\Hy@escapeform	2589, 2687, 2717, 2745,
\Fld@mappingname	2984, 2986, 3058, 3060	2784, 2819, 2857, 2938, 2945, 2948	
\Fld@maxlen	3268, 3270	\Hy@escapestring	2940,
\Fld@menulength	2608, 2614, 2661	2942, 2947, 2950, 2951, 2954, 2955,	
\Fld@name	2567, 2602, 2727, 2838, 2979, 3053	2964, 2967, 3003, 3007, 3011, 3015,	
\Fld@onblur@code	3021, 3023	3019, 3023, 3027, 3031, 3035, 3039,	
\Fld@onclick@code	3146	3095, 3146, 3209, 3265, 3266, 3303	
\Fld@onenter@code	3033, 3035	\Hy@finaltrue	1346
\Fld@onexit@code	3037, 3039	\Hy@FormObjects	2431, 2871, 2933
\Fld@onfocus@code	3017, 3019	\Hy@gttemp	2996, 2998
\Fld@onmousedown@code	3025, 3027	\Hy@href	247
\Fld@onmouseup@code	3029, 3031	\Hy@href@nextactionraw	212
\Fld@pageobjref	2537, 2551, 2936, 3068,	\Hy@href@page	200
3121, 3154, 3190, 3229, 3278, 3313		\Hy@hypertextnamesfalse	1356
\Fld@radiosymbol	3209	\Hy@hypertextnamestrue	1355
\Fld@rotation	3083,	\Hy@linkfileprefix	1358
3085, 3129, 3136, 3138, 3160, 3166,		\Hy@linktoc	1381
3168, 3197, 3199, 3237, 3246, 3248,		\Hy@localanchornamefalse	1361
3284, 3289, 3291, 3319, 3324, 3326		\Hy@localanchornametrue	1360
\Fld@submitflags	3305	\Hy@MakeCurrentHrefAuto	3417, 3433, 3444, 3474, 3488
\Fld@validate@code	3009, 3011	\Hy@naturalnamesfalse	1365
\Fld@value	2569, 2585, 2604, 2645, 2646, 3266	\Hy@naturalnamestrue	1364
\Fld@width	2570,	\Hy@next	3443, 3452, 3455
2584, 2591, 2595, 2605, 2644, 2655,		\Hy@numberline	143, 3375
2657, 2689, 2703, 2721, 2739, 2769,		\Hy@org@chapter	3441, 3456
2780, 2805, 2817, 2841, 2851, 2859		\Hy@OutlineName	3391
\Fld@X@additionalactions	3000, 3045, 3048	\Hy@OutlineRerunCheck	3389
\Form@action	3303	\Hy@pageanchorfalse	1369
\G@refundefinedtrue	374	\Hy@pageanchortrue	1368
\H@old@part	3459, 3470	\Hy@pdfmajorversion	149
\H@old@schapter	3429, 3438	\Hy@pdfminorversion	148
\H@old@ssect	3480, 3491, 3500	\Hy@pdfstringtrue	27, 482
\H@old@spart	3472, 3478	\Hy@pdfversion	2991
\H@old@ssect	3415, 3420	\Hy@plainpagesfalse	1373
\href@	247	\Hy@plainpagetrue	1372
\Hy@abspage	510	\Hy@pstringdef	492, 2971
\Hy@activeanchorfalse	555	\Hy@PutCatalog	498
\Hy@activeanchortrue	548	\Hy@raisedlink	3409, 3434, 3445, 3475
\Hy@AtBeginDocument	2350, 2366	\Hy@RestoreLastskip	537
\Hy@bookmarksnumberedfalse	1422	\Hy@safe@activetrue	639, 2388
\Hy@bookmarksnumberedtrue	1423	\Hy@SaveLastskip	517
\Hy@bookmarksopenfalse	1426	\Hy@secnum@part	3462, 3464, 3467
\Hy@bookmarksopentrue	1427	\Hy@SectionAnchorHref	3400, 3422, 3493
		\Hy@SectionHShift	3399, 3408, 3418, 3489
		\Hy@StepCount	2614, 2683

\Hy@temp	2389, 2404, 2953, 2954	\HyPsd@pdfencoding@unicode	146
\Hy@unicodedefalse	2993	\HyPsd@SanitizeForOutFile	3378
\Hy@unicodetrue	145	\if@files	2368, 2391, 2410
\Hy@VerboseAnchor	518	\ifFld@combo	2629, 2659
\Hy@verbosefalse	1332	\ifFld@hidden	2584, 2644
\Hy@VerboseLinkStart	588, 603	\ifFld@multiline	2572
\Hy@VerboseLinkStop	594, 616, 649, 709, 759, 797	\ifFld@popdown	2630
\Hy@verbosetrue	1331	\ifFld@radio	2626, 2649
\Hy@VersionChecked	136	\ifHy@implicit	3395
\Hy@WrapperDef	514	\ifHy@pdfescapeform	2939, 2949
\Hy@xspace@end	591, 648, 708, 758, 796	\ifHy@unicode	2992
\HyAnn@AbsPageLabel	2536, 2542, 2588, 2686, 2716, 2744, 2783, 2818, 2856	\kv@set@family@handler	160
\HyAnn@Count	2540, 2541, 2544, 2545, 2546, 2547, 2548, 2549, 2553, 2557	\ltx@empty	2345, 2346, 2385, 2535, 2536, 2537, 2568
\HyField@AddToFields	2379, 2596, 2705, 2722, 2753, 2797, 2826, 2863	\ltx@gobble	2348
\HyField@afields	2345	\ltx@gobbletwo	2349
\HyField@AfterAuxOpen	2350, 2376, 2409	\ltx@ifUndefined	3440, 3450
\HyField@AuxAddToCoFields	2349, 2361, 2373, 2394	\ltx@one	2544
\HyField@AuxAddToFields	2348, 2354, 2370, 2412	\ltx@ReturnAfterFi	2972
\HyField@cofields	2346	\ltx@zero	2541
\HyField@FlagsCheckBox	2848	\m@ne	3449, 3464
\HyField@FlagsChoice	2654	\OBJ@OCG@view	142
\HyField@FlagsPushButton	2736, 2776, 2815	\p@	2618
\HyField@FlagsRadioButton	2650	\pdf@ifdraftmode	2429
\HyField@FlagsSubmit	2777	\PDF@SetupDoc	137
\HyField@FlagsText	2583	\pdf@strcmp	2445
\HyField@PDFChoices	2714	\PDFForm@Check	2857, 2862, 3064
\HyField@SetKeys	2579, 2620, 2623, 2642, 2729, 2772, 2808, 2843	\PDFForm@List	2717, 2720, 3151
\HyPat@ObjRef	3347, 3354, 3361, 3368	\PDFForm@Name	2582, 2643, 2733, 2847, 2978
\hyper@anchor	540	\PDFForm@Push	2745, 2751, 3117
\hyper@anchorend	540, 3410, 3435, 3446, 3476	\PDFForm@Radio	2687, 2693, 3186
\hyper@anchorstart	540, 3410, 3435, 3446, 3476	\PDFForm@Reset	2819, 2825, 3310
\hyper@link	31, 31, 584	\PDFForm@Submit	2784, 2791, 3275
\hyper@linkend	30, 31, 610	\PDFForm@Text	2589, 2594, 3225
\hyper@linkfile	281, 661	\protected@edef	560
\hyper@linklaunch	34, 294, 719, 770	\strip@pt	3099, 3180, 3211, 3262, 3318
\hyper@linknamed	35, 157, 776	\toks@	3420, 3426, 3491, 3497
\hyper@linkstart	30, 31, 599	\tw@	3421, 3426, 3492, 3497
\hyper@linkurl	266, 316, 623	\XR@ext	1352
\hyper@normalise	247, 262, 276, 289, 310	\z@	2609, 3160, 3166, 3197, 3284, 3289, 3319, 3324, 3462
\HyPL@Labels	500, 510	\zref@labelbylist	2546
\HyPL@StorePageLabel	498	\zref@labelbyprops	2545
\HyPsd@LoadUnicode	144	\zref@refused	2547
\HyPsd@pdfencoding	146	\texorpdfstring	27
		text commands:	
		\text_expand:n	666
		\text_purify:n	8, 469
		\textcolor	2427, 2525, 2530
		\textsf	2525, 2530
		\the	510, 2545, 2546, 2547, 2548, 2549, 2553, 2557, 2657, 2661, 2664, 3426, 3497

<code>\thepage</code>	378	1986, 1987, 1988, 1989, 1990, 1991,	
<code>\ThisShouldNotHappen</code>	2958	1996, 2000, 2036, 2421, 2423, 2673	
<code>\title</code>	2	<code>\tl_set_eq:NN</code>	567
tl commands:		<code>\tl_to_str:N</code>	668, 675, 680
<code>\c_space_tl</code>	143, 1061,	<code>\tl_to_str:n</code>	18, 19, 20, 31, 377, 2122, 2136
1092, 1097, 1102, 2698, 2699, 2706,		token commands:	
2882, 2884, 2894, 2896, 2898, 2901,		<code>\token_to_str:N</code>	27
2903, 3099, 3102, 3234, 3262, 3263		<code>\toks</code>	3421, 3426, 3492, 3497
<code>\tl_const:Nn</code>	399		
<code>\tl_gput_right:Nn</code>	510		
<code>\tl_gset:Nn</code>	1822, 1842, 1846		
<code>\tl_if_blank:nTF</code>			
685, 2061, 2079, 2117, 2131, 2238, 2251			
<code>\tl_if_empty:nTF</code>			
... 561, 1408, 3087, 3091, 3100, 3263			
<code>\tl_if_empty:nTF</code> ...	206, 930, 966,		
985, 1012, 1031, 1599, 1655, 1760, 2045			
<code>\tl_if_empty_p:N</code>	1824, 1849		
<code>\tl_if_eq:NnTF</code>	1971		
<code>\tl_if_exist:nTF</code> ...	3140, 3250, 3254		
<code>\tl_if_exist_p:N</code> ...	3131, 3239, 3241		
<code>\tl_if_head_eq_charcode:nNTF</code> ...			
... 811, 828, 845			
<code>\tl_new:N</code>	179,		
180, 181, 182, 389, 393, 394, 395,			
396, 419, 420, 421, 422, 423, 424,			
425, 426, 427, 435, 2420, 2422, 2672			
<code>\tl_set:Nn</code>	199, 200, 212, 242,		
245, 257, 260, 305, 308, 328, 429,			
430, 431, 432, 433, 434, 666, 1807,			
1811, 1815, 1838, 1973, 1976, 1983,			

1986, 1987, 1988, 1989, 1990, 1991,	
1996, 2000, 2036, 2421, 2423, 2673	
<code>\tl_set_eq:NN</code>	567
<code>\tl_to_str:N</code>	668, 675, 680
<code>\tl_to_str:n</code>	18, 19, 20, 31, 377, 2122, 2136

token commands:	
<code>\token_to_str:N</code>	27
<code>\toks</code>	3421, 3426, 3492, 3497

U

unicode (hypersetup key)	1318
<code>\Url</code>	3, 5, 183
url (hypersetup key)	9, 1396
<code>\url</code>	3-6, 298
urlborder (hypersetup key)	13
urlborderstyle (hypersetup key) .	13, 1006
urlcolor (hypersetup key)	896
<code>\Urlfont</code>	6
<code>\UrlSpecial</code>	4
use commands:	
<code>\use:n</code>	765

V

verbose (hypersetup key)	1328
--------------------------------	------

W

<code>\write</code>	2369, 2372, 2392, 2411
---------------------------	------------------------

X

<code>\x</code>	2621, 2641, 3425, 3427, 3496, 3498
-----------------------	------------------------------------