

The `hyperref-generic` module

A generic driver for `hyperref`

The L^AT_EX Project*

Version 0.95q, released 2022-07-08

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
\DocumentMetadata % activates it
{
  %% options
  %% e.g. pdf version, backend:
  % pdfversion=1.7,
  % backend = dvipdfmx
}
```

Starting with L^AT_EX-2022-06-01 (or a suitable `latex-dev`) loading the testphase package is no longer required and this can be simplified to

```
\DocumentMetadata
{
  %% 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.

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

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.
- This driver uses the `l3color` module 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`.
- 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 `\DocumentMetadata` should be used:

```
\cs{DocumentMetadata}{lang=de-DE}
```

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

4 Dates

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

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

5 PDF page size (mediabox)

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

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

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

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

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

6 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. The optional argument can currently not be used together with the `\urldef` command.

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.

- Citations have by default no special color, they are colored like other internal links. You can use `citecolor` and `citebordercolor` to assign them a special color. This color is not reset if you use `allcolors` or switch to another color scheme. If you want the colors to follow `linkcolor` again you should remove the label `hyp/cite` and/or `hyp/citeborder` from the hook `hyp/link/cite`.
- 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 `cmymk`. `rgb` is the default. It does *not* change the model of text colors. Be aware that while the PDF format allows `cmymk` (4 numbers) in the `/C` key of an annotation, this is often ignored by pdf viewers and the colors can be wrong.
- The boolean keys `url`, `link`, `run`, `menu`, `file` allow to deactivate locally the link types.

`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 `primary-colors` (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_colorruncolor_bool,
\l_hyp_annot_colormenu_bool,
\l_hyp_annot_ocgcolorlink_bool,
\l_hyp_annot_ocgcolorurl_bool,
\l_hyp_annot_ocgcolorfile_bool,
\l_hyp_annot_ocgcolorruncolor_bool,
\l_hyp_annot_ocgcolormenu_bool,
```

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

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

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

The border colors aren't saved in color names currently, but if the need would arise it would 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	<code>urlbordercolor = [rgb]{1,1,0}</code>
color expression	<code>urlbordercolor = red!50!blue</code>
command	<code>urlbordercolor = \mycolor</code>

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 `LATEX` 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.

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 `\DocumentMetadata`

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 `\DocumentMetadata`:

```
\DocumentMetadata{pdfstandard=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 `pdftex` and `luatex` the automatic calculation of the coordinates from the encompassing box. With dvips and (x)dvipdfmx it will fall back to `Fit`. `pdfstartview` and `pdfremotestartview` will fallback to `Fit` too.

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

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

13.1 Names of destinations

`hyperref` creates two types of destination names: For numbered structures (so when the anchor is set by `\refstepcounter`) it builds the name from the counter name and the `\theH...` representation: `<counter name>.\theH<counter name>`.

For unnumbered structures, e.g. starred chapters or anchors created with `\phantomsection` it uses names like `section*.<number>` and `chapter*.<number>`.

Typically the name of destination can be retrieved by setting a label, this works also with unnumbered sections. The anchor and also the page can be retrieve in an expandable way with the help of commands from the `refcount` package which is loaded by `hyperref`. For example with the following commands it is possible to use the label to create a bookmark:

```
\bookmark[dest=\getrefbykeydefault{label}{anchor}{Doc-Start}]{my bookmark}
\bookmark[dest=page.\getrefbykeydefault{label}{page}{Doc-Start}]{my bookmark}
```

If a `\HyperDestNameFilter` is defined, this must be added around the definition, so actually the full code has to look like this

```
\bookmark[dest=
\HyperDestNameFilter{\getrefbykeydefault{label}{anchor}{Doc-Start}}]{mysection}
```

To simplify this hyperref provides `\hyperget{anchor}{label}` and `\hyperget{pageanchor}{label}`

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 `cmYk` (four numbers). While the PDF reference allows four numbers, PDF readers don't necessarily handle this correctly, so the value can be wrong.

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

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

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

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

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

will create

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

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

pdfborder (*setup key*) This key set accept as value three numbers or three numbers and an array describing a dash pattern, examples are `0 0 1` or `0 0 1 [3 2]`. The first two numbers should

linkborder (*setup key*)

urlborder (*setup key*)

runborder (*setup key*)

menuborder (*setup key*)

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.

<code>pdfborderstyle</code> (<i>setup key</i>)	The value of this key is the content of the BS dictionary. As an example		
<code>linkborderstyle</code> (<i>setup key</i>)	<code>/Type/Border /W 1 /S/U /D[3 2]</code>		
<code>urlborderstyle</code> (<i>setup key</i>)	Key	Values	comment / example
<code>fileborderstyle</code> (<i>setup key</i>)	<code>/Type</code>	<code>/Border</code>	optional
<code>runborderstyle</code> (<i>setup key</i>)	<code>/W</code>	<code><number></code>	Width of border line
<code>menuborderstyle</code> (<i>setup key</i>)	<code>/S</code>	<code>/S</code>	solid (default)
		<code>/D</code>	dash pattern can be set with <code>/D</code>
		<code>/B</code>	beveled
		<code>/I</code>	inset
		<code>/U</code>	underline
	<code>/D</code>	<code><array of numbers></code>	dash pattern (lines/gaps) (default [3])
<code>pdfcreationdate</code> (<i>setup key</i>)	Setting these keys is normally not needed. If they are used the values are stored di-		
<code>pdfmoddate</code> (<i>setup key</i>)	rectly in the Info dictionary for <code>/Creationdate</code> and <code>/ModDate</code> . The values are converted to strings but not processed further, so they should have the correct PDF format without the enclosing parentheses, e.g. <code>D:20200202111111+01'00'</code> .		
<code>pdflinkmargin</code> (<i>setup key</i>)	As described in the <code>hyperref</code> manual the behaviour differs between the backends: with dvips it is possible to change links locally, pdf _l atex and lua _l atex work by page, with dvipdfmx the setting is global (and has to be done in the preamble).		
<code>pdflang</code> (<i>setup key</i>)	The key will work, but it is recommended to set the language in <code>\DocumentMetadata</code> instead.		

File I

hyperref-generic driver implementation

```

1 <*package>
2 <@@=hyp>
3 \ProvidesFile{hggeneric-testphase.def}[2022-07-08 v0.95q %
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 \cs_if_exist:NTF \DocumentMetadata
13 {

```

```

14 \msg_new:nnnn
15 { hyp }
16 { missing-resource-management }
17 { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
18 {
19   Activate~it~with \\
20   \tl_to_str:n{\DocumentMetadata{<options>}}\\
21   before~\tl_to_str:n{\documentclass}
22 }
23 }
24 {
25   \msg_new:nnnn
26   { hyp }
27   { missing-resource-management }
28   { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
29   {
30     Activate~it~with \\
31     \tl_to_str:n{\RequirePackage{pdfmanagement-testphase}}\\
32     \tl_to_str:n{\DocumentMetadata{<options>}}\\
33     before~\tl_to_str:n{\documentclass}
34   }
35 }

```

The pdfversion should be set in \DocumentMetadata

```

36 \msg_new:nnnn
37 { hyp }
38 { pdfversion-disabled }
39 {
40   This~hyperref~driver~ignores~the~pdfversion~key!\\
41   Set~the~pdfversion~in~\token_to_str:N \DocumentMetadata
42 }
43 {
44   For example:\\
45   \tl_to_str:n
46   {
47     \DocumentMetadata { pdfversion=1.7 }
48   }
49 }

```

A generic message for ignored keys.

```

50 \msg_new:nnn
51 { hyp }
52 { key-dropped }
53 {
54   This~hyperref~driver~ignores~the~key~#1!\\
55   Please~check~the~documentation.
56 }

```

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

```

57 \msg_new:nnn
58 { hyp }
59 { pdfa-no-push-button }
60 { PDF/A:~Push~button~with~JavaScript~is~prohibited }
61
62 \msg_new:nnn

```

```

63 { hyp }
64 { pdfa-no-reset-button }
65 { PDF/A:~Reset~action~is~prohibited }

```

pdf/A message for not allowed Named actions

```

66 \msg_new:nnn
67 { hyp }
68 { pdfa-no-named-action }
69 { PDF/A:~Named~action~#1~is~prohibited }

```

A message if the destination name is empty.

```

70 \msg_new:nnn
71 { hyp }
72 { empty-destination-name }
73 {
74   Empty~destination~name,\\
75   using~'~#1'
76 }

```

A message if the destination check fails

```

77 \msg_new:nnn
78 { hyp }
79 { invalid-destination-value }
80 {
81   Invalid~value~'~#1'~of~'~#2'   \\
82   is~replaced~by~'Fit'~\msg_line_context:.
83 }

```

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

```

84 \msg_new:nnn
85 { hyp }
86 { ignore-deprecated-or-unknown-option-in-pdf-version }
87 {
88   Option~'~#1'~is~unknown~or~deprecated~in\\
89   pdf~version~#2.~Ignored.
90 }
91 \msg_new:nnn
92 { hyp }
93 { ignore-deprecated-or-unknown-value-in-pdf-version }
94 {
95   Value~'~#1'~is~unknown~or~deprecated~in\\
96   pdf~version~#2.~Ignored.
97 }
98 \msg_new:nnn
99 { hyp }
100 { replace-deprecated-or-unknown-value-in-pdf-version }
101 {
102   Value~'~#1'~is~unknown~or~deprecated~in\\
103   pdf~version~#2. Value~'~#3'~is used instead.
104 }

```

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:

```

105 \msg_new:nnn
106 { hyp }
107 { unknown-key }

```



```

108 {
109     unknown~key~#2~of~module~'#1'~set~to~'#3'.
110 }
111 \msg_new:nnn
112 { hyp }
113 { unknown-key-to-Hyp }
114 {
115     ignored~in~family~Hyp~unknown~key~#1.
116 }

```

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

```

117 \cs_new:Npn \__hyp_clist_display:n #1 {*~#1\\}
118 \msg_new:nnn
119 { hyp }
120 { unknown-choice }
121 {
122     Value~'#3'~is~invalid~for~key~'#1'.\\
123     The~key~accepts~only~the~choices\\
124     \clist_map_function:nN { #2 }\__hyp_clist_display:n
125 }
126
127 \msg_new:nnn
128 { hyp }
129 { unknown-choice+empty }
130 {
131     Value~'#3'~is~invalid~for~key~'#1'.\\
132     The~key~accepts~only~the~choices\\
133     \clist_map_function:nN { #2 }\__hyp_clist_display:n
134     An~empty~value~removes~the~setting.
135 }
136
137 \msg_new:nnn
138 { hyp }
139 { no-bool }
140 {
141     Value~'#2'~is~invalid~for~key~'#1'.\\
142     The~key~accepts~only~the~choices\\
143     *~true\\
144     *~false \\
145     *~and~an~empty~value~which~removes~the~setting.\\
146     No~value~is~equivalent~to~using~'true'.
147 }

```

A message for creator and producer which can't be removed.

```

148 \msg_new:nnn
149 { hyp }
150 { empty-info-value }
151 {
152     Empty~value~for~key~#1.\\
153     This~isn't~honored~by~all~backends.
154 }

```

2 Variants

```
155 \cs_generate_variant:Nn\pdf_destination:nn {nf}
156 \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

```
157 \chardef\Hy@VersionChecked=1 %don't check the version!
158 %\cs_set_protected:Npn \PDF@SetupDoc{}
159 %\PDF@FinishDoc{}% dummy needed for hyperref ...
```

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

`\hypercalcbp`

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

(End definition for `\hypercalcbp`. This function is documented on page 18.)

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

```
161 \providecommand\@pdfborder{}
162 \providecommand\@pdfborderstyle{}
163 \newcommand\OBJ@OCG@view {} % needed in hyperref
164 \def\Hy@numberline#1{#1\c_space_tl} %needed by bookmark
```

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

```
165 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
166 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:
167 \legacy_if:nT { Hy@setpdfversion }
168 {
169   \msg_warning:nn { hyp }{ pdfversion-disabled }
170 }
171 \Hy@DisableOption{pdfversion}
```

`\Acrobatmenu` should use the new internal link command

```
172 \RenewDocumentCommand \Acrobatmenu { m m }
173 {
174   \hyper@linknamed {#1} {#2}
175 }
```

`\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!

```
176 % TODO should go at some time ...
177 % \kv@set@family@handler{Hyp}
178 % { \msg_warning:nnx {hyp}{unknown-key-to-Hyp}{#1} }
179 \cs_set_protected:Npn \hypersetup #1
180 {
181   \kvsetkeys{Hyp} {#1}
182   \keys_set:nn { hyp / setup }{ #1 }
183 }
184 % TODO for now unknown keys should only give warnings.
```

```

185 \keys_define:nn { hyp / setup }
186 {
187   unknown .code:n =
188   {
189     \msg_warning:nnxxx { hyp } { unknown-key }
190     { hyp~/~setup }{ \l_keys_key_str } { #1 }
191   }
192 }

```

Hyperref creates a number of destinations automatically. E.g. in unnumbered chapters and sections and with `\phantomsection`. The following key allows to force a specific name for the destination so that it can be used by bookmarks.

```

193 \keys_define:nn { hyp / setup }
194 {
195   next-anchor .code:n =
196   {
197     \AddToHookNext{__hyp/dest/make}
198     {\Hy@MakeCurrentHref{#1}}
199   }
200 }

```

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`)

```

201
202 \bool_new:N \l__hyp_href_url_encode_bool
203 \bool_new:N \l__hyp_href_url_ismap_bool
204 \tl_new:N \l__hyp_href_url_protocol_tl
205 \tl_new:N \l__hyp_href_pdf_destination_tl
206 \tl_new:N \l__hyp_href_pdf_page_tl
207 \tl_new:N \l__hyp_href_run_parameter_tl
208 \cs_new_protected:Npn \__hyp_href_url_format: {\begingroup\Url}
209
210
211 \keys_define:nn { hyp / href }
212 {
213   ,urlencode .bool_set:N = \l__hyp_href_url_encode_bool
214   ,format .code:n = { \cs_set:Nn \__hyp_href_url_format: {#1} },
215   ,protocol .tl_set:N = \l__hyp_href_url_protocol_tl
216   ,destination .tl_set:N = \l__hyp_href_pdf_destination_tl
217   ,pdfremotestartview .code:n =
218   {
219     \keys_set:nn { hyp / setup }
220     { pdfremotestartview = #1 }
221   }
222   ,page .code:n =
223   {
224     \tl_set:Nn \l__hyp_href_pdf_page_tl {#1}
225     \tl_set:Nn \Hy@href@page {#1}
226   }
227   ,ismap .bool_set:N = \l__hyp_href_url_ismap_bool
228   ,run-parameter .tl_set:N = \l__hyp_href_run_parameter_tl
229   ,nextactionraw .code:n =

```

```

230 { %perhaps some safety match later, see hyperref code
231   \tl_if_empty:nTF {#1}
232   {
233     \pdfdict_remove:nn{l_hyp/annot/A}{Next}
234   }
235   {
236     \pdfdict_put:nnn{l_hyp/annot/A}{Next}{#1}
237     \tl_set:Nn \Hy@href@nextactionraw {/Next~#1}
238     \keys_set:nn {hyp / setup }{ pdfnewwindow = true}
239   }
240 }
241 ,afrelationship .code:n =
242 {
243   \pdfdict_put:nnx
244   { l_pdffile/Filespec}{AFRelationship}{ \pdf_name_from_unicode_e:n {#1}}
245 }
246
247 }
248
249 \keys_define:nn { hyp / setup }
250 {
251   ,href / urlencode .bool_set:N = \l__hyp_href_url_encode_bool
252   ,href / urlencode .default:n = {true}
253   ,href / urlencode .initial:n = {false}
254   ,href / protocol .tl_set:N = \l__hyp_href_url_protocol_tl
255   ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
256   ,href / format .code:n = { \cs_set:Nn \__hyp_href_url_format:{#1} }
257 }
258
259 \hook_new_pair:nn{cmd/href/before}{cmd/href/after}
260
261 \DeclareRobustCommand*{\href}[1][ ]{%
262   \mode_leave_vertical:
263   \hook_use:n{cmd/href/before}
264   \group_begin:
265   \keys_set:nn { hyp / href } {#1}
266   \bool_if:NTF \l__hyp_href_url_encode_bool
267   {
268     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
269   }
270   {
271     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
272   }
273   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
274 }
275
276 \begingroup
277   \catcode'\$=6 %
278   \catcode'\#=12 %
279   \gdef\href@\$1{\expandafter\href@split$1##\}%
280   \gdef\href@split$1#$2#$3\\$4{%
281     \hyper@@link{$1}{$2}{$4}%<---__hyp-docstrip doubling!
282   }
283   \hook_use:n{cmd/href/after}

```

```

284 }%
285 \endgroup
286
287 \hook_new_pair:nn{cmd/hrefurl/before}{cmd/hrefurl/after}
288
289 \DeclareRobustCommand*{\hrefurl}[1] []
290 {
291   \mode_leave_vertical:
292   \hook_use:n{cmd/href/before}
293   \group_begin:
294   \keys_set:nn { hyp / href } {#1}
295   \bool_if:NTF \l__hyp_href_url_encode_bool
296   {
297     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
298   }
299   {
300     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
301   }
302   \hyper@normalise\__hyp_href_url_aux:nn}
303
304 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
305 {
306   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
307   \group_end:
308   \hook_use:n{cmd/href/after}
309 }
310
311 \hook_new_pair:nn{cmd/hrefpdf/before}{cmd/hrefpdf/after}
312 \DeclareRobustCommand*{\hrefpdf}[1] []
313 {
314   \mode_leave_vertical:
315   \hook_use:n{cmd/hrefpdf/before}
316   \group_begin:
317   \keys_set:nn { hyp / href } {#1}
318   \hyper@normalise\__hyp_href_pdf_aux:nn
319 }
320
321 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
322 {
323   \exp_args:Nnno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
324   \group_end:
325   \hook_use:n{cmd/hrefpdf/after}
326 }
327
328 \hook_new_pair:nn{cmd/hrefrun/before}{cmd/hrefrun/after}
329 \DeclareRobustCommand*{\hrefrun}[1] []
330 {
331   \mode_leave_vertical:
332   \hook_use:n{cmd/hrefrun/before}
333   \group_begin:
334   \keys_set:nn { hyp / href } {#1}
335   \hyper@normalise\__hyp_href_run_aux:nn
336 }
337

```

```

338 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
339 {
340   \exp_args:Nno\hyper@linklaunch{#1}{#2}{\l__hyp_href_run_parameter_tl}
341   \group_end:
342   \hook_use:n{cmd/hrefrun/after}
343 }
344
345
346 \hook_new_pair:nn{cmd/url/before}{cmd/url/after}
347
348 \DeclareRobustCommand*\url}[1] []
349 {
350   \mode_leave_vertical:
351   \hook_use:n{cmd/url/before}
352   \group_begin:
353   \keys_set:nn {hyp / href } {#1}
354   \bool_if:NTF \l__hyp_href_url_encode_bool
355   {
356     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
357   }
358   {
359     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
360   }
361   \hyper@normalise\__hyp_href_url_aux:n
362 }
363
364 \cs_new_protected:Npn \__hyp_href_url_aux:n #1
365 {
366   \exp_args:Nno
367   \hyper@linkurl{\__hyp_href_url_format: {#1}}
368   {\l__hyp_href_url_protocol_tl#1}
369   \group_end:
370   \hook_use:n{cmd/url/after}
371 }
372

```

the `\urldef` command doesn't like the optional argument, so we overwrite locally the `\url` command here:

```

373
374 \def\urldef#1#2{\begingroup\def\url{\hyper@normalise\url@}\setbox\z@\hbox\bgroup
375   \def\Url@HyperHook##1\endgroup{\Url@def{#1}{#2}}%
376   % Because hyperref breaks \urldef and does not define its own (Grrrr!)...
377   \def\url@##1{\egroup\endgroup\DeclareRobustCommand#1{#2{##1}}}%
378   #2}
379

```

make the new commands compatible with `\pdfstringdef`:

```

380 \NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
381 \pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
382 \pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}
383 \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.

```
384 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
385 {
386   %\tl_set:cn {@#1}{#2}
387   \AddToDocumentProperties[hyperref]{#1}{#2}
388 }
389 \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.

```
390 \hook_new:n{hyp/link/cite}
391 %\color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
392 %\color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
393 \keys_define:nn { hyp / setup }
394 {
395   ,citecolor .code:n = {\__hyp_color_set:ne {hyp/color/cite}{#1}\__hyp_citecolor_hook_init}
396   ,citebordercolor
397     .code:n = {\__hyp_color_set:ne {hyp/color/citeborder}{#1}\__hyp_citebordercolor_hook_init}
398 }
399 \cs_new_protected:Npn \__hyp_citecolor_hook_init:
400 {
401   \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
402   {
403     \keys_set:nn { hyp / setup }
404     {
405       linkcolor      = hyp/color/cite
406     }
407   }
408   \cs_gset_eq:NN \__hyp_citecolor_hook_init: \prg_do_nothing:
409 }
410 \cs_new_protected:Npn \__hyp_citebordercolor_hook_init:
411 {
412   \hook_gput_code:nnn { hyp/link/cite }{hyp/citeborder}
413   {
414     \keys_set:nn { hyp / setup }
415     {
416       linkbordercolor = hyp/color/citeborder
417     }
418   }
419   \cs_gset_eq:NN \__hyp_citebordercolor_hook_init: \prg_do_nothing:
420 }
421
```

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

```

422 \bool_lazy_and:nnF
423 { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
424 { \msg_error:nn { hyp}{ missing-resource-management } }

```

Outlines/bookmarks require the bookmark package. TODO check pdfpagemode if bookmarks are suppressed. TODO We overwrite the color key here for now, but this should be moved to bookmark

```

425 \AddToHook { package/bookmark/after}
426 {
427   \define@key{BKM}{color}
428   {
429     \__hyp_color_set:ne {__hyp/tmpa}{#1}
430     \color_export:nVN
431       {__hyp/tmpa}
432       \g__hyp_bordercolormodel_str
433       \BKM@color
434   }
435 }
436 \legacy_if:nT { Hy@bookmarks }
437 {
438   \AddToHook{begindocument/before}[hyperref/bookmark]
439   {
440     \RequirePackage{bookmark}
441   }
442 }
443 \legacy_if:nT { Hy@draft}
444 {
445   \PassOptionsToPackage{draft}{bookmark}
446 }

```

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 447 %
\__hyp_ref_check:nn 448 \cs_new_protected:Npn \__hyp_ref_label:nn #1 #2 %label/attributes
\__hyp_ref_value:nn 449 {
450   \@bsphack
451   \ref_label:nn{#1}{#2}
452   \@esphack
453 }
454 \cs_generate_variant:Nn \__hyp_ref_label:nn {en}

```


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

```

455 \prg_new_eq_conditional:Nn \__hyp_ref_if_exist:nn \ref_if_exist:nn { p , T , F, TF }
456 \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.

```

457 \cs_new_protected:Npn \__hyp_ref_check:nn #1 #2 %label/attribute
458 {
459   \__hyp_ref_if_exist:nnF {#1}{#2}
460   {
461     \protect\G@refundefinedtrue
462     \@latex@warning
463     {
464       Reference~'\tl_to_str:n {#1}'~with~attribute~'\tl_to_str:n {#2}'~
465       on~page~\thepage~\space undefined
466     }
467   }
468 }
469 \cs_generate_variant:Nn \__hyp_ref_check:nn {en}

```

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

```

470 \cs_new:Npn \__hyp_ref_value:nn #1 #2
471 {
472   \ref_value:nn{#1}{#2}
473 }
474 \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 475 \box_new:N \l__hyp_tmpa_box
\l__hyp_tmpa_int 476 \tl_new:N \l__hyp_tmpa_tl
\l__hyp_tmpa_box 477 \seq_new:N \l__hyp_tmpa_seq
\l__hyp_tmpa_str 478 \int_new:N \l__hyp_tmpa_int
479 \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.

TODO: document and check use!

```

\l__hyp_dest_name_tmpa_tl
\l__hyp_uri_tmpa_tl 480 \tl_new:N \l__hyp_dest_name_tmpa_tl
\l__hyp_filename_tmpa_tl 481 \tl_new:N \l__hyp_uri_tmpa_tl
\__hyp_text_tmpa_str_uu\g__hyp_text_tmpa_str 482 \tl_new:N \l__hyp_filename_tmpa_tl
483 \tl_new:N \l__hyp_para_tmpa_tl
484 \str_new:N \l__hyp_text_tmpa_str
485 \str_new:N \g__hyp_text_tmpa_str

```

(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.

```
486 \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`

```
487 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
488 {url,link,file,menu,run}
489 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
490 {
491   URI    = url,
492   GoTo   = link,
493   GoToR  = file,
494   Named  = menu,
495   Launch = run
496 }
497 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
498 {
499   url    = URI,
500   link   = GoTo,
501   file   = GoToR,
502   menu   = Named,
503   run    = Launch
504 }
505
```

(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
`\g__hyp_dest_pdfstartview_tl` the current and remote files. The instruction is in “PDF format” but without the leading
`\l__hyp_dest_pdfremotestartview_tl` slash!

```
506 \tl_new:N \g__hyp_dest_pdfstartpage_tl
507 \tl_new:N \g__hyp_dest_pdfstartview_tl
508 \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
509 \tl_new:N \l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_dest_tl
510 \tl_new:N \l__hyp_text_enc_info_print_tl
\l__hyp_text_enc_dest_print_tl
511 \tl_new:N \l__hyp_text_enc_dest_tl
\l__hyp_text_enc_file_print_tl
512 \tl_new:N \l__hyp_text_enc_dest_print_tl
\l__hyp_text_enc_para_print_tl
513 \tl_new:N \l__hyp_text_enc_file_print_tl
514 \tl_new:N \l__hyp_text_enc_para_print_tl
```

```

515
516 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
517 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
518 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
519 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
520 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
521 \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.

```

522 \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*)

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

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

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

```

523 \str_new:N \g__hyp_bordercolormodel_str

```

(End definition for `\g__hyp_bordercolormodel_str`.)

7.4 Booleans

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

`\l_hyp_annot_colorurl_bool`

`\l_hyp_annot_colorfile_bool`

`\l_hyp_annot_colorrun_bool`

`\l_hyp_annot_colormenu_bool`

```

524 \seq_map_inline:Nn \c__hyp_annot_types_seq

```

```

525 {

```

```

526   \bool_new:c {l_hyp_annot_color#1_bool}

```

```

527 }

```

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

`\l_hyp_annot_ocgcolorlink_bool`

`\l_hyp_annot_ocgcolorurl_bool`

`\l_hyp_annot_ocgcolorfile_bool`

`\l_hyp_annot_ocgcolorrun_bool`

`\l_hyp_annot_ocgcolormenu_bool`

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

```

528 \seq_map_inline:Nn \c__hyp_annot_types_seq

```

```

529 {

```

```

530   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}

```

```

531 }

```

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

`\l_hyp_annot_Named_bool`

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

```

532 \seq_map_inline:Nn \c_pdfannot_link_types_seq

```

```

533 {

```

```

534   \bool_new:c {l__hyp_annot_#1_bool}

```

```

535   \bool_set_true:c {l__hyp_annot_#1_bool}

```

```

536 }

```

(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.

```
537 \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.

```
538 \regex_const:Nn \c__hyp_dest_startview_regex
539 {
540   \A\ *
541   (?:
542     (?:XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
543     |
544     (?:Fit\b|FitB\b)
545     |
546     (?:\b(?:FitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1})
547     |
548     (?:FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
549   )
550 }
```

(End definition for `\c__hyp_dest_startview_regex`.)

7.7 PDF dictionaries

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

```
551 \pdfdict_new:n {l__hyp_page/Trans}
552 \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`

```
553 \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`

```

554 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
555 {
556   \str_set:Nx #2 {\text_purify:n { #1 } }
557 }

```

(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`

```

558 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
559 {
560
561 }

```

(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`

```

562 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
563 {
564   \pdf_string_from_unicode:nVN { #1 } #2 #2
565 }

```

(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`

```

566 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
567 {
568   \group_begin:
569   \Hy@pdfstringtrue
570   \hook_use:n {hyp/text/pdfstring}
571   \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
572   \__hyp_text_cleanup:N \l__hyp_text_tmpa_str
573   \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
574   \str_gset_eq:NN \g__hyp_text_tmpa_str \l__hyp_text_tmpa_str
575   \group_end:
576   \str_set_eq:NN #3 \g__hyp_text_tmpa_str
577 }
578 \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
579 \cs_new_protected:Npn \Hy@pstringdef #1 #2
580 { \_hyp\_text\_pdfstring:xnN {#2} {utf8/string-raw}#1 }
This is a special version for info keys:
```

```
\_hyp\_text\_pdfstring\_info:nnN
```

```
581 \cs_new_protected:Npn \_hyp\_text\_pdfstring\_info:nnN #1 #2
582 {
583   \_hyp\_text\_pdfstring:nonN { #1 } { \l\_hyp\_text\_enc\_info\_print\_tl } #2
584 }
```

```
(End definition for \_hyp\_text\_pdfstring\_info:nnN.)
```

9 Pagelabels

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

The current implementation in hyperref/hyperref drivers:

xetex: hxdetex.def, line 80-110

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

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

dvipdfm: identical to xetex

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

luatex identical to pdftex

The code in hyperref inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-30 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
```

```
585 \cs_new_protected:Npn \_hyp\_PageLabels\_gpush:
586 {
587   \pdfmanagement\_add:nnx {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
```

```

588 }
589
590 \def\Hy@PutCatalog #1 {}
591
592
593 \legacy_if:nT { Hy@pdfpagelabels }
594 {
595   \cs_set_protected:Npn \HyPL@StorePageLabel #1
596   {
597     \tl_gput_right:Nx \HyPL@Labels { \the\Hy@abspage<<#1>> }
598     \__hyp_PageLabels_gppush:
599   }
600 }

```

(End definition for `__hyp_PageLabels_gppush:`, `\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

601 \Hy@WrapperDef \__hyp_destination:nn #1 #2
602 {
603   \mode_if_horizontal:T { \@savsf\spacefactor }
604   \Hy@SaveLastskip      %defined in hyperref
605   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
606   \__hyp_text_pdfstring:xoN
607   { \HyperDestNameFilter{#1} }

```

```

608     { \l__hyp_text_enc_dest_tl }
609     \l__hyp_tmpa_tl
610 \str_if_eq:nnTF {#2} {fitrbox}
611 {
612     \exp_args:NV
613     \pdf_destination:nnnn \l__hyp_tmpa_tl
614     { \box_wd:N \l__hyp_dest_box }
615     { \box_ht:N \l__hyp_dest_box }
616     { \box_dp:N \l__hyp_dest_box }
617 }
618 {
619     \exp_args:NV
620     \pdf_destination:nf
621     { \l__hyp_tmpa_tl }
622     { #2 }
623 }
624 \Hy@RestoreLastskip %defined in hyperref
625 \mode_if_horizontal:T { \spacefactor\@savsf }
626 }

```

(End definition for `__hyp_destination:nn`.)

This are the three destinations commands. They are modelled along the xetex version. It is not quite clear if really all three are needed for the backends supported by this driver, but changing the hyperref code would be difficult. We add a hook. This allows e.g. the tagging code to create also a structured destination. We don't use the cmd hook, as we want the same hook for both start commands. We make the current dest name available so that the hook code can use it.

```

\hyper@anchor
\hyper@anchorstart 627 \tl_new:N\l_hyp_current_dest_name_tl
\hyper@anchorend 628 \hook_new:n{hyp/anchor}
hyp/anchor 629 \cs_new_protected:Npn \hyper@anchor #1
\l_hyp_current_dest_name_tl 630 {
631     \exp_args:NnV
632     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
633     \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
634     \hook_use:n{hyp/anchor}
635 }
636
637 \cs_new_protected:Npn \hyper@anchorstart #1
638 {
639     \Hy@activeanchortrue
640     \exp_args:NnV
641     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
642     \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
643     \hook_use:n{hyp/anchor}
644 }
645
646 \cs_new_protected:Npn \hyper@anchorend
647 {
648     \Hy@activeanchorfalse
649 }

```

(End definition for `\hyper@anchor` and others. 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:

```

650 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
651 {
652   \mode_leave_vertical:
653   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
654   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
655   {
656     \msg_warning:nnx
657     { hyp }
658     { empty-destination-name }
659     { \c__hyp_dest_undefined_tl }
660     \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
661   }
662   {
663     \__hyp_text_pdfstring:xoN
664     { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
665     { \l__hyp_text_enc_dest_tl }
666     \l__hyp_dest_name_tmpa_tl
667   }
668   \exp_args:No
669   \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
670 }
671
672 \cs_new_protected:Npn \__hyp_link_goto_end:
673 {
674   \pdfannot_link_goto_end:
675 }
```

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.

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

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. We store again the destination name in the public `tl_l_hyp_current_dest_name_tl` so that the hook code can make use of it

```

676
677 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
678 {
679   \bool_if:NTF \l__hyp_annot_GoTo_bool
680   {
681     \Hy@VerboseLinkStart{#1}{#2}
682     \group_begin:
683     \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
684     \hook_use:n {hyp/link/#1}
685     \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
686     \__hyp_link_goto_end:
687     \group_end:
688     \Hy@VerboseLinkStop
689   }
690   {\let\protect\relax#3}}
691 }
692
693 \cs_new_protected:Npn \hyper@linkstart #1 #2 % #1 context, #2=destination name
694 {
695   \bool_if:NT \l__hyp_annot_GoTo_bool
696   {
697     \Hy@VerboseLinkStart{#1}{#2}% only for debug
698     \group_begin:
699     \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
700     \hook_use:n {hyp/link/#1}
701     \__hyp_link_goto_begin:nw {#2}
702   }
703 }
704
705 \cs_new_protected:Npn \hyper@linkend
706 {
707   \bool_if:NT \l__hyp_annot_GoTo_bool
708   {
709     \__hyp_link_goto_end:
710     \group_end:
711     \Hy@VerboseLinkStop
712   }
713 }
```

10.3 URI links

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

```

714 \pdfdict_new:n {l_hyp/annot/A/URI}
```

```

715 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
716 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
717
718 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
719 {
720   \bool_if:NTF \l__hyp_annot_URI_bool
721   {
722     \group_begin:
723     \__hyp_text_pdfstring:xoN
724     { #2}
725     { \l__hyp_text_enc_uri_print_tl }
726     \l__hyp_uri_tmpa_tl
727     \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
728     \bool_if:NT \l__hyp_href_url_ismap_bool
729     {
730       \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
731     }
732     \cs_set_eq:NN \# \c_hash_str
733     \cs_set_eq:NN \% \c_percent_str
734     \Hy@safe@activestrue
735     \mode_leave_vertical:
736     \pdfannot_dict_put:nxx {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
737     \pdfannot_link:nxx { URI }
738     {
739     }
740     {
741       \let\protect\relax
742       #1
743       \Hy@xspace@end
744       \Hy@VerboseLinkStop %where is the start??
745     }
746     \group_end:
747   }
748   {{\let\protect\relax#1}}
749 }
750

```

10.4 GoToR Links files

```

751 \pdfdict_new:n {l_hyp/annot/A/GoToR}
752 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
753 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
754
755
756 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
757 {
758   \bool_if:NTF \l__hyp_annot_GoToR_bool
759   {
760     \group_begin:
761     \tl_set:Nx \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
762     \exp_args:Nx
763     \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
764     {
765       \pdfdict_put:nxx { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n {ap

```

```

766         \exp_args:Nnox
767         \pdffile_embed_file:nnn
768         {}
769         {\l__hyp_filename_tmpa_tl }
770         {\__hyp_file_tl_to_str:N \l__hyp_filename_tmpa_tl }
771     }
772     \pdfdict_put:nnx
773     {l_hyp/annot/A/GoToR}
774     {F}
775     {\pdf_object_ref:e {\__hyp_file_tl_to_str:N \l__hyp_filename_tmpa_tl}}
776     \__hyp_text_pdfstring:nnN
777     { #3 }
778     { \l__hyp_text_enc_dest_print_tl }
779     \l__hyp_dest_name_tmpa_tl
780     \tl_if_blank:eTF {#3}
781     {
782         \pdfdict_put:nnx {l_hyp/annot/A/GoToR}{D}
783         {
784             [
785                 \int_eval:n
786                 { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
787                 /\l__hyp_dest_pdfremotestartview_tl
788             ]
789         }
790     }
791     {
792         \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
793     }
794     \mode_leave_vertical:

```

We use an extra object here, as ghostscript doesn't like the object reference in the dict
<https://chat.stackexchange.com/transcript/message/57361080#57361080>

```

795     \pdf_object_unnamed_write:nx{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
796     \pdfannot_dict_put:nnx {link/GoToR}{A}{\pdf_object_ref_last:}
797     \pdfannot_link:nnx %expansion??
798     { GoToR }
799     {
800     }
801     {
802         \let\protect\relax
803         #1\Hy@xspace@end
804         \Hy@VerboseLinkStop %where is the start??
805     }
806     \group_end:
807 }
808 {\let\protect\relax#1}}
809 }

```

10.5 Launch links

We define \hyper@linklaunch for naming consistency

```

810 \pdfdict_new:n {l_hyp/annot/A/Launch}
811 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
812 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
813

```

```

814 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
815 {
816   \bool_if:NTF \l__hyp_annot_Launch_bool
817   {
818     \group_begin:
819     \__hyp_text_pdfstring:nnN
820       { #1 }
821       { \l__hyp_text_enc_file_print_tl }
822       \l__hyp_filename_tmpa_tl
823     \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
824     \__hyp_text_pdfstring:noN
825       { #3 }
826       { \l__hyp_text_enc_para_print_tl }
827       \l__hyp_para_tmpa_tl
828     \bool_if:nTF
829     {
830       \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
831       ||
832       \pdf_version_compare_p:Nn > {1.9}
833     }
834     {
835       \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
836     }
837     {
838       \pdfdict_put:nnx {l_hyp/annot/A/Launch}{Win}
839       {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
840     }
841     \mode_leave_vertical:
842     \pdfannot_dict_put:nnx {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}>>}
843     \pdfannot_link:nxn
844       { Launch }
845       {
846         % /A
847         % <<
848         % \pdfdict_use:n {l_hyp/annot/A/Launch}
849         % >>
850       }
851     {
852       \let\protect\relax
853       #2\Hy@xspace@end
854       \Hy@VerboseLinkStop %where is the start??
855     }
856     \group_end:
857   }
858   {\let\protect\relax#2}}
859 }

```

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

```

860 \use:x
861 { % filename, anchor text, linkname
862   \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str ##1 \exp_not:N \ \ ##2 #
863 }
864 {

```

```

865 \hyper@linklaunch {#1}{#2}{#3}
866 }

```

10.6 Named links (menu)

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

```

867 \pdfdict_new:n {l_hyp/annot/A/Named}
868 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
869 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
870
871 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
872 {
873   \bool_if:NTF \l__hyp_annot_Named_bool
874   {
875     \group_begin:
876     \pdfmeta_standard_verify:nnTF {named_actions}{#1}
877     {
878       \mode_leave_vertical:
879       \pdfdict_put:nnx {l_hyp/annot/A/Named}{N}
880       {\pdf_name_from_unicode_e:n{#1}}
881       \pdfannot_dict_put:nnx {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}>>}
882       \pdfannot_link:nnx { Named }
883       {
884         % /A
885         % <<
886         % \pdfdict_use:n { l_hyp/annot/A/Named }
887         % >>
888       }
889       {
890         #2
891         \Hy@xspace@end
892         \Hy@VerboseLinkStop
893       }
894     }
895     {
896       \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
897       #2
898     }
899     \group_end:
900   }
901   {{\let\protect\relax#2}}
902 }
903

```

11 Link decorations

11.1 Functions to export and select colors

We support two input syntax: color expressions and model with values. Exporting can be done by first setting the color with `__hyp_color_set:nn` (if needed to a temporary color name) and then using `\color_export:nnN`. But we need a variant as the export format `space-sep-cmyk` or `space-sep-rgb` is stored in a tl.

904 \cs_generate_variant:Nn \color_export:nnN {nVN}

_hyp_color_select:n _hyp_color_select:n {<color>}

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

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

```

905 \cs_new_protected:Npn \_hyp_color_select:n #1
906 {
907   \tl_if_head_eq_charcode:nNTF {#1}[ %]
908   {
909     \_hyp_color_select_aux:wn #1
910   }
911   {
912     \color_select:n {#1}
913   }
914 }
915
916 \cs_new_protected:Npn \_hyp_color_select_aux:wn [#1] #2
917 {
918   \color_select:nn {#1}{#2}
919 }
920
921 \cs_generate_variant:Nn \_hyp_color_select:n {e}

```

(End definition for _hyp_color_select:n and _hyp_color_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}
_hyp_color_set_aux:nwn and color=red!50!blue.

```

922 \cs_new_protected:Npn \_hyp_color_set:nn #1 #2
923 {
924   \tl_if_head_eq_charcode:nNTF {#2}[ %]
925   {
926     \_hyp_color_set_aux:nwn { #1 } #2
927   }
928   {
929     \color_set:nn {#1} {#2}
930   }
931 }
932
933 \cs_new_protected:Npn \_hyp_color_set_aux:nwn #1 [#2] #3
934 {
935   \color_set:nnn {#1}{#2}{#3}
936 }
937
938 \cs_generate_variant:Nn \_hyp_color_set:nn {ne}

```

(End definition for _hyp_color_set:nn and _hyp_color_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.

```

939 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
940 {
941   \hook_gput_code:nnn
942     {pdfannot/link/#2/begin}
943     {hyp/color}
944   {
945     \bool_if:cT { l_hyp_annot_color#1_bool }
946     {
947       \group_begin:
948       \color_select:n { hyp/color/#1}
949     }
950   }
951   \hook_gput_code:nnn
952     {pdfannot/link/#2/end}
953     {hyp/color}
954   {
955     \bool_if:cT { l_hyp_annot_color#1_bool }
956     {
957       \group_end:
958     }
959   }
960 }
```

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

```

961 \keys_define:nn { hyp / setup }
962 {
963   ,colorlinks .meta:n =
964   {
965     ,pdfborder={0~0~0}
966     ,pdfborderstyle=
967     ,colorurl =#1
968     ,colorlink =#1
969     ,colorrurl =#1
970     ,colormenu =#1
971     ,colorfile =#1
972   }
973   ,colorlinks .default:n = {true}
974 }
```

`colorurl` (*setup key*)

`colorlink` (*setup key*)

`colorrurl` (*setup key*)

`colormenu` (*setup key*)

`colorfile` (*setup key*)

`urlcolor` (*setup key*)

`linkcolor` (*setup key*)

`runcolor` (*setup key*)

`menucolor` (*setup key*)

`filecolor` (*setup key*)

`allcolors` (*setup key*)

```

975 \seq_map_inline:Nn \c__hyp_annot_types_seq
976 {
977   \keys_define:nn { hyp / setup }
978   {
979     ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
980     ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
981   }
982 }
983
```



```

984 \keys_define:nn { hyp / setup }
985 {
986   ,allcolors .meta:n =
987   {
988     ,urlcolor=#1
989     ,linkcolor=#1
990     ,runcolor=#1
991     ,filecolor=#1
992     ,menucolor=#1
993   }
994   ,allcolors .value_required:n = true
995 }

```

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*)

```

996 \keys_define:nn { hyp / setup }
997 {
998   ,bordercolormodel .choices:nn =
999   {rgb,cmyk}
1000   { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
1001   ,bordercolormodel .initial:n = {rgb}
1002 }

1003 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1004 {
1005   \keys_define:nn { hyp / setup }
1006   {
1007     #1bordercolor .code:n =
1008     {
1009       \tl_if_empty:nTF { ##1 }
1010       {
1011         \pdfannot_dict_remove:nn
1012         {link/#2}
1013         { C }
1014       }
1015       {
1016         \__hyp_color_set:ne {hyp/color/#1border}{##1}
1017         \color_export:nVN
1018         {hyp/color/#1border}
1019         \g__hyp_bordercolormodel_str
1020         \l__hyp_tmpa_tl
1021         \pdfannot_dict_put:nnx
1022         {link/#2}
1023         { C }
1024         { [\l__hyp_tmpa_tl] }
1025       }
1026     }
1027   }
1028 }

```

```

1029
1030 \keys_define:nn { hyp / setup }
1031 {
1032   ,allbordercolors .meta:n =
1033   {
1034     ,linkbordercolor=#1
1035     ,urlbordercolor =#1
1036     ,filebordercolor=#1
1037     ,menubordercolor=#1
1038     ,runbordercolor =#1
1039   }
1040   ,allbordercolors .value_required:n = true
1041 }
1042

```

11.3.2 Borderwidth and -arc

```

1043 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1044 {
1045   \keys_define:nn { hyp / setup }
1046   {
1047     #1border .code:n =
1048     {
1049       \tl_if_empty:nTF { ##1 }
1050       {
1051         \pdfannot_dict_remove:nn
1052         {link/#2}
1053         { Border }
1054       }
1055       {
1056         \pdfannot_dict_put:nnn
1057         {link/#2}
1058         { Border }
1059         { [##1] }
1060       }
1061     }
1062   }
1063 }
1064 \keys_define:nn { hyp / setup }
1065 {
1066   ,pdfborder .code:n =
1067   {
1068     \tl_if_empty:nTF { #1 }
1069     {
1070       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1071       {
1072         \pdfannot_dict_remove:nn
1073         {link/##2}
1074         { Border }
1075       }
1076     }
1077     {
1078       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1079       {

```

```

1080             \pdfannot_dict_put:nnn
1081             {link/##2}
1082             { Border }
1083             { [#1] }
1084         }
1085     }
1086 }
1087 ,pdfborder .initial:n = {0~0~1},
1088 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1089 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1090 {
runborderstyle (setup key) 1091     \keys_define:nn { hyp / setup }
fileborderstyle (setup key) 1092     {
menuborderstyle (setup key) 1093         #1borderstyle .code:n =
1094         {
1095             \tl_if_empty:nTF { ##1 }
1096             {
1097                 \pdfannot_dict_remove:nn
1098                 {link/#2}
1099                 { BS }
1100             }
1101             {
1102                 \pdfannot_dict_put:nnn
1103                 {link/#2}
1104                 { BS }
1105                 { <<##1>> }
1106             }
1107         }
1108     }
1109 }
1110 \keys_define:nn { hyp / setup }
1111 {
1112     ,pdfborderstyle .code:n =
1113     {
1114         \tl_if_empty:nTF { #1 }
1115         {
1116             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1117             {
1118                 \pdfannot_dict_remove:nn
1119                 {link/##2}
1120                 { BS }
1121             }
1122         }
1123         {
1124             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1125             {
1126                 \pdfannot_dict_put:nnn
1127                 {link/##2}
1128                 { BS }

```

```

1129         { <<#1>> }
1130     }
1131 }
1132 }
1133 ,pdfborderstyle .initial:n = {},
1134 }

```

11.4 ocolorlinks

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.

```

1135 \cs_new_protected:Npn \__hyp_ocg_init:
1136 {
1137     \pdf_object_new:nn { __hyp/OCG/View } { dict }
1138     \pdf_object_new:nn { __hyp/OCG/Print } { dict }
1139     \pdf_object_new:nn { __hyp/OCG/config } { dict }
1140     \pdf_object_new:nn { __hyp/OCG/refarray } { array }
1141     \pdf_object_write:nx { __hyp/OCG/refarray }
1142     {
1143         \pdf_object_ref:n { __hyp/OCG/View }
1144         \c_space_tl
1145         \pdf_object_ref:n { __hyp/OCG/Print }
1146     }
1147     \pdf_object_write:nn { __hyp/OCG/View }
1148     {
1149         /Type/OCG
1150         /Name(View)
1151         /Usage
1152         <<
1153         /Print <</PrintState/OFF>>~
1154         /View <</ViewState/ON >>~
1155         >>
1156     }
1157     \pdf_object_write:nn { __hyp/OCG/Print }
1158     {
1159         /Type/OCG
1160         /Name(Print)
1161         /Usage
1162         <<
1163         /Print <</PrintState/ON>>~
1164         /View <</ViewState/OFF>>~
1165         >>
1166     }
1167     \pdfmanagement_add:nnx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/View} }
1168     \pdfmanagement_add:nnx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/Print} }
1169     \pdf_object_write:nx { __hyp/OCG/config }
1170     {
1171         /OFF[\pdf_object_ref:n { __hyp/OCG/Print } ]
1172         /AS[
1173         <<
1174         /Event/View

```

```

1175         /OCGs\c_space_t1 \pdf_object_ref:n { __hyp/OCG/refarray }
1176         /Category[/View]
1177     >>
1178     <<
1179         /Event/Print
1180         /OCGs\c_space_t1 \pdf_object_ref:n { __hyp/OCG/refarray }
1181         /Category[/Print]
1182     >>
1183     <<
1184         /Event/Export
1185         /OCGs\c_space_t1 \pdf_object_ref:n { __hyp/OCG/refarray }
1186         /Category[/Print]
1187     >>
1188     ]
1189 }
1190 \pdfmanagement_add:nnx { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG
1191 \cs_gset:Npn \__hyp_ocg_init: {}
1192 }

```

(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.

```

1193 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1194 {
1195     \hook_gput_code:nnn
1196     {pdfannot/link/#2/begin}
1197     {hyp/ocg}
1198     {
1199         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1200         {
1201             \__hyp_ocg_init:
1202             \group_begin:
1203             \hbox_set:Nw \l__hyp_tmpa_box
1204         }
1205     }
1206     \hook_gput_code:nnn
1207     {pdfannot/link/#2/end}
1208     {hyp/ocg}
1209     {
1210         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1211         {
1212             \hbox_set_end:
1213             \mbox
1214             {
1215                 \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1216                 \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1217                 \pdf_emc:
1218                 \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1219                 \group_begin:
1220                 \color_select:n { hyp/color/#1 }
1221                 \box_use_drop:N \l__hyp_tmpa_box
1222                 \group_end:
1223                 \pdf_emc:
1224             }

```

```

1225         \group_end:
1226     }
1227 }
1228 }

```

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

```

1229 \bool_lazy_or:nnTF
1230 { \pdf_version_compare_p:Nn > {1.4} }
1231 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1232 {
1233     \keys_define:nn { hyp / setup }
1234     {
1235         ,_ocgcolorlinks .meta:n =
1236         {
1237             ocgcolorlink=#1,
1238             ocgcolorurl=#1,
1239             ocgcolorfile=#1,
1240             ocgcolorrun=#1,
1241             ocgcolormenu=#1
1242         }
1243         ,_ocgcolorlinks .default:n = true
1244     }
1245 }
1246 {
1247     \keys_define:nn { hyp / setup }
1248     {
1249         ,_ocgcolorlinks .code:n =
1250         {
1251             \msg_warning:nnxx
1252             { hyp }
1253             { ignore-deprecated-or-unknown-option-in-pdf-version }
1254             { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1255         }
1256     }
1257 }
1258
1259 \keys_define:nn { hyp / setup }
1260 {
1261     ,ocgcolorlinks .choice:
1262     ,ocgcolorlinks / true .meta:n =
1263     {
1264         pdfborder      = {0~0~0},
1265         pdfborderstyle = {},
1266         colorlinks     = false,
1267         _ocgcolorlinks = true
1268     }
1269     ,ocgcolorlinks / false .meta:n =
1270     {
1271         _ocgcolorlinks = false
1272     }
1273     ,ocgcolorlinks .default:n = {true}
1274 }
1275
1276 \seq_map_inline:Nn \c__hyp_annot_types_seq

```

```

1277 {
1278   \bool_lazy_or:nnTF
1279   { \pdf_version_compare_p:Nn > {1.4} }
1280   { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1281   {
1282     \keys_define:nn { hyp / setup }
1283     {
1284       ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1285     }
1286   }
1287   {
1288     \keys_define:nn { hyp / setup }
1289     {
1290       ,ocgcolor#1 .code:n=
1291       {
1292         \msg_warning:nnxx
1293         { hyp }
1294         { ignore-deprecated-or-unknown-option-in-pdf-version }
1295         { ocgcolor#1 }
1296         { \pdf_version_major:.\pdf_version_minor: }
1297       }
1298     }
1299   }
1300 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1301 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1302 {
1303   \keys_define:nn { hyp / setup }
1304   {
1305     ,#1highlight .choices:nn =
1306     { /I, /N, /O, /P }
1307     {
1308       \pdfannot_dict_put:nnn
1309       { link/#2 }
1310       { H }
1311       { ##1 }
1312     }
1313     ,#1highlight / .code:n =
1314     {
1315       \pdfannot_dict_remove:nn
1316       { link/#2 }
1317       { H }
1318     }
1319   }
1320   ,#1highlight / unknown .code:n =
1321   {
1322     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1323     { pdfhighlight }
1324     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1325     { \exp_not:n {##1} }
1326   }

```

```

1327     }
1328   }
1329 }
1330
1331
1332 \keys_define:nn { hyp / setup }
1333 {
1334   ,pdfhighlight .choices:nn =
1335   { /I, /N, /O, /P}
1336   {
1337     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1338     {
1339       \pdfannot_dict_put:nnn
1340       {link/####2}
1341       { H }
1342       { #1 }
1343     }
1344   }
1345   ,pdfhighlight / .code:n =
1346   {
1347     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1348     {
1349       \pdfannot_dict_remove:nn
1350       {link/##2}
1351       { H }
1352     }
1353   }
1354   ,pdfhighlight .initial:n = {/I},
1355   ,pdfhighlight / unknown .code:n =
1356   {
1357     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1358     { pdfhighlight }
1359     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1360     { \exp_not:n {#1} }
1361   }
1362 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1363 \keys_define:nn { hyp / setup }
hideurl (setup key) 1364 {
hidefile (setup key) 1365   hidelinks .meta:n =
hiderun (setup key) 1366   {
hidemenu (setup key) 1367     ,colorlinks      = false
1368     ,ocgcolorlinks = false
1369     ,pdfborder      = { 0~0~0 }
1370     ,pdfborderstyle=
1371   }
1372 }
1373
1374 \seq_map_inline:Nn \c__hyp_annot_types_seq

```



```

1375 {
1376   \keys_define:nn { hyp / setup }
1377   {
1378     hide#1 .meta:n =
1379     {
1380       ,color#1      = false
1381       ,ocgcolor#1   = false
1382       ,#1border     = { 0~0~0 }
1383       ,#1borderstyle =
1384     }
1385   }
1386 }

```

11.7 color schemes and settings

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

`colorscheme (setup key)`

```

1387 \keys_define:nn { hyp / setup }
1388 {
1389   colorscheme .code:n =
1390   {
1391     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1392     {
1393       \keys_set:nn { hyp / setup }
1394       {
1395         ##1 = ##2
1396       }
1397     }
1398   }
1399 }
1400 \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)`

```

1401 \keys_define:nn { hyp / setup }
1402 {
1403   ,unicode      .code:n = {}
1404   ,pdfencoding  .code:n = {}
1405   ,pdfversion   .code:n =
1406   {
1407     \msg_warning:nn { hyp }{ pdfversion-disabled }
1408   }
1409 }
1410 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
  debug (setup key) 1411 \keys_define:nn { hyp / setup }
  draft (setup key) 1412 {
  final (setup key) 1413   ,verbose .choice:
1414   ,verbose / true .code:n = { \Hy@verbosetrue}
1415   ,verbose / false .code:n = { \Hy@verbosefalse}
1416   ,verbose .default:n = {true}
1417   ,debug .meta:n = {verbose=#1}
1418   ,debug .default:n = {true}
1419 }
1420 \keys_define:nn { hyp / setup }
1421 {
1422   ,draft .code:n =
1423   {
1424     \Hy@drafttrue
1425     \PassOptionsToPackage{draft}{bookmark}
1426   }
1427   ,final .code:n =
1428   {
1429     \Hy@finaltrue
1430     \PassOptionsToPackage{final}{bookmark}
1431   }
1432 }

extension (setup key)
hypertextnames (setup key) 1433 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1434 {
  pageanchor (setup key) 1435   ,extension .tl_set:N = \XR@ext
  linktoc (setup key) 1436   ,extension .initial:n= pdf
  linktocpage (setup key) 1437   ,hypertextnames .choice:
  plainpages (setup key) 1438   ,hypertextnames / true .code:n = { \Hy@hypertextnamestrue}
localanchorname (setup key) 1439   ,hypertextnames / false .code:n = { \Hy@hypertextnamesfalse}
linkfileprefix (setup key) 1440   ,hypertextnames .default:n = {true}
1441   ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1442   ,localanchorname .choice:
1443   ,localanchorname / true .code:n = { \Hy@localanchorname true }
1444   ,localanchorname / false .code:n = { \Hy@localanchorname false }
1445   ,localanchorname .default:n = {true}
1446   ,naturalnames .choice:
1447   ,naturalnames / true .code:n = { \Hy@naturalnames true}
1448   ,naturalnames / false .code:n = { \Hy@naturalnames false}
1449   ,naturalnames .default:n = {true}
1450   ,pageanchor .choice:
1451   ,pageanchor / true .code:n = { \Hy@pageanchortrue}
1452   ,pageanchor / false .code:n = { \Hy@pageanchorfalse}
1453   ,pageanchor .default:n = {true}
1454   ,plainpages .choice:
1455   ,plainpages / true .code:n = { \Hy@plainpagestrue}
1456   ,plainpages / false .code:n = { \Hy@plainpagesfalse}
1457   ,plainpages .default:n = {true}

```

```

1458 }
1459
1460 \keys_define:nn { hyp / setup }
1461 {
1462   ,linktoc .choices:nn = { none, section, all, page }
1463   {
1464     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1465   }
1466   ,linktoc / unknown .code:n =
1467   {
1468     \msg_warning:nnxxx { hyp } { unknown-choice }
1469     { linktoc }
1470     { none, section, all, page }
1471     { \exp_not:n {#1} }
1472   }
1473   ,linktocpage .choice:
1474   ,linktocpage / true .meta:n = {linktoc=page}
1475   ,linktocpage / false .meta:n = {linktoc=section}
1476   ,linktocpage .default:n = true
1477 }
1478

```

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

```

url (setup key) 1479 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1480 {
menu (setup key) 1481   \keys_define:nn { hyp / setup }
run (setup key) 1482   {
1483     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1484   }
1485 }

```

```

1486 \keys_define:nn { hyp / setup }
1487 {
1488   ,baseurl .code:n =
1489   {
1490     \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1491     \tl_if_empty:NTF \l__hyp_tmpa_tl
1492     {
1493       \pdfmanagement_remove:nn {Catalog} { URI }
1494     }
1495     {
1496       \pdfmanagement_add:nnx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1497     }
1498   }
1499   %only false does something ...
1500   ,bookmarks .choice:
1501   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1502   ,bookmarks / true .code:n = {}
1503   ,bookmarks .default:n = {true}
1504   ,bookmarksnumbered .choice:
1505   ,bookmarksnumbered / false .code:n = { \Hy@bookmarksnumberedfalse }
1506   ,bookmarksnumbered / true .code:n = { \Hy@bookmarksnumberedtrue }
1507   ,bookmarksnumbered .default:n = {true}
1508   ,bookmarksopen .choice:

```

```

1509 ,bookmarksopen / false .code:n = { \Hy@bookmarksopenfalse }
1510 ,bookmarksopen / true .code:n = { \Hy@bookmarksopentruetrue }
1511 ,bookmarksopen .default:n = {true}
1512 ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1513 ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1514 ,pdfcenterwindow .choice:
1515 ,pdfcenterwindow / false .code:n =
1516 {
1517     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1518 }
1519 ,pdfcenterwindow / true .code:n =
1520 {
1521     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1522 }
1523 ,pdfcenterwindow / .code:n =
1524 {
1525     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1526 }
1527 ,pdfcenterwindow / unknown .code:n =
1528 {
1529     \msg_warning:nnxx { hyp } { no-bool }
1530     { pdfcenterwindow }
1531     { \exp_not:n {#1} }
1532 }
1533 ,pdfcenterwindow .default:n = true
1534 ,pdfdirection .choice:
1535 ,pdfdirection / L2R .code:n =
1536 {
1537     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1538 }
1539 ,pdfdirection / R2L .code:n =
1540 {
1541     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1542 }
1543 ,pdfdirection / .code:n =
1544 {
1545     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1546 }
1547 ,pdfdirection / unknown .code:n =
1548 {
1549     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1550     { pdfdirection }
1551     { L2R , R2L }
1552     { \exp_not:n {#1} }
1553 }
1554 ,pdfdisplaydoctitle .choice:
1555 ,pdfdisplaydoctitle / false .code:n =
1556 {
1557     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1558 }
1559 ,pdfdisplaydoctitle / true .code:n =
1560 {
1561     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1562 }

```

```

1563 ,pdffdisplaydoctitle .default:n = true
1564 ,pdfduplex .choices:nn =
1565 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1566 {
1567     \pdf_version_compare:NnTF > {1.6}
1568     {
1569         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1570         { PrintDuplex } { /#1 }
1571     }
1572     {
1573         \msg_warning:nnxx
1574         {hyp}
1575         {ignore-deprecated-or-unknown-option-in-pdf-version}
1576         {pdfduplex}
1577         {\pdf_version:}
1578     }
1579 }%
1580 ,pdfduplex / .code:n =
1581 {
1582     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1583 }
1584 ,pdfduplex / unknown .code:n =
1585 {
1586     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1587     { pdfduplex }
1588     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1589     { \exp_not:n {#1} }
1590 }
1591 ,pdffitwindow .choice:
1592 ,pdffitwindow / false .code:n =
1593 {
1594     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1595 }
1596 ,pdffitwindow / true .code:n =
1597 {
1598     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1599 }
1600 ,pdffitwindow / .code:n =
1601 {
1602     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1603 }
1604 ,pdffitwindow .default:n = true
1605 ,pdffitwindow / unknown .code:n =
1606 {
1607     \msg_warning:nnxx { hyp } { no-bool }
1608     { pdffitwindow }
1609     { \exp_not:n {#1} }
1610 }
1611 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1612 ,pdflinkmargin .initial:n = {1pt}
1613 ,pdfmenubar .choice:
1614 ,pdfmenubar / true .code:n =
1615 {
1616     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }

```

```

1617     }
1618 ,pdfmenubar / false .code:n =
1619     {
1620         \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1621         { HideMenubar } { true }
1622     }
1623 ,pdfmenubar / .code:n =
1624     {
1625         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1626     }
1627 ,pdfmenubar .default:n = true
1628 ,pdfmenubar / unknown .code:n =
1629     {
1630         \msg_warning:nnxx { hyp } { no-bool }
1631         { pdfmenubar }
1632         { \exp_not:n {#1} }
1633     }
1634 ,pdfnewwindow .choice:
1635 ,pdfnewwindow / true .code:n =
1636     {
1637         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1638         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1639     }
1640 ,pdfnewwindow / false .code:n =
1641     {
1642         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1643         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1644     }
1645 ,pdfnewwindow / .code:n =
1646     {
1647         \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1648         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1649     }
1650 ,pdfnonfullscreenpagemode .choices:nn =
1651     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1652     {
1653         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1654         { NonFullScreenPageMode } {/#1}
1655     }
1656 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1657     {
1658         \pdf_version_compare:NnTF < {1.6}
1659         {
1660             %message
1661         }
1662         {
1663             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1664             { NonFullScreenPageMode } {/UseAttachments}
1665         }
1666     }
1667 ,pdfnonfullscreenpagemode / .code:n =
1668     {
1669         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1670     }

```

```

1671 ,pdfnonfullscreenpagemode / unknown .code:n =
1672 {
1673     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1674     { pdfnonfullscreenpagemode }
1675     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1676     { \exp_not:n {#1} }
1677 }
1678 ,pdfnumcopies .code:n =
1679 {
1680     \pdf_version_compare:NnTF > {1.6}
1681     {
1682         \tl_if_empty:nTF {#1}
1683         {
1684             \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1685         }
1686         {
1687             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1688             {NumCopies}{#1}
1689         }
1690     }
1691     {
1692         \msg_warning:nnxx
1693         {hyp}
1694         {ignore-deprecated-or-unknown-option-in-pdf-version}
1695         {pdfnumcopies}
1696         {\pdf_version:}
1697     }
1698 }
1699 ,pdfpagelayout .choices:nn =
1700 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1701 { \pdfmanagement_add:nnx {Catalog} { PageLayout }{ /#1 } }
1702 ,pdfpagelayout / .code:n =
1703 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1704 ,pdfpagelayout / unknown .code:n =
1705 {
1706     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1707     { pdfpagelayout }
1708     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1709     { \exp_not:n {#1} }
1710 }
1711 ,pdfpagemode .choices:nn =
1712 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1713 { \pdfmanagement_add:nnx {Catalog} { PageMode }{ /#1 } }
1714 ,pdfpagemode / UseAttachments .code:n =
1715 {
1716     \pdf_version_compare:NnTF > {1.5}
1717     {
1718         \pdfmanagement_add:nnx {Catalog} { PageMode }{ /UseAttachments }
1719     }
1720     {
1721         \msg_warning:nnxx
1722         {hyp}
1723         {ignore-deprecated-or-unknown-value-in-pdf-version}
1724         {UseAttachments}

```

```

1725         {\pdf_version:}
1726     }
1727 }
1728 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1729 ,pdfpagemode / unknown .code:n =
1730 {
1731     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1732     { pdfpagemode }
1733     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1734     { \exp_not:n {#1} }
1735 }
1736 ,pdfpagescrop .code:n =
1737 {
1738     \tl_if_empty:nTF {#1} %or blank?
1739     {
1740         \pdfmanagement_remove:nn {Pages} { CropBox }
1741     }
1742     {
1743         \pdfmanagement_add:nnx {Pages} { CropBox } { [#1] }
1744     }
1745 }
1746 ,pdfpicktraybypdfsize .choice:
1747 ,pdfpicktraybypdfsize / true .code:n =
1748 {
1749     \pdf_version_compare:NnTF > {1.6}
1750     {
1751         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1752         { PickTrayByPDFSize } { true }
1753     }
1754     {
1755         \msg_warning:nnxx
1756         {hyp}
1757         {ignore-deprecated-or-unknown-option-in-pdf-version}
1758         {pdfpicktraybypdfsize}
1759         {\pdf_version:}
1760     }
1761 }
1762 ,pdfpicktraybypdfsize / false .code:n =
1763 {
1764     \pdf_version_compare:NnTF > {1.6}
1765     {
1766         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1767         { PickTrayByPDFSize } { false }
1768     }
1769     {
1770         \msg_warning:nnxx
1771         {hyp}
1772         {ignore-deprecated-or-unknown-option-in-pdf-version}
1773         {pdfpicktraybypdfsize}
1774         {\pdf_version:}
1775     }
1776 }
1777 ,pdfpicktraybypdfsize / .code:n =
1778 {

```



```

1779     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1780 }
1781 ,pdfpicktraybypdfsize / unknown .code:n =
1782 {
1783     \msg_warning:nxxx { hyp } { no-bool }
1784     { picktraybypdfsize }
1785     { \exp_not:n {#1} }
1786 }
1787 ,pdfprintarea .choices:nn =
1788 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1789 {
1790     \pdf_version_compare:NnTF < {2.0}
1791     {
1792         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1793         { PrintArea } { /#1 }
1794     }
1795     {
1796         \msg_warning:nxxx
1797         {hyp}
1798         {ignore-deprecated-or-unknown-option-in-pdf-version}
1799         {pdfprintarea}
1800         {\pdf_version:}
1801     }
1802 }%
1803 ,pdfprintarea / .code:n =
1804 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1805 ,pdfprintarea / unknown .code:n =
1806 {
1807     \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1808     { pdfprintarea }
1809     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1810     { \exp_not:n {#1} }
1811 }
1812 ,pdfprintclip .choices:nn =
1813 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1814 {
1815     \pdf_version_compare:NnTF < {2.0}
1816     {
1817         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1818         { PrintClip } { /#1 }
1819     }
1820     {
1821         \msg_warning:nxxx
1822         {hyp}
1823         {ignore-deprecated-or-unknown-option-in-pdf-version}
1824         {pdfprintclip}
1825         {\pdf_version:}
1826     }
1827 }%
1828 ,pdfprintclip / .code:n =
1829 {
1830     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1831 }
1832 ,pdfprintclip / unknown .code:n =

```

```

1833 {
1834     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1835     { pdfprintclip }
1836     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1837     { \exp_not:n {#1} }
1838 }
1839 ,pdfprintpagerange .code:n =
1840 {
1841     \pdf_version_compare:NnTF > {1.6}
1842     {
1843         \tl_if_empty:nTF { #1}
1844         {
1845             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1846             { PrintPageRange }
1847         }
1848         {
1849             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1850             {PrintPageRange}{[#1]}
1851         }
1852     }
1853     {
1854         \msg_warning:nnxx
1855         {hyp}
1856         {ignore-deprecated-or-unknown-option-in-pdf-version}
1857         {pdfprintpagerange}
1858         {\pdf_version:}
1859     }
1860 }
1861 ,pdfprintscaling .choices:nn =
1862 { None, AppDefault }
1863 {
1864     \pdf_version_compare:NnTF > {1.5}
1865     {
1866         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1867         { PrintScaling } { /#1 }
1868     }
1869     {
1870         \msg_warning:nnxx
1871         {hyp}
1872         {ignore-deprecated-or-unknown-option-in-pdf-version}
1873         {pdfprintscaling}
1874         {\pdf_version:}
1875     }
1876 }%
1877 ,pdfprintscaling / .code:n =
1878 {
1879     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1880 }
1881 ,pdfprintscaling / unknown .code:n =
1882 {
1883     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1884     { pdfprintarea }
1885     { None, AppDefault }
1886     { \exp_not:n {#1} }

```

```

1887     }
1888 ,pdfremotestartview .code:n =
1889 {
1890     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1891     \exp_args:NNV
1892     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1893     {
1894         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1895     }
1896     {
1897         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1898         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1899     }
1900 }
1901 ,pdfremotestartview .initial:n = {Fit}
1902 % pdfstartpage is special as it shares code with pdfstartview
1903 ,pdfstartpage .code:n =
1904 {
1905     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1906     \bool_if:nTF
1907     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1908     {
1909         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1910     }
1911     {
1912         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1913         {
1914             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1915         }
1916     }
1917 }
1918 ,pdfstartpage .initial:n =1
1919 ,pdfstartview .code:n =
1920 {
1921     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1922     \exp_args:NNV
1923     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1924     {
1925         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1926     }
1927     {
1928         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1929         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1930     }
1931     \bool_if:nTF
1932     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1933     {
1934         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1935     }
1936     {
1937         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1938         {
1939             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1940         }

```

```

1941     }
1942 }
1943 ,pdfstartview .initial:n = Fit
1944 ,pdftoolbar .choice:
1945 ,pdftoolbar / true .code:n =
1946 {
1947     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1948 }
1949 ,pdftoolbar / false .code:n =
1950 {
1951     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1952     { HideToolbar } { true }
1953 }
1954 ,pdftoolbar / true .code:n =
1955 {
1956     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1957 }
1958 ,pdftoolbar .default:n = true
1959 ,pdftoolbar / unknown .code:n =
1960 {
1961     \msg_warning:nxxx { hyp } { no-bool }
1962     { pdftoolbar }
1963     { \exp_not:n {#1} }
1964 }
1965 % pdfview see below.
1966 ,pdfviewarea .choices:nn =
1967 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1968 {
1969     \pdf_version_compare:NnTF < {2.0}
1970     {
1971         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1972         { ViewArea } { /#1 }
1973     }
1974     {
1975         \msg_warning:nxxx
1976         {hyp}
1977         {ignore-deprecated-or-unknown-option-in-pdf-version}
1978         {pdfviewarea}
1979         {\pdf_version:}
1980     }
1981 }%
1982 ,pdfviewarea / .code:n =
1983 {
1984     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1985 }
1986 ,pdfviewarea / unknown .code:n =
1987 {
1988     \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1989     { pdfviewarea }
1990     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1991     { \exp_not:n {#1} }
1992 }
1993 ,pdfviewclip .choices:nn =
1994 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }

```

```

1995     {
1996         \pdf_version_compare:NnTF < {2.0}
1997         {
1998             \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1999             { ViewClip } { /#1 }
2000         }
2001         {
2002             \msg_warning:nnxx
2003             {hyp}
2004             {ignore-deprecated-or-unknown-option-in-pdf-version}
2005             {pdfviewclip}
2006             {\pdf_version:}
2007         }
2008     }%
2009 ,pdfviewclip / .code:n =
2010 {
2011     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
2012 }
2013 ,pdfviewclip / unknown .code:n =
2014 {
2015     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
2016     { pdfviewclip }
2017     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2018     { \exp_not:n {#1} }
2019 }
2020 ,pdfwindowui .choice:
2021 ,pdfwindowui / true .code:n =
2022 {
2023     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2024 }
2025 ,pdfwindowui / false .code:n =
2026 {
2027     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2028     { HideWindowUI } { true }
2029 }
2030 ,pdfwindowui / .code:n =
2031 {
2032     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2033 }
2034 ,pdfwindowui / unknown .code:n =
2035 {
2036     \msg_warning:nnxx { hyp } { no-bool }
2037     { pdfwindowui }
2038     { \exp_not:n {#1} }
2039 }
2040 ,pdfwindowui .default:n = true
2041 }

```

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

```

2042 \keys_define:nn { hyp / setup }
2043 {
2044     ,pdfview .code:n =
2045     {
2046         \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}

```

```

2047 \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
2048 {
2049   { xyz }
2050   {
2051     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
2052     {
2053       \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2054       \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2055       {
2056         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2057       }
2058       {
2059         \tl_set:Nx \l__hyp_dest_pdfview_tl
2060         {
2061           \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2062         }
2063       }
2064     }
2065     {
2066       \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2067     }
2068   }
2069   { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2070   { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2071   { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2072   { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2073   { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2074   { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2075   { fitr }
2076   {
2077     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2078     {
2079       \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2080     }
2081     {
2082       %ensure 4 values ...
2083       \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2084       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2085       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2086       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2087       \hbox_set_to_wd:Nnn \l__hyp_dest_box
2088       {
2089         \fp_eval:n
2090         {
2091           round
2092           (
2093             abs
2094             (
2095               \seq_item:Nn\l__hyp_tmpa_seq{4}
2096               -
2097               (\seq_item:Nn\l__hyp_tmpa_seq{2})
2098             ),
2099           3
2100         )

```

```

2101         }bp
2102     }{}
2103     \box_set_dp:Nn \l__hyp_dest_box
2104     {
2105         \fp_eval:n
2106         {
2107             round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2108         }bp
2109     }
2110     \box_set_ht:Nn \l__hyp_dest_box
2111     {
2112         \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2113     }
2114 }
2115 }
2116 }
2117 {
2118     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2119     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2120 }
2121 }
2122 ,pdfview .initial:n = {xyz}
2123 }

```

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.

```

2124 \keys_define:nn { hyp / setup }
2125 {
2126     ,pdflang .code:n =
2127     {
2128         \tl_if_empty:nTF { #1 }
2129         {
2130             \pdfmanagement_remove:nn {Catalog} { Lang }
2131         }
2132         {
2133             \pdfmanagement_add:nnx {Catalog} { Lang } { (#1) }
2134         }
2135         \__hyp_store_metadata:nn {pdflang}{#1}
2136     }
2137 }

```

12.3.1 “info keys”

pdfauthor (*setup key*) The keys store their value also in the metadata container, so that hyperxmp can use them. Creator and Producer can’t be removed with the pdfmanagement, but we allow

pdfcreator (*setup key*) to set an empty value.

pdfsubject (*setup key*)

pdfproducer (*setup key*)

pdfkeywords (*setup key*)

```

2138 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2139 {
2140     \keys_define:nn { hyp / setup }

```

```

2141 {
2142     pdf#1 .code:n =
2143     {
2144         \tl_if_blank:nTF {##1}
2145         {
2146             \str_case:nnF { #1 }
2147             {
2148                 {creator}
2149                 {
2150                     \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2151                     \pdfmanagement_add:nnx {Info}{Creator}{()}
2152                 }
2153                 {producer}
2154                 {
2155                     \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2156                     \pdfmanagement_add:nnx {Info}{Producer}{()}
2157                 }
2158             }
2159             {
2160                 \pdfmanagement_remove:nn {Info}{#2}
2161             }
2162         }
2163         {
2164             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2165             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2166             {
2167                 \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2168             }
2169         }
2170         \__hyp_store_metadata:nn {pdf#1}{##1}
2171     }
2172 }
2173 \keys_define:nn { hyp / info }
2174 {
2175     #2 .code:n =
2176     {
2177         \tl_if_blank:nTF {##1}
2178         {
2179             \pdfmanagement_remove:nn {Info}{#2}
2180         }
2181         {
2182             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2183             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2184             {
2185                 \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2186             }
2187         }
2188         \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2189     }
2190 ,unknown .code:n =
2191 {
2192     \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2193     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2194     {

```



```

2195         \exp_args:Nno
2196         \pdfmanagement_add:nnx {Info}
2197         { \l_keys_key_str } {\l__hyp_tmpa_str}
2198     }
2199 }
2200 }
2201 }
2202 \__hyp_setup_info_key:nn {author} {Author}
2203 \__hyp_setup_info_key:nn {title} {Title}
2204 \__hyp_setup_info_key:nn {producer} {Producer}
2205 \__hyp_setup_info_key:nn {creator} {Creator}
2206 % ignored key: addtopdfcreator
2207 \__hyp_setup_info_key:nn {subject} {Subject}
2208 \__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.

```

2209 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2210 {
2211     \keys_define:nn { hyp / setup }
2212     {
2213         pdf#1 .code:n =
2214         {
2215             \tl_if_blank:nTF {##1}
2216             {
2217                 \pdfmanagement_remove:nn {Info}{#2}
2218             }
2219             {
2220                 \pdfmanagement_add:nnx {Info}{#2}{(##1)}
2221             }
2222             \__hyp_store_metadata:nn {pdf#1}{##1}
2223         }
2224     }
2225     \keys_define:nn { hyp / info }
2226     {
2227         #2 .code:n =
2228         {
2229             \tl_if_blank:nTF {##1}
2230             {
2231                 \pdfmanagement_remove:nn {Info}{#2}
2232             }
2233             {
2234                 \pdfmanagement_add:nnx {Info}{#2}{(##1)}
2235             }
2236             \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2237         }
2238     }
2239 }
2240
2241 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2242 \__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 ...

```

2243 \keys_define:nn { hyp / setup }

```

```

2244 {
2245   ,pdftrapped .code:n =
2246   {
2247     \exp_args:Nne
2248     \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2249   }
2250   ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2251   {
2252     \pdfmanagement_add:nnx {Info}{Trapped}
2253     {/
2254       \str_uppercase:f { \str_head:n { #1 } }
2255       \str_lowercase:f { \str_tail:n { #1 } }
2256     }
2257     \__hyp_store_metadata:nx {pdftrapped}
2258     {
2259       \str_uppercase:f { \str_head:n { #1 } }
2260       \str_lowercase:f { \str_tail:n { #1 } }
2261     }
2262   }
2263   ,_pdftrapped / unknown .code:n =
2264   {
2265     \msg_warning:nnxxx { hyp } { unknown-choice }
2266     { pdftrapped }
2267     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2268     { \exp_not:n {#1} }
2269   }
2270 }

```

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

```

2271 \keys_define:nn { hyp / setup }
2272 {
2273   pdfinfo .code:n =
2274   {
2275     \keys_set:nn { hyp / info } { #1 }
2276   }
2277 }

```

Now we set some default values

```

2278 \keys_set:nn { hyp / setup } {pdfcreator = LaTeX-with-hyperref}
2279 \keys_set:nn { hyp / setup } {pdfauthor = }
2280 \keys_set:nn { hyp / setup } {pdftitle = }
2281 \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:

```

2282 \clist_map_inline:nn
2283 {
2284   ,pdfcopyright
2285   ,pdftype
2286   ,pdflicenseurl
2287   ,pdfauthortitle

```

```

2288 ,pdfcaptionwriter
2289 ,pdfmetalang
2290 ,pdfapart
2291 ,pdfaconformance
2292 ,pdfuapart
2293 ,pdfxstandard
2294 ,pdfsource
2295 ,pdfdocumentid
2296 ,pdfinstanceid
2297 ,pdfversionid
2298 ,pdfrendition
2299 ,pdfpublication
2300 ,pdfpubtype
2301 ,pdfbytes
2302 ,pdfnumpages
2303 ,pdfissn
2304 ,pdfeissn
2305 ,pdfisbn
2306 ,pdfbookedition
2307 ,pdfpublisher
2308 ,pdfvolumenum
2309 ,pdfissuenum
2310 ,pdfpagerange
2311 ,pdfdoi
2312 ,pdfurl
2313 ,pdfidentifier
2314 ,pdfsubtitle
2315 ,pdfpubstatus
2316 ,pdfcontactaddress
2317 ,pdfcontactcity
2318 ,pdfcontactregion
2319 ,pdfcontactpostcode
2320 ,pdfcontactcountry
2321 ,pdfcontactphone
2322 ,pdfcontactemail
2323 ,pdfcontacturl
2324 ,pdfdate
2325 }
2326 {
2327   \keys_define:nn { hyp / setup }
2328   {
2329     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2330   }
2331 }
2332
2333 \str_if_eq:eeT {A}{\pdfmeta_standard_item:n{type}}
2334 {
2335   \exp_args:Nnx
2336   \keys_set:nn {hyp / setup}
2337   {
2338     ,pdfapart = \pdfmeta_standard_item:n {level}
2339     ,pdfaconformance = \pdfmeta_standard_item:n {conformance}
2340   }
2341 }

```

12.5 Transitions

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

```

2342 \keys_define:nn { hyp / setup }
2343 {
2344   pdfpageduration .code:n =
2345   {
2346     \tl_if_blank:nTF { #1 }
2347     {
2348       \pdfmanagement_remove:nn {Page}{Dur}
2349     }
2350     {
2351       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2352     }
2353   }
2354 }

```

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}}
```

The keys allowed in the argument of `pdfpagetransition` are

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

```

2355 \keys_define:nn { hyp / setup }
2356 {
2357   pdfpagetransition .code:n =
2358   {
2359     \tl_if_blank:nTF {#1}
2360     {
2361       \pdfmanagement_remove:nn {Page}{Trans}
2362     }

```

```

2363     {
2364         \group_begin:
2365         \keys_set:nn { hyp / trans }{style=R,#1}
2366         \pdf_object_unnamed_write:nx { dict }
2367         {
2368             \pdfdict_use:n {l__hyp_page/Trans}
2369         }
2370         \pdfmanagement_add:nnx {Page}{Trans}{\pdf_object_ref_last:}
2371         \group_end:
2372     }
2373 }
2374 }
2375 \keys_define:nn { hyp / trans }
2376 {
2377     ,style .choices:nn =
2378     {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2379     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2380     ,style / unknown .code:n =
2381     {
2382         \msg_warning:nnxxx { hyp } { unknown-choice }
2383         { trans / style }
2384         { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2385         { \exp_not:n {#1} }
2386     }
2387     ,duration .code:n =
2388     {
2389         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}
2390     }
2391     ,direction .choices:nn =
2392     {H,V}
2393     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2394     ,direction .choices:nn =
2395     {0,90,180,270,315}
2396     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2397     ,direction / None .code:n =
2398     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2399     ,direction / unknown .code:n =
2400     {
2401         \msg_warning:nnxxx { hyp } { unknown-choice }
2402         { trans / direction }
2403         {
2404             H~(horizontal,~only~Split,~Blinds),
2405             V~(vertical,~only~Split,~Blinds),
2406             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2407             90~(bottom~to~top,~only~Wipe),
2408             180~(right~to~left,~only~Wipe),
2409             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2410             315~(top~left~to~bottom,~only~Glitter),
2411             None~(only~Fly)
2412         }
2413         { \exp_not:n {#1} }
2414     }
2415     ,motion .choices:nn =
2416     {I,0}

```

```

2417 { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2418 ,motion / unknown .code:n =
2419 {
2420   \msg_warning:nnxxx { hyp } { unknown-choice }
2421   { trans / motion }
2422   { I~(inwards) , O~(outwards) }
2423   { \exp_not:n {#1} }
2424 }
2425 ,scale .code:n =
2426 { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2427 ,opaque .choices:nn = {true,false}
2428 { \pdfdict_put:nnn { l__hyp_page/Trans }{ B } { #1 } }
2429 ,opaque / unknown .code:n =
2430 {
2431   \msg_warning:nnxxx { hyp } { unknown-choice }
2432   { trans / B }
2433   { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2434   { \exp_not:n {#1} }
2435 }
2436 % try to set unknown keys as style
2437 ,unknown .code:n =
2438 {
2439   % warning ...
2440   \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2441 }
2442 }

```

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

```

2443 \keys_set:known:nv{hyp/setup}{opt@hyperref.sty}

```

Unfinished Form field code

```

2444 \NewDocumentCommand \MakeFieldObject { m m }
2445 {
2446   \pdfxform_new:nnn { #2 }{ } { #1 }
2447 }
2448
2449
2450 \prop_new:N \g__hyp_AcroForm_CoFields_prop
2451 \prop_new:N \g__hyp_AcroForm_Fields_prop
2452
2453 \let\HyField@afields\ltx@empty
2454 \let\HyField@cofields\ltx@empty
2455 %% UF test for old pdftex removed
2456 %\let\HyField@AuxAddToFields\ltx@gobble
2457 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2458 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2459
2460 % the value doesn't matter, but with a prop we avoid duplicates and it is
2461 % clearly faster than removing them from a sequence
2462 \def\HyField@AuxAddToFields#1
2463 {
2464   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2465 }%
2466
2467 %fields with empty key get a value too -- lets hope that

```

```

2468 %this give the expected behaviour
2469 \def\HyField@AuxAddToCoFields #1 #2
2470 {
2471   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2472 }
2473
2474 \Hy@AtBeginDocument
2475 {
2476   \if@filesw
2477     \immediate\write\@mainaux{%
2478       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2479     }%
2480     \immediate\write\@mainaux{%
2481       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2482     }%
2483   \fi
2484   \let\HyField@AfterAuxOpen\@firstofone
2485 }%
2486
2487 \def\HyField@AddToFields
2488 {
2489   \exp_args:Nx\HyField__hypAddToFields
2490   {
2491     \pdfannot_box_ref_last:
2492   }
2493   \ifx\Fld@calculate@code\ltx@empty
2494   \else
2495     \begingroup
2496     \Hy@safe@activetrue
2497     \edef\Hy@temp{%
2498       \endgroup
2499       \if@filesw
2500         \write\@mainaux
2501         {
2502           \string\HyField@AuxAddToCoFields
2503           {
2504             \Fld@calculate@sortkey
2505           }
2506           {
2507             \pdfannot_box_ref_last:
2508           }
2509         }
2510       \fi
2511     }%
2512     \Hy@temp
2513   \fi
2514 }%
2515
2516 \def\HyField__hypAddToFields#1{
2517   \HyField@AfterAuxOpen{%
2518     \if@filesw
2519       \write\@mainaux{%
2520         \string\HyField@AuxAddToFields{#1}%
2521       }%

```

```

2522     \fi
2523   }%
2524 }%
2525
2526 \ExplSyntaxOff
2527 \ExplSyntaxOn
2528
2529 \def \@Form[#1]
2530 {
2531   \kvsetkeys{Form}{#1}
2532   \pdf@ifdraftmode{}
2533   {
2534     \Hy@FormObjects
2535     \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2536     {
2537       \pdfmanagement_add:nnx { Catalog / AcroForm } { Fields }{##1}
2538       %\pdfmanagement_show:n { Catalog / AcroForm }
2539     }
2540     \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2541     {
2542       \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2543       {
2544         \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2545       }
2546       \seq_sort:Nn \l__hyp_tmpa_seq
2547       {
2548         \str_compare:nNnTF {##1} > {##2}
2549         { \sort_return_swapped: }
2550         { \sort_return_same: }
2551       }
2552       \seq_map_inline:Nn \l__hyp_tmpa_seq
2553       {
2554         \pdfmanagement_add:nnx { Catalog / AcroForm }
2555         { CO }
2556         {
2557           \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2558         }
2559       }
2560     }
2561     \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2562     {ZaDb} {\pdf_object_ref:n {__hyp/Font/ZaDb} }
2563     \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2564     {Helv} {\pdf_object_ref:n {__hyp/Font/Helv} }
2565     \pdfmanagement_add:nnx {Catalog /AcroForm}
2566     {DA}{(/Helv~10~Tf~0~g)}
2567     \pdfmeta_standard_verify:nTF {form_no_NeedAppearance}
2568     {
2569       \legacy_if:nT { HyField@NeedAppearances }
2570       {
2571         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2572       }
2573     }
2574     {
2575       \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}

```



```

2576     }
2577   }
2578 }
2579 \ExplSyntaxOff
2580 \let\@endForm\ltx@empty
2581 \let\HyAnn@AbsPageLabel\ltx@empty
2582 \let\Fld@pageobjref\ltx@empty
2583
2584 \ExplSyntaxOn
2585 \newcount\HyAnn@Count
2586 \HyAnn@Count=\ltx@zero
2587 \def\HyAnn@AbsPageLabel
2588 {
2589   \global\advance\HyAnn@Count by\ltx@one
2590   %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2591   %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}%
2592   %\zref@refused{HyAnn@\the\HyAnn@Count}%
2593   \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
2594   \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
2595 }%
2596 \def\Fld@pageobjref
2597 {
2598   \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
2599   {
2600     /P~\pdf_pageobject_ref:n
2601     {
2602       \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
2603     }
2604   }
2605 }
2606 \ExplSyntaxOff
2607 \ExplSyntaxOn
2608 %% check if the attr should be set through
2609 %% hooks.
2610 %% check if options are missing.
2611 \def\@TextField[#1]#2{% parameters, label
2612   \def\Fld@name{#2}%
2613   \let\Fld@default\ltx@empty
2614   \let\Fld@value\@empty
2615   \def\Fld@width{\DefaultWidthofText}%
2616   \def\Fld@height{%
2617     \ifFld@multiline
2618       \DefaultHeightofTextMultiline
2619     \else
2620       \DefaultHeightofText
2621     \fi
2622   }%
2623   \begingroup
2624     \expandafter\HyField@SetKeys\expandafter{%
2625       \DefaultOptionsofText,#1%
2626     }%
2627     \PDFForm@Name
2628     \HyField@FlagsText
2629     \ifFld@hidden\def\Fld@width{1sp}\fi

```

```

2630 \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2631 \LayoutTextField{#2}{%
2632 \leavevmode
2633 \HyAnn@AbsPageLabel
2634 \Hy@escapeform\PDFForm@Text
2635 \pdfannot_box:nnnn
2636 {\Fld@width}
2637 {\Fld@height}
2638 {Opt} %is this correct?
2639 {\PDFForm@Text}
2640 \MakeTextField{\Fld@width}{\Fld@height}
2641 \HyField@AddToFields
2642 }%
2643 \endgroup
2644 }
2645 \providecommand\@curropt{}
2646 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2647 \def\Fld@name{#2}
2648 \let\Fld@default\relax
2649 \let\Fld@value\relax
2650 \def\Fld@width{\DefaultWidthofChoiceMenu}
2651 \def\Fld@height{\DefaultHeightofChoiceMenu}
2652 \begingroup
2653 \Fld@menulength=0 %
2654 \@tempdima\z@
2655 \clist_map_variable:nNn { #3 } \@curropt
2656 %\@for\@curropt:=#3\do
2657 {%
2658 \expandafter\Fld@checkequals\@curropt==\%
2659 \Hy@StepCount\Fld@menulength
2660 \settowidth{\@tempdimb}{\@currDisplay}%
2661 \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2662 }%
2663 \advance\@tempdima by~15\p@
2664 \begingroup
2665 \HyField@SetKeys{#1}
2666 \edef\x{\endgroup
2667 \noexpand\expandafter
2668 \noexpand\HyField@SetKeys
2669 \noexpand\expandafter{%
2670 \expandafter\noexpand\curname DefaultOptionsof%
2671 \ifFld@radio
2672 Radio%
2673 \else
2674 \ifFld@combo
2675 \ifFld@popdown
2676 PopdownBox%
2677 \else
2678 ComboBox%
2679 \fi
2680 \else
2681 ListBox%
2682 \fi
2683 \fi

```

```

2684         \endcsname
2685     }%
2686 } \x
2687 \HyField@SetKeys{#1}%
2688 \PDFForm@Name
2689 \ifFld@hidden\def\Fld@width{1sp}\fi
2690 \ifx\Fld@value\relax
2691     \let\Fld@value\Fld@default
2692 \fi
2693 \LayoutChoiceField{#2}{%
2694     \ifFld@radio
2695         \HyField@FlagsRadioButton
2696         \__hypRadio{#3}%
2697     \else
2698         \beginngroup
2699             \HyField@FlagsChoice
2700             \ifdim\Fld@width<\@tempdima
2701                 \ifdim\@tempdima<1cm\@tempdima1cm\fi
2702                 \edef\Fld@width{\the\@tempdima}%
2703             \fi
2704             \ifFld@combo
2705             \else
2706                 \@tempdima=\the\Fld@menulength\Fld@charsize
2707                 \advance\@tempdima by~\Fld@borderwidth bp %
2708                 \advance\@tempdima by~\Fld@borderwidth bp %
2709                 \edef\Fld@height{\the\@tempdima}%
2710             \fi
2711             \__hypListbox{#3}%
2712         \endgroup
2713     \fi
2714 }%
2715 \endgroup
2716 }
2717
2718 \def\__hypRadio#1{%
2719     \Fld@listcount=0~%
2720     %\show\Fld@default
2721     \EdefEscapeName\Fld@default{\Fld@default}%
2722     \clist_map_variable:nNn { #1 } \@curropt
2723     %\@for\@curropt:=#1\do
2724     {%
2725         \expandafter\Fld@checkequals\@curropt==\%
2726         \EdefEscapeName\@currValue{\@currValue}%
2727         \Hy@StepCount\Fld@listcount
2728         \@currDisplay\space
2729         \leavevmode
2730         \HyAnn@AbsPageLabel
2731         \Hy@escapeform\PDFForm@Radio
2732         \pdfxform_if_exist:nF { __hyp_xform_Ding }
2733         {
2734             \pdfxform_new:nnn { __hyp_xform_Ding } {}
2735             {
2736                 \group_begin:
2737                 \fontfamily{pzd}

```

```

2738         \fontencoding{U}
2739         \fontseries{m}
2740         \fontshape{n}
2741         \selectfont
2742         \char123
2743         \group_end:
2744     }
2745 }
2746 \pdfannot_box:nnnx
2747 {\Fld@width}
2748 {\Fld@height}
2749 {Opt} %is this correct?
2750 {
2751     \PDFForm@Radio
2752     /AP
2753     <<
2754     /N
2755     <<
2756     /@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2757     %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2758     >>
2759     >>
2760 }
2761 {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2762 \int_compare:nNtT { \Fld@listcount } = { 1 }
2763 { \HyField@AddToFields }
2764 \c_space_tl % deliberate space between radio buttons
2765 % to do: --> should be configurable
2766 }%
2767 }
2768
2769 \newcount\Fld@listcount
2770 \def\__hypListbox#1
2771 {
2772     \HyField@PDFChoices{#1}
2773     \mode_leave_vertical:
2774     \HyAnn@AbsPageLabel
2775     \Hy@escapeform\PDFForm@List
2776     \pdf_link_user:nnn
2777         {widget} %perhaps we need more types??
2778         {\PDFForm@List}
2779         {\MakeChoiceField{\Fld@width}{\Fld@height}}
2780     \HyField@AddToFields
2781 }
2782
2783
2784 \def\@PushButton[#1]#2{% parameters, label
2785     \def\Fld@name{#2}%
2786     \group_begin:
2787         \exp_args:No\HyField@SetKeys
2788         {
2789             \DefaultOptionsofPushButton,#1
2790         }
2791     \PDFForm@Name

```

```

2792 \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2793 {
2794   \HyField@FlagsPushButton
2795   \legacy_if:nT {Fld@hidden}
2796   {
2797     \def\Fld@width{1sp}
2798   }
2799   \LayoutPushButtonField
2800   {
2801     \mode_leave_vertical:
2802     \HyAnn@AbsPageLabel
2803     \Hy@escapeform\PDFForm@Push
2804     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2805     \pdfannot_box:nnnn
2806     {\box_wd:N\l_tmpa_box}
2807     {\box_ht:N\l_tmpa_box}
2808     {\box_dp:N\l_tmpa_box} %is this correct?
2809     {\PDFForm@Push}
2810     {\box_use:N\l_tmpa_box}
2811     \HyField@AddToFields
2812   }
2813 }
2814 {
2815   \msg_error:nn { hyp }{ pdfa-no-push-button }
2816   \LayoutPushButtonField
2817   {
2818     \mode_leave_vertical:
2819     \MakeButtonField{#2}
2820   }
2821 }
2822 \group_end:
2823 }
2824
2825 \def\@Submit[#1]#2
2826 {
2827   \def\Fld@width {\DefaultWidthofSubmit}
2828   \def\Fld@height{\DefaultHeightofSubmit}
2829   \group_begin:
2830     \exp_args:No\HyField@SetKeys
2831     {
2832       \DefaultOptionsofSubmit,#1
2833     }
2834   \HyField@FlagsPushButton
2835   \HyField@FlagsSubmit
2836   \legacy_if:nT { Fld@hidden }
2837   {
2838     \def\Fld@width{1sp}
2839   }
2840   \mode_leave_vertical:
2841   \HyAnn@AbsPageLabel
2842   \Hy@escapeform\PDFForm@Submit
2843   \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2844   \pdfxform_if_exist:nF
2845   { __hyp_xform_Submit }

```

```

2846 {
2847   \pdfxform_new:nnn { __hyp_xform_Submit }{}
2848   {
2849     \fbox{\color_select:n{yellow}\textsf{Submit}}
2850   }
2851   \pdfxform_new:nnn { __hyp_xform_SubmitP }{}
2852   {
2853     \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2854   }
2855 }
2856 \pdfannot_box:nnnn
2857 {\box_wd:N\l_tmpa_box}
2858 {\box_ht:N\l_tmpa_box}
2859 {\box_dp:N\l_tmpa_box} %is this correct?
2860 {
2861   \PDFForm@Submit
2862   /AP<<
2863     /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2864     /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2865   >>
2866 }
2867 \HyField@AddToFields
2868 \box_use:N\l_tmpa_box
2869
2870 \group_end:
2871 }
2872
2873 \def\@Reset[#1]#2
2874 {
2875   \def\Fld@width {\DefaultWidthofReset}
2876   \def\Fld@height{\DefaultHeightofReset}
2877   \group_begin:
2878     \exp_args:No\HyField@SetKeys
2879     {
2880       \DefaultOptionsofReset,#1
2881     }
2882   \mode_leave_vertical:
2883   \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2884   {
2885     \HyField@FlagsPushButton
2886     \legacy_if:nT { Fld@hidden }
2887     { \def\Fld@width{1sp} }
2888     \HyAnn@AbsPageLabel
2889     \Hy@escapeform\PDFForm@Reset
2890     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2891     \pdfannot_box:nnnn
2892     {\box_wd:N\l_tmpa_box}
2893     {\box_ht:N\l_tmpa_box}
2894     {\box_dp:N\l_tmpa_box} %is this correct?
2895     { \PDFForm@Reset }
2896     \HyField@AddToFields
2897     \box_use:N \l_tmpa_box
2898   }
2899   {

```

```

2900         \msg_error:nn { hyp }{ pdfa-no-reset-button }
2901         \MakeButtonField{#2}
2902     }
2903     \group_end:
2904 }
2905
2906 \def\@CheckBox[#1]#2
2907 {% parameters, label
2908     \def\Fld@name{#2}
2909     \def\Fld@default{0}
2910     \group_begin:
2911         \def\Fld@width {\DefaultWidthofCheckBox}
2912         \def\Fld@height{\DefaultHeightofCheckBox}
2913         \exp_args:No\HyField@SetKeys
2914         {
2915             \DefaultOptionsofCheckBox,#1
2916         }
2917         \PDFForm@Name
2918         \HyField@FlagsCheckBox
2919         \legacy_if:nT { Fld@hidden }
2920         {
2921             \def\Fld@width{1sp}
2922         }
2923     \LayoutCheckField{#2}
2924     {
2925         \mode_leave_vertical:
2926         \HyAnn@AbsPageLabel
2927         \Hy@escapeform\PDFForm@Check
2928         \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
2929         {
2930             \pdfxform_new:nnn
2931             {__hyp_xform_CheckMarkYes}{-}
2932             {
2933                 \group_begin:
2934                 \fontfamily{pzd}
2935                 \fontencoding{U}
2936                 \fontseries{m}
2937                 \fontshape{n}
2938                 \selectfont
2939                 \char51
2940                 \group_end:
2941             }
2942             \pdfxform_new:nnn
2943             {__hyp_xform_CheckMarkOff}{-}
2944             {
2945                 \group_begin:
2946                 \fontfamily{pzd}
2947                 \fontencoding{U}
2948                 \fontseries{m}
2949                 \fontshape{n}
2950                 \selectfont
2951                 \phantom{\char51} %perhaps xetex needs some small glyph ..
2952                 \group_end:
2953             }

```

```

2954         }
2955         \pdfannot_box:nnnn
2956         {\Fld@width}
2957         {\Fld@height}
2958         {Opt} %is this correct?
2959         {\PDFForm@Check}
2960         \HyField@AddToFields %check if this works with xelatex ...
2961     }
2962     \group_end:
2963 }
2964 \ExplSyntaxOff
2965
2966 %hm. Should a luatex driver use type1 fonts in fields????
2967 \ExplSyntaxOn
2968 \def\Hy@FormObjects
2969 {
2970     \pdf_object_new:nn    {__hyp/Encoding/pdfdoc } { dict }
2971     \pdf_object_new:nn    {__hyp/Font/ZaDb } { dict }
2972     \pdf_object_new:nn    {__hyp/Font/Helv } { dict }
2973     \pdf_object_write:nx {__hyp/Encoding/pdfdoc }
2974     {
2975         /Type/Encoding
2976         /Differences[
2977             24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2978             /ring/tilde
2979             \c_space_tl
2980             39/quotesingle
2981             \c_space_tl
2982             96/grave %
2983             \iow_newline:
2984             128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2985             /fraction/guilsinglleft/guilsinglright/minus/perthousand
2986             /quotedblbase/quotedblleft/quotedblright/quoteleft
2987             /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2988             /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2989             \iow_newline:
2990             164/currency
2991             \c_space_tl
2992             166/brokenbar
2993             \c_space_tl
2994             168/dieresis/copyright/ordfeminine
2995             \c_space_tl
2996             172/logicalnot/.notdef/registered/macron/degree/plusminus
2997             /twosuperior/threesuperior/acute/mu
2998             \c_space_tl
2999             183/periodcentered/cedilla/onesuperior/ordmasculine
3000             \c_space_tl
3001             188/onequarter/onehalf/threequarters
3002             \iow_newline:
3003             192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
3004             /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
3005             /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
3006             /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
3007             /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls

```



```

3008         /grave/aacute/acircumflex/atilde/adieresis/aring/ae
3009         /cedilla/egrave/eacute/ecircumflex/edieresis/igrave
3010         /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
3011         /ocircumflex/otilde/odieresis/divide/oslash/ugrave
3012         /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
3013     ]
3014 }
3015 \pdf_object_write:nn {__hyp/Font/ZaDb }
3016 {
3017     /Type/Font
3018     /Subtype/Type1
3019     /Name/ZaDb
3020     /BaseFont/ZapfDingbats
3021 }
3022 \pdf_object_write:nx {__hyp/Font/Helv }
3023 {
3024     /Type/Font
3025     /Subtype/Type1
3026     /Name/Helv
3027     /BaseFont/Helvetica
3028     /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
3029 }
3030 \global\let\Hy@FormObjects\relax
3031 }
3032 \ExplSyntaxOff
3033 \providecommand*{\Fld@pageobjref}{\}
3034 \ifcsname pdf@escapestring\endcsname
3035   \def\Hy@escapeform#1{%
3036     \ifHy@pdfescapeform
3037       \let\Hy@escapestring\pdfescapestring
3038     \else
3039       \let\Hy@escapestring\@firstofone
3040     \fi
3041   }%
3042   \Hy@escapeform{}\}
3043 \else
3044   \let\Hy@escapestring\@firstofone
3045   \def\Hy@escapeform#1{%
3046     \ifHy@pdfescapeform
3047       \def\Hy@escapestring##1{%
3048         \noexpand\Hy@escapestring{\noexpand##1}%
3049       }%
3050       \edef\Hy@temp{#1}%
3051       \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{\}\@nil
3052       \def\Hy@escapestring##1{%
3053         \@ifundefined{Hy@esc@\string##1}{%
3054           ##1%
3055           \ThisShouldNotHappen
3056         }{%
3057           \csname Hy@esc@\string##1\endcsname
3058         }%
3059       }%
3060     \else
3061       \let\Hy@escapestring\@firstofone

```

```

3062     \fi
3063 }%
3064 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3065     \ifx\#3\%
3066     \else
3067     \expandafter
3068     \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3069     \ltx@ReturnAfterFi{%
3070     \Hy__hypescapeform#3\@nil
3071     }%
3072     \fi
3073 }%
3074 \fi
3075 \def\PDFForm@Name{%
3076     \PDFForm__hypName\Fld@name
3077     \ifx\Fld@altname\relax
3078     \else
3079     \PDFForm__hypName\Fld@altname
3080     \fi
3081     \ifx\Fld@mappingname\relax
3082     \else
3083     \PDFForm__hypName\Fld@mappingname
3084     \fi
3085 }
3086 \def\PDFForm__hypName#1{%
3087     \begingroup
3088     \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3089     \ifHy@unicode
3090     \Hy@unicodedefalse
3091     \fi
3092     \fi
3093     \pdfstringdef\Hy@gtemp#1%
3094     \endgroup
3095     \let#1\Hy@gtemp
3096 }
3097 \def\Fld@X@additionalactions{%
3098     \ifx\Fld@keystroke@code\@empty
3099     \else
3100     /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3101     \fi
3102     \ifx\Fld@format@code\@empty
3103     \else
3104     /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3105     \fi
3106     \ifx\Fld@validate@code\@empty
3107     \else
3108     /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3109     \fi
3110     \ifx\Fld@calculate@code\@empty
3111     \else
3112     /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3113     \fi
3114     \ifx\Fld@onfocus@code\@empty
3115     \else

```

```

3116     /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3117 \fi
3118 \ifx\Fld@onblur@code\@empty
3119 \else
3120     /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3121 \fi
3122 \ifx\Fld@onmousedown@code\@empty
3123 \else
3124     /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3125 \fi
3126 \ifx\Fld@onmouseup@code\@empty
3127 \else
3128     /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3129 \fi
3130 \ifx\Fld@onenter@code\@empty
3131 \else
3132     /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3133 \fi
3134 \ifx\Fld@onexit@code\@empty
3135 \else
3136     /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3137 \fi
3138 }
3139 \ExplSyntaxOn
3140 \def\Fld@additionalactions
3141 {%
3142     \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{}
3143     {
3144         \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3145         {/AA<<\Fld@X@additionalactions>>}
3146     }
3147 }
3148 \ExplSyntaxOff
3149 \def\Fld@annotnames{%
3150     /T(\Fld@name)%
3151     \ifx\Fld@altname\relax
3152     \else
3153         /TU(\Fld@altname)%
3154     \fi
3155     \ifx\Fld@mappingname\relax
3156     \else
3157         /TM(\Fld@mappingname)%
3158     \fi
3159 }
3160 \ExplSyntaxOn
3161 \def\PDFForm@Check
3162 {
3163     /Subtype/Widget
3164     ~\Fld@annotflags
3165     ~\Fld@pageobjref
3166     ~\Fld@annotnames
3167     /FT/Btn
3168     \Fld@flags
3169     /Q~\Fld@align

```

```

3170 /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3171 /AP
3172 <<
3173 /N
3174 <<
3175 /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3176 /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3177 >>
3178 >>
3179 /MK<<
3180 \int_compare:nNf {\Fld@rotation}={0}
3181 {
3182 /R~\Fld@rotation
3183 }
3184 \tl_if_empty:NF\Fld@bordercolor
3185 {
3186 /BC[\Fld@bordercolor]
3187 }
3188 \tl_if_empty:NF\Fld@bcolor
3189 {
3190 /BG[\Fld@bcolor]
3191 }
3192 /CA(\Hy@escapestring{\Fld@cbsymbol})%
3193 >>
3194 /DA
3195 (
3196 /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3197 \tl_if_empty:NF \Fld@color
3198 {
3199 \c_space_tl \Fld@color
3200 }
3201 )
3202 /H/P
3203 \legacy_if:nTF {\Fld@checked}
3204 {
3205 /V/Yes /AS/Yes
3206 }
3207 {
3208 /V/Off /AS/Off
3209 }
3210 \Fld@additionalactions
3211 }
3212 \ExplSyntaxOff
3213 \ExplSyntaxOn
3214 \def\PDFForm@Push
3215 {
3216 /Subtype/Widget
3217 ~\Fld@annotflags
3218 ~\Fld@pageobjref
3219 ~\Fld@annotnames
3220 /FT/Btn
3221 ~\Fld@flags
3222 /H/P
3223 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>

```

```

3224     \bool_if:nT
3225     {
3226         !\int_compare_p:nNn {\Fld@rotation} = {0}
3227         ||
3228         \tl_if_exist_p:N \Fld@bordercolor
3229     }
3230     {
3231         /MK
3232         <<
3233         \int_compare:nNf {\Fld@rotation} = {0}
3234         {
3235             /R~\Fld@rotation
3236         }
3237         \tl_if_exist:NT \Fld@bordercolor
3238         {
3239             /BC[\Fld@bordercolor]
3240         }
3241         >>
3242     }
3243     /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3244     \Fld@additionalactions
3245 }
3246
3247 \ExplSyntaxOff
3248 \def\PDFForm@List{%
3249     /Subtype/Widget%
3250     \Fld@annotflags
3251     \Fld@pageobjref
3252     \Fld@annotnames
3253     /FT/Ch%
3254     \Fld@flags
3255     /Q \Fld@align
3256     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3257     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3258         \ifx\Fld@bordercolor\relax\else 1\fi
3259         \ifx\Fld@bcolor\relax \else 1\fi
3260         \space
3261     \else
3262         /MK<<%
3263         \ifnum\Fld@rotation=\z@
3264         \else
3265             /R \Fld@rotation
3266         \fi
3267         \ifx\Fld@bordercolor\relax
3268         \else
3269             /BC[\Fld@bordercolor]%
3270         \fi
3271         \ifx\Fld@bcolor\relax
3272         \else
3273             /BG[\Fld@bcolor]%
3274         \fi
3275         >>%
3276     \fi
3277     /DA(/Helv \strip@pt\Fld@charsize\space Tf%

```

```

3278         \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3279     \Fld@choices
3280     \Fld@additionalactions
3281 }
3282 \ExplSyntaxOn
3283 \def\PDFForm@Radio
3284 {
3285     /Subtype/Widget
3286     ~\Fld@annotflags
3287     ~\Fld@pageobjref
3288     ~\Fld@annotnames
3289     /FT/Btn
3290     \Fld@flags
3291     /H/P
3292     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3293     /MK<<
3294     \ifnum\Fld@rotation=\z@
3295     \else
3296     /R~\Fld@rotation
3297     \fi
3298     \ifx\Fld@bordercolor\relax
3299     \else
3300     /BC[\Fld@bordercolor]%
3301     \fi
3302     \ifx\Fld@bcolor\relax
3303     \else
3304     /BG[\Fld@bcolor]%
3305     \fi
3306     /CA(\Hy@escapestring{\Fld@radiosymbol})%
3307     >>
3308     /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3309     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3310     \ifx\Fld@default\@empty
3311     /V/Off%
3312     /DV/Off%
3313     \else
3314     /V/\Fld@default
3315     /DV/\Fld@default
3316     \fi
3317     \Fld@additionalactions
3318 }
3319 \ExplSyntaxOff
3320 \ExplSyntaxOn
3321 % Does an appeareance dict make sense here?
3322 \def\PDFForm@Text
3323 {
3324     /Subtype/Widget
3325     ~\Fld@annotflags
3326     ~\Fld@pageobjref
3327     ~\Fld@annotnames
3328     /FT/Tx
3329     ~\Fld@flags
3330     /Q~\Fld@align
3331     /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>

```

```

3332 \bool_if:nT
3333 {
3334     !\int_compare_p:nNn {\Fld@rotation} = {0}
3335     ||
3336     \tl_if_exist_p:N \Fld@bordercolor
3337     ||
3338     \tl_if_exist_p:N \Fld@bcolor
3339 }
3340 {
3341     /MK
3342     <<
3343     \int_compare:nNf {\Fld@rotation} = {0}
3344     {
3345         /R~\Fld@rotation
3346     }
3347     \tl_if_exist:NT \Fld@bordercolor
3348     {
3349         /BC[\Fld@bordercolor]
3350     }
3351     \tl_if_exist:NT \Fld@bcolor
3352     {
3353         /BG[\Fld@bcolor]
3354     }
3355     >>
3356 }
3357 /DA
3358 (
3359     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3360     \tl_if_empty:NF {\c_space_tl\Fld@color}
3361 )
3362 /DV(\Hy@escapestring{\Fld@default})
3363 /V(\Hy@escapestring{\Fld@value})
3364 ~\Fld@additionalactions
3365 \int_compare:nNt { \Fld@maxlen}>{0}
3366 {
3367     /MaxLen~\Fld@maxlen
3368 }
3369 }
3370 \ExplSyntaxOff
3371
3372 \def\PDFForm@Submit{%
3373     /Subtype/Widget%
3374     \Fld@annotflags
3375     \Fld@pageobjref
3376     \Fld@annotnames
3377     /FT/Btn%
3378     \Fld@flags
3379     /H/P%
3380     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3381     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3382         \ifx\Fld@bordercolor\relax\else 1\fi
3383         \space
3384     \else
3385         /MK<<%

```

```

3386     \ifnum\Fld@rotation=\z@
3387     \else
3388         /R \Fld@rotation
3389     \fi
3390     \ifx\Fld@bordercolor\relax
3391     \else
3392         /BC[\Fld@bordercolor]%
3393     \fi
3394     >>%
3395 \fi
3396 /A<<%
3397     /S/SubmitForm%
3398     /F<<%
3399         /FS/URL%
3400         /F(\Hy@escapestring{\Form@action})%
3401     >>%
3402     \Fld@submitflags
3403     >>%
3404     \Fld@additionalactions
3405 }
3406 \ExplSyntaxOn
3407     \def\PDFForm@Reset{%
3408         /Subtype/Widget%
3409         \Fld@annotflags
3410         \Fld@pageobjref
3411         \Fld@annotnames
3412         /FT/Btn%
3413         \Fld@flags
3414         /H/P%
3415         /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3416         \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3417             \ifx\Fld@bordercolor\relax\else 1\fi
3418             \space
3419         \else
3420             /MK<<%
3421                 \ifnum\Fld@rotation=\z@
3422                 \else
3423                     /R~\Fld@rotation
3424                 \fi
3425                 \ifx\Fld@bordercolor\relax
3426                 \else
3427                     /BC[\Fld@bordercolor]%
3428                 \fi
3429             >>%
3430         \fi
3431         /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3432         /A<</S/ResetForm>>%
3433         \Fld@additionalactions
3434     }%
3435
3436
3437 %these patterns are used in hyperref checks.
3438 %it is unclear if they are really useful and if a backend support is
3439 %needed.

```



```

3440 \str_case:VnF \c_sys_backend_str
3441 {
3442   { pdfmode }
3443   {
3444     \def\HyPat@ObjRef
3445     {
3446       [0-9]*[1-9][0-9]*~0~R
3447     }
3448   }
3449   { dvipdfmx }
3450   {
3451     \def\HyPat@ObjRef
3452     {
3453       @[\~]+
3454     }
3455   }
3456   { xdvipdfmx }
3457   {
3458     \def\HyPat@ObjRef
3459     {
3460       @[\~]+
3461     }
3462   }
3463 }
3464 { %also set in hyperref sty, so probably not needed.
3465   \def\HyPat@ObjRef/{.+}
3466 }
3467
3468
3469 \ExplSyntaxOff
3470 % UF: removed Hy@writebookmark
3471 %   \Hy@currentbookmarklevel{0}
3472 %   \Hy@numberline
3473 %   \__hypwritetorep
3474 %   counter{bookmark@seq@number}
3475 % removed \HyPsd@SanitizeForOutFile, not needed
3476 % removed \currentpdfbookmark, defined by bookmark,
3477 % should use \newcommand there
3478 % removed \subpdfbookmark, defined by bookmark,
3479 % should use \newcommand there
3480 % removed \belowpdfbookmark, defined by bookmark,
3481 % should use \newcommand there
3482 % removed \pdfbookmark, defined by bookmark,
3483 % \BOOKMARK
3484 % \@BOOKMARK
3485 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3486 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3487 %% removed \ReadBookmarks / unneeded with bookmark.
3488 %% removed \Hy@OutlineName
3489 %% removed \check@bm@number
3490 %% removed \calc@bm@number
3491
3492 \ifHy@implicit
3493 \else

```

```

3494 \expandafter\endinput
3495 \fi
3496 \newlength\Hy@SectionHShift
3497 \def\Hy@SectionAnchorHref#1{%
3498 \ifx\protect\@typeset@protect
3499 \Hy__hypSectionAnchor{#1}%
3500 \fi
3501 }
3502 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%
3503 \leavevmode
3504 \hbox to Opt{%
3505 \kern-\Hy@SectionHShift
3506 \Hy@raisedlink{%
3507 \hyper@anchorstart{#1}\hyper@anchorend
3508 }%
3509 \hss
3510 }%
3511 }
3512 \let\H@old@ssect\@ssect
3513 \def\@ssect#1#2#3#4#5{%
3514 \Hy@MakeCurrentHrefAuto{section*}%
3515 \setlength{\Hy@SectionHShift}{#1}%
3516 \begingroup
3517 \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3518 \toks\tw\expandafter{%
3519 \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3520 #5%
3521 }%
3522 \edef\x{\endgroup
3523 \the\toks@\the\toks\tw}%
3524 }\x
3525 }
3526 \let\H@old@schapter\@schapter
3527 \def\@schapter#1{%
3528 \begingroup
3529 \let\@mkboth\@gobbletwo
3530 \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3531 \Hy@raisedlink{%
3532 \hyper@anchorstart{\@currentHref}\hyper@anchorend
3533 }%
3534 \endgroup
3535 \H@old@schapter{#1}%
3536 }
3537 \ltx@ifundefined{@chapter}{-}{%
3538 \let\Hy@org@chapter\@chapter
3539 \def\@chapter{%
3540 \def\Hy@next{%
3541 \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3542 \Hy@raisedlink{%
3543 \hyper@anchorstart{\@currentHref}\hyper@anchorend
3544 }%
3545 }%
3546 \ifnum\c@secnumdepth>\m@ne
3547 \ltx@ifundefined{if@mainmatter}%

```

```

3548     \iftrue{\csname if@mainmatter\endcsname}%
3549     \let\Hy@next\relax
3550     \fi
3551 \fi
3552 \Hy@next
3553 \Hy@org@chapter
3554 }%
3555 }
3556 \let\H@old@part\@part
3557 \begingroup\expandafter\expandafter\expandafter\endgroup
3558 \expandafter\ifx\csname chapter\endcsname\relax
3559     \let\Hy@secnum@part\z@
3560 \else
3561     \let\Hy@secnum@part\m@ne
3562 \fi
3563 \def\@part{%
3564     \ifnum\Hy@secnum@part>\c@secnumdepth
3565         \phantomsection
3566     \fi
3567     \H@old@part
3568 }
3569 \let\H@old@spart\@spart
3570 \def\@spart#1{%
3571     \Hy@MakeCurrentHrefAuto{part*}%
3572     \Hy@raisedlink{%
3573         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3574     }%
3575     \H@old@spart{#1}%
3576 }
3577 \let\H@old@sect\@sect
3578 \def\@sect#1#2#3#4#5#6[#7]#8{%
3579     \ifnum #2>\c@secnumdepth
3580         \expandafter\@firstoftwo
3581     \else
3582         \expandafter\@secondoftwo
3583     \fi
3584     {%
3585         \Hy@MakeCurrentHrefAuto{section*}%
3586         \setlength{\Hy@SectionHShift}{#3}%
3587         \begingroup
3588         \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3589         \toks\tw@\expandafter{%
3590             \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3591             #8%
3592         }%
3593         \edef\x{\endgroup
3594             \the\toks@{\the\toks\tw@}%
3595         }\x
3596     }{%
3597         \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3598     }%
3599 }
3600 \expandafter\def\csname Parent-4\endcsname{}
3601 \expandafter\def\csname Parent-3\endcsname{}

```

```

3602 \expandafter\def\csname Parent-2\endcsname{}
3603 \expandafter\def\csname Parent-1\endcsname{}
3604 \expandafter\def\csname Parent0\endcsname{}
3605 \expandafter\def\csname Parent1\endcsname{}
3606 \expandafter\def\csname Parent2\endcsname{}
3607 \expandafter\def\csname Parent3\endcsname{}
3608 \expandafter\def\csname Parent4\endcsname{}
3609 %%
3610 %% End of file 'hgeneric-testphase.def'.
3611 \package

3612 \*colorscheme)
3613 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3614 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3615 % linkcolor=
3616 %,filecolor=
3617 %,urlcolor=
3618 %,menucolor=
3619 %,runcolor=
3620 %,linkbordercolor=
3621 %,filebordercolor=
3622 %,urlbordercolor=
3623 %,menubordercolor=
3624 %,runbordercolor=
3625
3626 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }
3627 {
3628     linkcolor      = [rgb]{1,0,0}, %red
3629     filecolor      = [rgb]{0,1,1}, %cyan
3630     urlcolor       = [rgb]{1,0,1}, %magenta
3631     menucolor      = [rgb]{1, 0, 0}, %red
3632     runcolor       = [rgb]{0,1,1}, %cyan
3633     %-----
3634     linkbordercolor = [rgb]{1, 0 ,0 },
3635     filebordercolor = [rgb]{0, .5, .5},
3636     urlbordercolor  = [rgb]{0, 1, 1},
3637     menubordercolor = [rgb]{1, 0, 0},
3638     runbordercolor  = [rgb]{0, .7, .7}
3639 }
3640
3641 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3642 {
3643     linkcolor      = [rgb]{0,0.2,0.6},
3644     filecolor      = [rgb]{0.8,0,0.8},
3645     urlcolor       = [rgb]{0.8,0,0.8},
3646     menucolor      = [rgb]{0,0.2,0.6},
3647     runcolor       = [rgb]{0.8,0,0.8},
3648     %----- %-----
3649     linkbordercolor = [rgb]{0,0.2,0.6},
3650     filebordercolor = [rgb]{0.8,0,0.8},
3651     urlbordercolor  = [rgb]{0.8,0,0.8},
3652     menubordercolor = [rgb]{0,0.2,0.6},
3653     runbordercolor  = [rgb]{0.8,0,0.8}
3654 }
3655

```

```

3656 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3657 { %two colors: intern/extern
3658   linkcolor      = [rgb]{0.79216, 0, 0.12549},
3659   filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3660   urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3661   menucolor      = [rgb]{0.79216, 0, 0.12549 },
3662   runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3663   %----- %-----
3664   linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3665   filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3666   urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3667   menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3668   runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3669 }
3670
3671 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3672 { %all darkgray
3673   linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3674   filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3675   urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3676   menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3677   runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3678   %----- %-----
3679   linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3680   filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3681   urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3682   menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3683   runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3684 }
3685
3686 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3687 { %dvipsnam.def
3688   linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3689   filecolor      = [rgb]{1, 0, 0}, %Red
3690   urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3691   menucolor      = [rgb]{1, 0, 0}, %Red
3692   runcolor       = [rgb]{1, 0, 0}, %Red
3693   %----- %-----
3694   linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3695   filebordercolor = [rgb]{1, 0, 0}, %Red
3696   urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3697   menubordercolor = [rgb]{1, 0, 0}, %Red
3698   runbordercolor  = [rgb]{1, 0, 0} %Red
3699 }
3700
3701 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3702 { %dvipsnam.def
3703   linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3704   filecolor      = [rgb]{0, 1, 0}, %Green
3705   urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3706   menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3707   runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3708   %----- %-----
3709   linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed

```

```

3710     filebordercolor = [rgb]{0, 1, 0},      %Green
3711     urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3712     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3713     runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3714 }
3715
3716
3717 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3718 {
3719     linkcolor      = [rgb]{0.50196, 0, 0.02353},
3720     filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3721     urlcolor       = [rgb]{0.54118, 0, 0.52941},
3722     menucolor      = [rgb]{0.44706, 0.45882, 0},
3723     runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3724     %----- %-----
3725     linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3726     filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3727     urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3728     menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3729     runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3730 }
3731
3732 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3733 {
3734     linkcolor      = [rgb]{0,0,0},
3735     filecolor      = [rgb]{0,0,0},
3736     urlcolor       = [rgb]{0,0,0},
3737     menucolor      = [rgb]{0,0,0},
3738     runcolor       = [rgb]{0,0,0},
3739     %----- %-----
3740     linkbordercolor = [rgb]{0,0,0},
3741     filebordercolor = [rgb]{0,0,0},
3742     urlbordercolor  = [rgb]{0,0,0},
3743     menubordercolor = [rgb]{0,0,0},
3744     runbordercolor  = [rgb]{0,0,0}
3745 }
3746 </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	
\#	278, 732
\\$	277
\%	733
\.	542, 546, 548
@curropt commands:	
\@curropt:	2656, 2723
\\	19, 20, 30, 31, 32, 40, 44, 54, 74, 81, 88, 95, 102, 117, 122, 123, 131, 132, 141, 142, 143, 144, 145, 152, 159, 279, 280, 862, 2658, 2725, 3065
_	540, 542, 546, 548
A	
\A	540
\Acrobatmenu	18, 172
\addcontentsline	13
\AddToDocumentProperties	387
\AddToHook	425, 438
\AddToHookNext	197
\advance	2589, 2663, 2707, 2708
allcolors (hypersetup key)	975
\author	2
B	
\b	544
\begingroup	208, 276, 374, 2495, 2623, 2652, 2664, 2698, 3087, 3516, 3528, 3557, 3587
\belowpdfbookmark	3480
\bgroup	273, 374
\BOOKMARK	3483
bookmarkstype (hypersetup key)	13
bool commands:	
\bool_if:NTF	266, 295, 354, 679, 695, 707, 720, 728, 758, 816, 873, 945, 955, 1199, 1210
\bool_if:nTF	828, 1906, 1931, 3224, 3332
\bool_lazy_and:nnTF	422
\bool_lazy_or:nnTF	1229, 1278
\bool_new:N	202, 203, 526, 530, 534
\bool_set_true:N	535
bordercolormodel (hypersetup key)	13, 996
box commands:	
\box_dp:N	616, 2808, 2859, 2894
\box_ht:N	615, 2807, 2858, 2893
\box_new:N	475, 537
\box_set_dp:Nn	2103
\box_set_ht:Nn	2110
\box_use:N	1216, 2810, 2868, 2897
\box_use_drop:N	1221
\box_wd:N	614, 2806, 2857, 2892
\l_tmpa_box	2804, 2806, 2807, 2808, 2810, 2843, 2857, 2858, 2859, 2868, 2890, 2892, 2893, 2894, 2897
C	
\catcode	277, 278
\char	2742, 2939, 2951
\chardef	157
\cite	33
clist commands:	
\clist_map_function:nN	124, 133
\clist_map_inline:nn	2282
\clist_map_variable:nNn	2655, 2722
color commands:	
\color_export:nnN	38, 430, 904, 1017
\color_select:n	912, 948, 1220, 2849, 2853
\color_select:nn	918
\color_set:nn	2, 392, 929
\color_set:nnn	2, 391, 935
color names:	
hyp/annot/file	523
hyp/annot/link	523
hyp/annot/menu	523
hyp/annot/run	523
hyp/annot/url	523
colorfile (hypersetup key)	975
colorlink (hypersetup key)	975
colorlinks (hypersetup key)	961
colormenu (hypersetup key)	975
colorrn (hypersetup key)	975
colorscheme (hypersetup key)	1, 1387
colorurl (hypersetup key)	975
cs commands:	
\cs_generate_variant:Nn	155, 156, 389, 454, 469, 474, 578, 904, 921, 938
\cs_gset:Npn	1191
\cs_gset_eq:NN	408, 419
\cs_if_exist:NTF	12
\cs_if_exist_p:N	423
\cs_new:Npn	117, 470
\cs_new_protected:Npn	208, 304, 321, 338, 364, 384, 399, 410, 448, 457, 554, 558, 562, 566, 579,

581, 585, 629, 637, 646, 650, 672, 677, 693, 705, 718, 756, 814, 871, 905, 916, 922, 933, 1135, 2138, 2209	\DocumentMetadata
\cs_set:Nn 214, 256	. . . 3, 10, 14, 15, 18, 12, 20, 32, 41, 47
\cs_set_eq:NN	draft (hypersetup key) 1411
. 160, 165, 166, 732, 733, 1464	
\cs_set_protected:Npn	E
. 158, 179, 595, 862	\edef 2497, 2666, 2702, 2709, 3050, 3522, 3593
\csname 2670, 3057, 3068, 3548, 3558, 3600, 3601, 3602, 3603, 3604, 3605, 3606, 3607, 3608	\EdefEscapeName 2721, 2726
\currentpdfbookmark 3476	\egroup 377
	\else 2494, 2619, 2673, 2677, 2680, 2697, 2705, 3038, 3043, 3060, 3066, 3078, 3082, 3099, 3103, 3107, 3111, 3115, 3119, 3123, 3127, 3131, 3135, 3152, 3156, 3257, 3258, 3259, 3261, 3264, 3268, 3272, 3278, 3295, 3299, 3303, 3309, 3313, 3381, 3382, 3384, 3387, 3391, 3416, 3417, 3419, 3422, 3426, 3493, 3560, 3581
D	\endcsname 2684, 3034, 3057, 3068, 3548, 3558, 3600, 3601, 3602, 3603, 3604, 3605, 3606, 3607, 3608
\d 542, 546, 548	\endgroup 282, 285, 375, 377, 2498, 2643, 2666, 2712, 2715, 3094, 3522, 3534, 3557, 3593
debug (hypersetup key) 1411	\endinput 3494
\DeclareRobustCommand	exp commands:
. . . 261, 289, 312, 329, 348, 377, 3502	\exp_args:Ne 3142
\def 164, 374, 375, 377, 590, 2458, 2462, 2469, 2487, 2516, 2529, 2587, 2596, 2611, 2612, 2615, 2616, 2629, 2630, 2646, 2647, 2650, 2651, 2689, 2718, 2770, 2784, 2785, 2797, 2825, 2827, 2828, 2838, 2873, 2875, 2876, 2887, 2906, 2908, 2909, 2911, 2912, 2921, 2968, 3035, 3045, 3047, 3052, 3064, 3075, 3086, 3097, 3140, 3149, 3161, 3214, 3248, 3283, 3322, 3372, 3407, 3444, 3451, 3458, 3465, 3497, 3513, 3527, 3539, 3540, 3563, 3570, 3578, 3600, 3601, 3602, 3603, 3604, 3605, 3606, 3607, 3608	\exp_args:Nne 2247
\DefaultHeightofCheckBox 2912	\exp_args:Nnno 323, 340
\DefaultHeightofChoiceMenu 2651	\exp_args:Nno 306, 366, 2195
\DefaultHeightofReset 2876	\exp_args:Nnox 766
\DefaultHeightofSubmit 2828	\exp_args:NNV 1891, 1922
\DefaultHeightofText 2620	\exp_args:NnV 631, 640
\DefaultHeightofTextMultiline 2618	\exp_args:Nnx 2335, 2440
\DefaultOptionsofCheckBox 2915	\exp_args:No
\DefaultOptionsofPushButton 2789 664, 668, 2787, 2830, 2878, 2913
\DefaultOptionsofReset 2880	\exp_args:NV 612, 619
\DefaultOptionsofSubmit 2832	\exp_args:Nx 762, 2188, 2236, 2489
\DefaultOptionsofText 2625	\exp_not:N 862
\DefaultWidthofCheckBox 2911	\exp_not:n
\DefaultWidthofChoiceMenu 2650	. . . 1326, 1360, 1471, 1531, 1552, 1589, 1609, 1632, 1676, 1709, 1734, 1785, 1810, 1837, 1886, 1963, 1991, 2018, 2038, 2268, 2385, 2413, 2423, 2434
\DefaultWidthofReset 2875	\expandafter 279, 2624, 2658, 2667, 2669, 2670, 2725, 3051, 3067, 3494, 3518, 3519, 3557, 3558, 3580, 3582, 3589, 3590, 3600, 3601, 3602, 3603, 3604, 3605, 3606, 3607, 3608
\DefaultWidthofSubmit 2827	\ExplSyntaxOff
\DefaultWidthofText 2615	. . . 9, 2526, 2579, 2606, 2964, 3032, 3148, 3212, 3247, 3319, 3370, 3469
destlabel (hypersetup key) 13	\ExplSyntaxOn
dim commands:	. . . 7, 10, 2527, 2584, 2607, 2967, 3139, 3160, 3213, 3282, 3320, 3406
\dim_to_decimal_in_bp:n 11, 160	
\do 2656, 2723	
\documentclass 21, 33	

extension (hypersetup key)	13, 1433	hidemenu (hypersetup key)	1363
F			
\fbbox	2761, 2849, 2853	hiderun (hypersetup key)	1363
\fi	2483,	hideurl (hypersetup key)	1363
	2510, 2513, 2522, 2621, 2629, 2630,	hook commands:	
	2661, 2679, 2682, 2683, 2689, 2692,	\hook_gput_code:nnn	
	2701, 2703, 2710, 2713, 3040, 3062,	401, 412, 941, 951, 1195, 1206
	3072, 3074, 3080, 3084, 3091, 3092,	\hook_new:n	390, 553, 628
	3101, 3105, 3109, 3113, 3117, 3121,	\hook_new_pair:nn	
	3125, 3129, 3133, 3137, 3154, 3158,	259, 287, 311, 328, 346
	3257, 3258, 3259, 3266, 3270, 3274,	\hook_use:n	
	3276, 3278, 3297, 3301, 3305, 3309,	263, 283, 292, 308, 315, 325, 332,
	3316, 3381, 3382, 3389, 3393, 3395,		342, 351, 370, 570, 634, 643, 684, 700
	3416, 3417, 3424, 3428, 3430, 3495,	\href	4, 5, 19, 261
	3500, 3550, 3551, 3562, 3566, 3583	\hreflaunch	5
file (hypersetup key)	10, 1479	\hrefpdf	5, 19, 312, 382
file commands:		\hrefrun	5, 329, 383
\file_input:n	8	\hrefurl	5, 19, 289, 381
fileborderstyle (hypersetup key)	14, 1089	\hss	3509
filecolor (hypersetup key)	975	Hy internal commands:	
final (hypersetup key)	1411	\Hy_hyescapeform	3051, 3064, 3070
\fontencoding	2738, 2935, 2947	\Hy_hypSectionAnchor	3499, 3502
\fontfamily	2737, 2934, 2946	HyField internal commands:	
\fontseries	2739, 2936, 2948	\HyField_hypAddToFields	2489, 2516
\fontshape	2740, 2937, 2949	hyp commands:	
fp commands:		\l_hyp_annot_colorfile_bool	524
\fp_eval:n	2061, 2089, 2105	\l_hyp_annot_colorlink_bool	524
G		\l_hyp_annot_colormenu_bool	524
\gdef	279, 280	\l_hyp_annot_colorrun_bool	524
\GetDocumentProperties	3	\l_hyp_annot_colorurl_bool	524
\global	2589, 3030	\l_hyp_annot_ocgcolorfile_bool	528
group commands:		\l_hyp_annot_ocgcolorlink_bool	528
\group_begin:	264, 293, 316, 333,	\l_hyp_annot_ocgcolormenu_bool	528
	352, 568, 682, 698, 722, 760, 818,	\l_hyp_annot_ocgcolorrun_bool	528
	875, 947, 1202, 1219, 2364, 2736,	\l_hyp_annot_ocgcolorurl_bool	528
	2786, 2829, 2877, 2910, 2933, 2945	\l_hyp_current_dest_name_tl	627, 683, 699
\group_end:	307, 324, 341,	hyp internal commands:	
	369, 575, 687, 710, 746, 806, 856,	\g__hyp_AcroForm_CoFields_prop	
	899, 957, 1222, 1225, 2371, 2743,	2450, 2471, 2540, 2542, 2557
	2822, 2870, 2903, 2940, 2952, 2962	\g__hyp_AcroForm_Fields_prop	
H		2451, 2464, 2535
\hbox	374, 3504	\l__hyp_annot_GoTo_bool	679, 695, 707
hbox commands:		\l__hyp_annot_GoTo_bool\l__-	
\hbox_overlap_right:n	1216	_hyp_annot_URI_bool\l__-	
\hbox_set:Nn	2804, 2843, 2890	hyp_annot_GoToR_bool\l__-	
\hbox_set:Nw	1203	hyp_annot_Named_bool\l__-	
\hbox_set_end:	1212	hyp_annot_Launch_bool	532
\hbox_set_to_wd:Nnn	2087	\l__hyp_annot_GoToR_bool	758
hidefile (hypersetup key)	1363	\l__hyp_annot_Launch_bool	816
hidelink (hypersetup key)	1363	\l__hyp_annot_Named_bool	873
hidelinks (hypersetup key)	1363	\c__hyp_annot_types_seq	
		487, 524, 528, 975, 1276, 1374
		\l__hyp_annot_URI_bool	720

\g__hyp_bordercolormodel_str . . .	__hyp_href_url_aux:nn 302, 304
. 432, 523, 1000, 1019	\l__hyp_href_url_encode_bool . . .
__hyp_citebordercolor_hook_- 202, 213, 251, 266, 295, 354
init: 397, 410, 419	__hyp_href_url_format:
__hyp_citecolor_hook_init: 208, 214, 256, 367
. 395, 399, 408	\l__hyp_href_url_ismap_bool
__hyp_clist_display:n . 117, 124, 133 203, 227, 728
__hyp_color_select:n 39, 905, 905, 921	\l__hyp_href_url_protocol_tl . . .
__hyp_color_select_aux:wn 204, 215, 254, 306, 368
. 905, 909, 916	__hyp_link_goto_begin:nw
__hyp_color_set:nn 38, 39, 650, 685, 701
395, 397, 429, 922, 922, 938, 980, 1016	__hyp_link_goto_end: . 672, 686, 709
__hyp_color_set_aux:nwn 922, 926, 933	\c__hyp_map_annot_hyp_prop 487
\c__hyp_colorscheme_daleif_prop 3641	\c__hyp_map_hyp_annot_prop
\c__hyp_colorscheme_henryford_- 487, 939,
prop 3732	1003, 1043, 1070, 1078, 1089, 1116,
\c__hyp_colorscheme_julian_prop 3656	1124, 1193, 1301, 1337, 1347, 1479
\c__hyp_colorscheme_phelype_prop	__hyp_ocg_init: 1135, 1135, 1191, 1201
. 3717	l__hyp_page/Trans 551
\c__hyp_colorscheme_szabolcsA_-	__hyp_PageLabels_gpush: 585, 585, 598
prop 3686	\l__hyp_para_tmpa_tl 483, 827, 830, 839
\c__hyp_colorscheme_szabolcsB_-	\l__hyp_para_tmpa_tl\l__hyp_-
prop 3701	text_tmpa_str\g__hyp_text_-
\c__hyp_colorscheme_tivv_prop . 3671	tmpa_str 480
\l__hyp_dest_box 31,	__hyp_ref_check:nn 447, 457, 469, 2594
537, 614, 615, 616, 2087, 2103, 2110	__hyp_ref_if_exist:nn . 447, 455, 456
\l__hyp_dest_name_tmpa_tl . . 480,	__hyp_ref_if_exist:nnTF . . 459, 2598
653, 654, 660, 664, 666, 669, 779, 792	__hyp_ref_label:nn 447, 448, 454, 2593
\l__hyp_dest_pdfremotestartview_-	__hyp_ref_value:nn 447, 470, 474, 2602
tl 506, 787, 1894, 1898	__hyp_secondoftwoewithopt:wnn . . .
\g__hyp_dest_pdfstartpage_tl 380, 381, 382, 383
. . . 506, 1905, 1907, 1914, 1932, 1939	__hyp_setup_info_date_key:nn . . .
\g__hyp_dest_pdfstartview_tl 2209, 2241, 2242
506, 1907, 1914, 1925, 1929, 1932, 1939	__hyp_setup_info_key:nn . . 2138,
\l__hyp_dest_pdfview_tl	2202, 2203, 2204, 2205, 2207, 2208
. 522, 632, 641,	__hyp_store_metadata:nn
2056, 2059, 2066, 2069, 2070, 2071, 384, 389, 2135,
2072, 2073, 2074, 2079, 2083, 2119	2170, 2188, 2222, 2236, 2257, 2329
\c__hyp_dest_startview_regex . . .	__hyp_text_cleanup:N . 558, 558, 572
. 538, 1892, 1923	\l__hyp_text_enc_dest_print_tl . .
\c__hyp_dest_undefined_tl 509, 778
. 486, 659, 660	\l__hyp_text_enc_dest_tl
__hyp_destination:nn 31, 509, 608, 665
. 31, 601, 601, 632, 641	\l__hyp_text_enc_file_print_tl . .
\l__hyp_filename_tmpa_tl . . . 480, 509, 821
761, 763, 769, 770, 775, 822, 823, 839	\l__hyp_text_enc_info_print_tl . .
__hyp_href_pdf_aux:nn 318, 321 509, 583
\l__hyp_href_pdf_destination_tl . .	\l__hyp_text_enc_para_print_tl . .
. 205, 216, 255, 323 509, 826
\l__hyp_href_pdf_page_tl 206, 224, 786	\l__hyp_text_enc_uri_print_tl . . .
__hyp_href_run_aux:nn 335, 338 268,
\l__hyp_href_run_parameter_tl . . .	271, 297, 300, 356, 359, 509, 725, 1490
. 207, 228, 340	
__hyp_href_url_aux:n 361, 364	

_hyp_text_pdfstring:nnN	566, 566, 578, 580, 583, 606, 663, 723, 776, 819, 824, 1490
_hyp_text_pdfstring_info:nN	581, 581, 2164, 2182, 2192
_hyp_text_purify:nN	554, 554, 571
_hyp_text_string_from_unicode:nN	562, 562, 573
\g_hyp_text_tmpa_str	485, 574, 576
\l_hyp_text_tmpa_str	484, 571, 572, 573, 574
\l_hyp_tmpa_box	475, 1203, 1216, 1221
\l_hyp_tmpa_int	475
\l_hyp_tmpa_seq	475, 1892, 1894, 1923, 1925, 2046, 2047, 2051, 2053, 2077, 2084, 2085, 2086, 2095, 2097, 2107, 2112, 2544, 2546, 2552
\l_hyp_tmpa_str	475, 2164, 2165, 2167, 2182, 2183, 2185, 2192, 2193, 2197
\l_hyp_tmpa_tl	475, 609, 613, 621, 1020, 1024, 1490, 1491, 1496, 1890, 1892, 1921, 1923, 2053, 2054, 2061
\l_hyp_uri_tmpa_tl	480, 726, 727
hyp/anchor	627
hyp/annot/file (color name)	523
hyp/annot/link (color name)	523
hyp/annot/menu (color name)	523
hyp/annot/run (color name)	523
hyp/annot/url (color name)	523
hyp/text/pdfstring	553
\hypercalcbp	11, 18, 160
\HyperDestNameFilter	13, 607, 664
\hypersetup	2, 3, 5, 9, 10, 13, 18, 66, 99, 179
\hypersetup keys:	
allcolors	975
bookmarkstyle	13
bordercolormodel	13, 996
colorfile	975
colorlink	975
colorlinks	961
colormenu	975
colorrund	975
colorscheme	1, 1387
colorurl	975
debug	1411
destlabel	13
draft	1411
extension	13, 1433
file	10, 1479
fileborderstyle	14, 1089
filecolor	975
final	1411
hidefile	1363
hidelink	1363
hidelinks	1363
hidemenu	1363
hiderun	1363
hideurl	1363
hypertextnames	1433
link	10, 1479
linkborder	13
linkborderstyle	14, 1089
linkcolor	975
linkfileprefix	1433
linktoc	1433
linktocpage	1433
localanchorname	1433
menu	10, 1479
menuborder	13
menuborderstyle	14, 1089
menucolor	975
naturalnames	1433
nesting	13
ocgcolorfile	1229
ocgcolorlink	1229
ocgcolorlinks	1229
ocgcolormenu	1229
ocgcolorrun	1229
ocgcolorurl	1229
pageanchor	1433
pdfauthor	2138
pdfborder	13
pdfborderstyle	14, 1089
pdfcreationdate	14, 2209
pdfcreator	2138
pdfencoding	1401
pdfinfo	2271
pdfkeywords	2138
pdflang	14, 2124
pdflinkmargin	14
pdfmoddate	14, 2209
pdfproducer	2138
pdfremotestartview	11
pdfstartview	11
pdfsubject	2138
pdftitle	2138
pdftrapped	2243
pdfversion	1401
pdfview	11, 2042
plainpages	1433
run	10, 1479
runborder	13
runborderstyle	14, 1089
runcolor	975
unicode	1401
url	10, 1479
urlborder	13

urlborderstyle	14, 1089		
urlcolor	975		
verbose	1411		
hypertextnames (hypersetup key)	1433		
hypListbox internal commands:			
_hypListbox	2711, 2770		
hypRadio internal commands:			
_hypRadio	2696, 2718		
hypwritetorep internal commands:			
_hypwritetorep	3473		
I			
\ifcase	3257, 3381, 3416		
\ifcsname	3034		
\ifdim	2661, 2700, 2701		
\ifnum	3088, 3257, 3263, 3294, 3381, 3386, 3416, 3421, 3546, 3564, 3579		
\iftrue	3548		
\ifx	2493, 2630, 2690, 3065, 3077, 3081, 3098, 3102, 3106, 3110, 3114, 3118, 3122, 3126, 3130, 3134, 3151, 3155, 3258, 3259, 3267, 3271, 3278, 3298, 3302, 3309, 3310, 3382, 3390, 3417, 3425, 3498, 3558		
\immediate	2477, 2480		
int commands:			
\int_compare:nNnTF	2051, 2077, 2762, 3180, 3233, 3343, 3365		
\int_compare_p:nNn	3226, 3334		
\int_eval:n	785		
\int_max:nn	786		
\int_new:N	478		
iow commands:			
\iow_newline:	2983, 2989, 3002		
K			
\kern	3505		
keys commands:			
\keys_define:nn	185, 193, 211, 249, 393, 961, 977, 984, 996, 1005, 1030, 1045, 1064, 1091, 1110, 1233, 1247, 1259, 1282, 1288, 1303, 1332, 1363, 1376, 1387, 1401, 1411, 1420, 1433, 1460, 1481, 1486, 2042, 2124, 2140, 2173, 2211, 2225, 2243, 2271, 2327, 2342, 2355, 2375		
\l_keys_key_str	190, 2197, 2440		
\keys_set:nn	182, 219, 238, 265, 294, 317, 334, 353, 403, 414, 1393, 1400, 2248, 2275, 2278, 2279, 2280, 2281, 2336, 2365, 2440		
\keys_set_known:nn	2443		
\kvsetkeys	181, 2531		
L			
\label	9, 13		
\LayoutCheckField	2923		
\LayoutChoiceField	2693		
\LayoutPushButtonField	2799, 2816		
\LayoutTextField	2631		
\leavevmode	2632, 2729, 3503		
legacy commands:			
\legacy_if:nTF	167, 436, 443, 593, 2569, 2795, 2836, 2886, 2919, 3203		
\let	381, 382, 383, 690, 741, 748, 802, 808, 852, 858, 901, 2453, 2454, 2456, 2457, 2484, 2580, 2581, 2582, 2613, 2614, 2648, 2649, 2691, 3030, 3037, 3039, 3044, 3061, 3095, 3512, 3526, 3529, 3538, 3549, 3556, 3559, 3561, 3569, 3577		
link (hypersetup key)	10, 1479		
linkborder (hypersetup key)	13		
linkborderstyle (hypersetup key)	14, 1089		
linkcolor (hypersetup key)	975		
linkfileprefix (hypersetup key)	1433		
linktoc (hypersetup key)	1433		
linktocpage (hypersetup key)	1433		
localanchorname (hypersetup key)	1433		
M			
\MakeButtonField	2804, 2819, 2843, 2890, 2901		
\MakeChoiceField	2779		
\MakeFieldObject	2444		
\MakeRadioField	2761		
\MakeTextField	2640		
\mbox	1213		
menu (hypersetup key)	10, 1479		
menuborder (hypersetup key)	13		
menuborderstyle (hypersetup key)	14, 1089		
menucolor (hypersetup key)	975		
mode commands:			
\mode_if_horizontal:TF	603, 625		
\mode_leave_vertical:	262, 291, 314, 331, 350, 652, 735, 794, 841, 878, 2773, 2801, 2818, 2840, 2882, 2925		
msg commands:			
\msg_error:nn	424, 2815, 2900		
\msg_info:nnn	2150, 2155		
\msg_line_context:	82		
\g_msg_module_name_prop	11		
\msg_new:nnn	50, 57, 62, 66, 70, 77, 84, 91, 98, 105, 111, 118, 127, 137, 148		
\msg_new:nnnn	14, 25, 36		
\msg_warning:nn	169, 1407		
\msg_warning:nnn	178, 656, 896		

<code>\msg_warning:nnnn</code>	<code>\pdf_version:</code>
..... 1251, 1292, 1529, 1573,	... 1577, 1696, 1725, 1759, 1774,
1607, 1630, 1692, 1721, 1755, 1770,	1800, 1825, 1858, 1874, 1979, 2006
1783, 1796, 1821, 1854, 1870, 1897,	<code>\pdf_version_compare:NnTF</code>
1928, 1961, 1975, 2002, 2036, 2118	1567, 1658, 1680, 1716, 1749, 1764,
<code>\msg_warning:nnnnn</code>	1790, 1815, 1841, 1864, 1969, 1996
189,	<code>\pdf_version_compare_p:Nn</code>
1323, 1357, 1468, 1549, 1586, 1673, 832, 1230, 1279
1706, 1731, 1807, 1834, 1883, 1988,	<code>\pdf_version_major:</code>
2015, 2265, 2382, 2401, 2420, 2431 166, 1231, 1254, 1280, 1296
	<code>\pdf_version_minor:</code> .. 165, 1254, 1296
	pdfannot commands:
	<code>\pdfannot_box:nnnn</code>
	.. 2635, 2746, 2805, 2856, 2891, 2955
	<code>\pdfannot_box_ref_last:</code> .. 2491, 2507
	<code>\pdfannot_dict_put:nnn</code>
 736, 796, 842, 881, 1021,
	1056, 1080, 1102, 1126, 1308, 1339
	<code>\pdfannot_dict_remove:nn</code> .. 1011,
	1051, 1072, 1097, 1118, 1316, 1349
	<code>\pdfannot_link:nnn</code> . 737, 797, 843, 882
	<code>\pdfannot_link_goto_begin:nw</code> .. 669
	<code>\pdfannot_link_goto_end:</code>
	674
	<code>\pdfannot_link_margin:n</code>
	8, 1611
	<code>\c_pdfannot_link_types_seq</code> 532
	pdfauthor (hypersetup key)
	2138
	<code>\pdfbookmark</code>
	3482
	pdfborder (hypersetup key)
	13
	<code>\pdfborderstyle</code> (hypersetup key) . 14, 1089
	pdfcreationdate (hypersetup key) 14, 2209
	pdfcreator (hypersetup key)
	2138
	<code>\pdfdest</code>
	11
	pdffdict commands:
	<code>\pdffdict_new:n</code> 551, 714, 751, 810, 867
	<code>\pdffdict_put:nnn</code>
 236, 243, 552, 715, 716, 727,
	730, 752, 753, 765, 772, 782, 792,
	811, 812, 823, 838, 868, 869, 879,
	1637, 1638, 1642, 1643, 2379, 2389,
	2393, 2396, 2398, 2417, 2426, 2428
	<code>\pdffdict_remove:nn</code> 233, 835, 1647, 1648
	<code>\pdffdict_use:n</code>
	... 736, 795, 842, 848, 881, 886, 2368
	pdfencoding (hypersetup key)
	1401
	<code>\pdfescapestring</code>
	3037
	pdffile commands:
	<code>\pdffile_embed_file:nnn</code>
	767
	PDFForm internal commands:
	<code>\PDFForm_hypName</code>
 3076, 3079, 3083, 3086
	pdfinfo (hypersetup key)
	2271
	pdfkeywords (hypersetup key)
	2138
	pdflang (hypersetup key)
	14, 2124
	pdflinkmargin (hypersetup key)
	14

<code>\msg_warning:nnnn</code>	
..... 1251, 1292, 1529, 1573,	
1607, 1630, 1692, 1721, 1755, 1770,	
1783, 1796, 1821, 1854, 1870, 1897,	
1928, 1961, 1975, 2002, 2036, 2118	
<code>\msg_warning:nnnnn</code>	189,
1323, 1357, 1468, 1549, 1586, 1673,	
1706, 1731, 1807, 1834, 1883, 1988,	
2015, 2265, 2382, 2401, 2420, 2431	
	N
naturalnames (hypersetup key)	1433
nesting (hypersetup key)	13
<code>\newcommand</code>	163, 3477, 3479, 3481
<code>\newcount</code>	2585, 2769
<code>\NewDocumentCommand</code>	2444
<code>\NewExpandableDocumentCommand</code>	380
<code>\newlength</code>	3496
<code>\noexpand</code> ...	2667, 2668, 2669, 2670, 3048
<code>\nolinkurl</code>	4
	O
ocgcolorfile (hypersetup key)	1229
ocgcolorlink (hypersetup key)	1229
ocgcolorlinks (hypersetup key)	1229
ocgcolormenu (hypersetup key)	1229
ocgcolorrun (hypersetup key)	1229
ocgcolorurl (hypersetup key)	1229
	P
pageanchor (hypersetup key)	1433
<code>\paperwidth</code>	3
<code>\PassOptionsToPackage</code> ...	445, 1425, 1430
pdf commands:	
<code>\pdf_bdcobject:nn</code>	1215, 1218
<code>\pdf_destination:nn</code>	27, 155, 620
<code>\pdf_destination:nnnn</code>	613
<code>\pdf_emc:</code>	1217, 1223
<code>\pdf_link_user:nnn</code>	2776
<code>\pdf_name_from_unicode_e:n</code>	
.....	244, 765, 880
<code>\pdf_object_if_exist:nTF</code>	763
<code>\pdf_object_new:nn</code>	1137,
1138, 1139, 1140, 2970, 2971, 2972	
<code>\pdf_object_ref:n</code>	156, 775,
1143, 1145, 1167, 1168, 1171, 1175,	
1180, 1185, 1190, 2562, 2564, 3028	
<code>\pdf_object_ref_last:</code>	796, 2370
<code>\pdf_object_unnamed_write:nn</code> ...	
.....	795, 2366
<code>\pdf_object_write:nn</code>	1141,
1147, 1157, 1169, 2973, 3015, 3022	
<code>\pdf_pageobject_ref:n</code> 1914, 1939, 2600	
<code>\pdf_string_from_unicode:nnN</code> ..	564

pdfmanagement commands:	
\pdfmanagement_add:nn	1620
\pdfmanagement_add:nnn	587, 1167, 1168, 1190, 1496, 1521, 1537, 1541, 1561, 1569, 1598, 1653, 1663, 1687, 1701, 1713, 1718, 1743, 1751, 1766, 1792, 1817, 1849, 1866, 1912, 1937, 1951, 1971, 1998, 2027, 2133, 2151, 2156, 2167, 2185, 2196, 2220, 2234, 2252, 2351, 2370, 2537, 2554, 2561, 2563, 2565, 2571
\pdfmanagement_if_active:p	423
\pdfmanagement_remove:nn	1493, 1517, 1525, 1545, 1557, 1582, 1594, 1602, 1616, 1625, 1669, 1684, 1703, 1740, 1779, 1804, 1830, 1845, 1879, 1909, 1934, 1947, 1956, 1984, 2011, 2023, 2032, 2130, 2160, 2179, 2217, 2231, 2348, 2361, 2575
\pdfmanagement_show:n	2538
pdfmeta commands:	
\pdfmeta_standard_item:n	2333, 2338, 2339
\pdfmeta_standard_verify:nnTF	876, 2792, 2883
\pdfmeta_standard_verify:nTF	2567, 3144
pdfmoddate (hypersetup key)	14, 2209
pdfproducer (hypersetup key)	2138
pdfremotestartview (hypersetup key)	11
pdfstartview (hypersetup key)	11
\pdfstringdef	8, 9, 22, 28, 3093
\pdfstringdefDisableCommands	381, 382, 383
pdfsubject (hypersetup key)	2138
pdftitle (hypersetup key)	2138
pdftrapped (hypersetup key)	2243
pdfversion (hypersetup key)	1401
pdfview (hypersetup key)	11, 2042
pdfxform commands:	
\pdfxform_if_exist:nTF	2732, 2844, 2928
\pdfxform_new:nnn	2446, 2734, 2847, 2851, 2930, 2942
\pdfxform_ref:n	2756, 2757, 2863, 2864, 3175, 3176
\phantom	2951
\phantomsection	12, 19, 3565
plainpages (hypersetup key)	1433
prg commands:	
\prg_do_nothing:	408, 419
\prg_generate_conditional_variant:Nnn	456
\prg_new_eq_conditional:NNn	455
prop commands:	
\prop_const_from_keyval:Nn	489, 497, 3626, 3641, 3656, 3671, 3686, 3701, 3717, 3732
\prop_gput:Nnn	11, 2464, 2471
\prop_if_empty:NTF	2540
\prop_item:Nn	2557
\prop_map_inline:Nn	939, 1003, 1043, 1070, 1078, 1089, 1116, 1124, 1193, 1301, 1337, 1347, 1391, 1479, 2535, 2542
\prop_new:N	2450, 2451
\protect	461, 690, 741, 748, 802, 808, 852, 858, 901, 3498
\providecommand	161, 162, 2478, 2481, 2645, 3033
\ProvidesFile	3
R	
\ReadBookmarks	3487
ref commands:	
\ref_if_exist:nn	455
\ref_label:nn	451
\ref_value:nn	25, 472
\refstepcounter	12
regex commands:	
\regex_const:Nn	538
\regex_extract_once:NnNTF	1892, 1923
\relax	690, 741, 748, 802, 808, 852, 858, 901, 2648, 2649, 2690, 3030, 3077, 3081, 3151, 3155, 3258, 3259, 3267, 3271, 3298, 3302, 3382, 3390, 3417, 3425, 3549, 3558
\RemoveFromHook	1501
\RenewDocumentCommand	172
\RequirePackage	6, 31, 440, 3485
run (hypersetup key)	10, 1479
runborder (hypersetup key)	13
runborderstyle (hypersetup key)	14, 1089
runcolor (hypersetup key)	975
S	
\selectfont	2741, 2938, 2950
seq commands:	
\seq_const_from_clist:Nn	487
\seq_count:N	2051, 2077
\seq_get_right:NN	2053
\seq_item:Nn	1894, 1925, 2047, 2095, 2097, 2107, 2112
\seq_map_inline:Nn	524, 528, 532, 975, 1276, 1374, 2552
\seq_new:N	477
\seq_put_right:Nn	2084, 2085, 2086, 2544

<code>\seq_set_split:Nnn</code>	2046	<code>\@currDisplay</code>	2660, 2728
<code>\seq_sort:Nn</code>	2546	<code>\@currValue</code>	2726, 2756
<code>\setbox</code>	374	<code>\@currentHref</code>	
<code>\setlength</code>	3515, 3586	3519, 3532, 3543, 3573, 3590
<code>\setpdflinkmargin</code>	8	<code>\@curropt</code> ..	2645, 2655, 2658, 2722, 2725
<code>\settowidth</code>	2660	<code>\@empty</code>	2614, 2630, 3098,
<code>\show</code>	2720	3102, 3106, 3110, 3114, 3118, 3122,	
sort commands:		3126, 3130, 3134, 3278, 3309, 3310	
<code>\sort_return_same:</code>	2550	<code>\@endForm</code>	2580
<code>\sort_return_swapped:</code>	2549	<code>\@esphack</code>	452
<code>\space</code>	465, 2728, 3260, 3277,	<code>\@firstofone</code> ..	2484, 3039, 3044, 3061
3278, 3308, 3309, 3383, 3415, 3418		<code>\@firstoftwo</code>	3580
<code>\spacefactor</code>	603, 625	<code>\@for</code>	2656, 2723
<code>\stockwidth</code>	3	<code>\@gobbletwo</code>	3529
str commands:		<code>\@hyper@launch</code>	37, 862
<code>\c_colon_str</code>	862	<code>\@ifnextchar</code>	273
<code>\c_hash_str</code>	732	<code>\@ifundefined</code>	3053
<code>\c_percent_str</code>	733	<code>\@latex@warning</code>	462
<code>\str_case:nnTF</code>	2146, 3440	<code>\@mainaux</code>	2477, 2480, 2500, 2519
<code>\str_case_e:nnTF</code>	2047	<code>\@mkboth</code>	3529
<code>\str_compare:nNnTF</code>	2548	<code>\@nil</code>	3051, 3064, 3070
<code>\str_gset:Nn</code>	1000	<code>\@part</code>	3556, 3563
<code>\str_gset_eq:NN</code>	574	<code>\@pdfauthor</code>	23
<code>\str_head:n</code>	2254, 2259	<code>\@pdfborder</code>	161
<code>\str_if_eq:nnTF</code>		<code>\@pdfborderstyle</code>	162
... 610, 2165, 2183, 2193, 2333, 3142		<code>\@savsf</code>	603, 625
<code>\str_if_eq_p:nn</code>	830, 1231, 1280	<code>\@schapter</code>	3526, 3527
<code>\str_lowercase:n</code>		<code>\@secondoftwo</code>	3582
... 2047, 2188, 2236, 2255, 2260		<code>\@sect</code>	3577, 3578
<code>\str_new:N</code>	479, 484, 485, 523	<code>\@spart</code>	3569, 3570
<code>\str_set:Nn</code>	556	<code>\@ssect</code>	3512, 3513
<code>\str_set_convert:Nnnn</code>	8	<code>\@tempdima</code> ..	2654, 2661, 2663, 2700,
<code>\str_set_eq:NN</code>	576	2701, 2702, 2706, 2707, 2708, 2709	
<code>\str_tail:n</code>	2255, 2260	<code>\@tempdimb</code>	2660, 2661
<code>\str_uppercase:n</code>	2248, 2254, 2259	<code>\@typeset@protect</code>	3498
<code>\string</code>	2478,	<code>\BKM@color</code>	433
2481, 2502, 2520, 3053, 3057, 3068		<code>\c@secnumdepth</code>	3546, 3564, 3579
<code>\subpdfbookmark</code>	3478	<code>\calc@bm@number</code>	3490
sys commands:		<code>\check@bm@number</code>	3489
<code>\c_sys_backend_str</code>	3440	<code>\define@key</code>	427
T			
T _E X and L ^A T _E X 2 _ε commands:			
<code>\@BOOKMARK</code>	3484	<code>\Fld@additionalactions</code> ..	3140, 3210,
<code>\@CheckBox</code>	2906	3244, 3280, 3317, 3364, 3404, 3433	
<code>\@ChoiceMenu</code>	2646	<code>\Fld@align</code>	3169, 3255, 3330
<code>\@Form</code>	2529	<code>\Fld@altname</code> ..	3077, 3079, 3151, 3153
<code>\@PushButton</code>	2784	<code>\Fld@annotflags</code>	3164,
<code>\@Reset</code>	2873	3217, 3250, 3286, 3325, 3374, 3409	
<code>\@Submit</code>	2825	<code>\Fld@annotnames</code>	3149, 3166,
<code>\@TextField</code>	2611	3219, 3252, 3288, 3327, 3376, 3411	
<code>\@bookmarksopenlevel</code>	1512	<code>\Fld@bcolor</code>	3188, 3190, 3271,
<code>\@bsphack</code>	450	3273, 3302, 3304, 3338, 3351, 3353	
<code>\@chapter</code>	3538, 3539	<code>\fld@bcolor</code>	3259
		<code>\Fld@bordercolor</code>	3184,
		3186, 3228, 3237, 3239, 3258, 3267,	

3269, 3298, 3300, 3336, 3347, 3349, 3382, 3390, 3392, 3417, 3425, 3427	\Fld@X@additionalactions
\Fld@borderstyle 3170, 3223, 3256, 3292, 3331, 3380, 3431 3097, 3142, 3145
\Fld@borderwidth . 2707, 2708, 3170, 3223, 3256, 3292, 3331, 3380, 3431	\Form@action 3400
\Fld@calculate@code . 2493, 3110, 3112	\G@refundefinedtrue 461
\Fld@calculate@sortkey 2504	\H@old@part 3556, 3567
\Fld@cbsymbol 3192	\H@old@schapter 3526, 3535
\Fld@charsize	\H@old@ssect 3577, 3588, 3597
.. 2706, 3196, 3277, 3308, 3359, 3415	\H@old@spart 3569, 3575
\Fld@checkequals 2658, 2725	\H@old@ssect 3512, 3517
\Fld@choices 3279	\href@ 273, 279
\Fld@color 3197, 3199, 3278, 3309, 3360	\href@split 279, 280
\Fld@default	\Hy@abspage 597
... 2613, 2630, 2648, 2691, 2720, 2721, 2909, 3310, 3314, 3315, 3362	\Hy@activeanchorfalse 648
\Fld@flags 3168, 3221, 3254, 3290, 3329, 3378, 3413	\Hy@activeanchortrue 639
\Fld@format@code 3102, 3104	\Hy@AtBeginDocument 2458, 2474
\Fld@height	\Hy@bookmarksnumberedfalse ... 1505
2616, 2637, 2640, 2651, 2709, 2748, 2761, 2779, 2828, 2876, 2912, 2957	\Hy@bookmarksnumberedtrue 1506
\Fld@keystroke@code 3098, 3100	\Hy@bookmarksopenfalse 1509
\Fld@listcount 2719, 2727, 2762, 2769	\Hy@bookmarksopentruetrue 1510
\Fld@mappingname 3081, 3083, 3155, 3157	\Hy@bookmarkstype 1513
\Fld@maxlen 3365, 3367	\Hy@chapapp 3530, 3541
\Fld@menulength 2653, 2659, 2706	\Hy@colorlink 34
\Fld@name	\Hy@currentbookmarklevel 3471
.. 2612, 2647, 2785, 2908, 3076, 3150	\Hy@DisableOption 171
\Fld@onblur@code 3118, 3120	\Hy@drafttrue 1424
\Fld@onclick@code 3243	\Hy@escapeform
\Fld@onenter@code 3130, 3132 2634, 2731, 2775, 2803, 2842, 2889, 2927, 3035, 3042, 3045
\Fld@onexit@code 3134, 3136	\Hy@escapestring 3037, 3039, 3044, 3047, 3048, 3051, 3052, 3061, 3064, 3100, 3104, 3108, 3112, 3116, 3120, 3124, 3128, 3132, 3136, 3192, 3243, 3306, 3362, 3363, 3400
\Fld@onfocus@code 3114, 3116	\Hy@finaltrue 1429
\Fld@onmousedown@code ... 3122, 3124	\Hy@FormObjects 2534, 2968, 3030
\Fld@onmouseup@code 3126, 3128	\Hy@gtemp 3093, 3095
\Fld@pageobjref	\Hy@href 273
..... 2582, 2596, 3033, 3165, 3218, 3251, 3287, 3326, 3375, 3410	\Hy@href@nextactionraw 237
\Fld@radiosymbol 3306	\Hy@href@page 225
\Fld@rotation 3180, 3182, 3226, 3233, 3235, 3257, 3263, 3265, 3294, 3296, 3334, 3343, 3345, 3381, 3386, 3388, 3416, 3421, 3423	\Hy@hypertextnamesfalse 1439
\Fld@submitflags 3402	\Hy@hypertextnametrue 1438
\Fld@validate@code 3106, 3108	\Hy@linkfileprefix 1441
\Fld@value	\Hy@linktoc 1464
.. 2614, 2630, 2649, 2690, 2691, 3363	\Hy@localanchornamefalse 1444
\Fld@width 2615, 2629, 2636, 2640, 2650, 2689, 2700, 2702, 2747, 2761, 2779, 2797, 2827, 2838, 2875, 2887, 2911, 2921, 2956	\Hy@localanchornametrue 1443
	\Hy@MakeCurrentHref 198
	\Hy@MakeCurrentHrefAuto
 3514, 3530, 3541, 3571, 3585
	\Hy@naturalnamesfalse 1448
	\Hy@naturalnametrue 1447
	\Hy@next 3540, 3549, 3552
	\Hy@numberline 164, 3472
	\Hy@org@chapter 3538, 3553
	\Hy@OutlineName 3488

\Hy@OutlineRerunCheck	3486	\HyField@SetKeys	2624, 2665, 2668, 2687, 2787, 2830, 2878, 2913
\Hy@pageanchorfalse	1452	\HyPat@ObjRef	3444, 3451, 3458, 3465
\Hy@pageanchortrue	1451	\hyper@link	281
\Hy@pdfmajorversion	166	\hyper@anchor	627
\Hy@pdfminorversion	165	\hyper@anchorend	627, 3507, 3532, 3543, 3573
\Hy@pdfstringtrue	29, 569	\hyper@anchorstart	627, 3507, 3532, 3543, 3573
\Hy@pdfversion	3088	\hyper@link	33, 34, 677
\Hy@plainpagesfalse	1456	\hyper@linkend	33, 34, 705
\Hy@plainpagetrue	1455	\hyper@linkfile	323, 756
\Hy@pstringdef	579, 3068	\hyper@linklaunch	36, 340, 814, 865
\Hy@PutCatalog	585	\hyper@linknamed	38, 174, 871
\Hy@raisedlink	3506, 3531, 3542, 3572	\hyper@linkstart	33, 34, 693
\Hy@RestoreLastskip	624	\hyper@linkurl	306, 367, 718
\Hy@safe@activetrue	734, 2496	\hyper@normalise	273, 302, 318, 335, 361, 374
\Hy@SaveLastskip	604	\HyPL@Labels	587, 597
\Hy@secnum@part	3559, 3561, 3564	\HyPL@storePageLabel	585
\Hy@SectionAnchorHref	3497, 3519, 3590	\HyPsd@SanitizeForOutFile	3475
\Hy@SectionHShift	3496, 3505, 3515, 3586	\if@filesw	2476, 2499, 2518
\Hy@StepCount	2659, 2727	\ifFld@combo	2674, 2704
\Hy@temp	2497, 2512, 3050, 3051	\ifFld@hidden	2629, 2689
\Hy@unicodefalse	3090	\ifFld@multiline	2617
\Hy@VerboseAnchor	605	\ifFld@popdown	2675
\Hy@verbosefalse	1415	\ifFld@radio	2671, 2694
\Hy@VerboseLinkStart	681, 697	\ifHy@implicit	3492
\Hy@VerboseLinkStop	688, 711, 744, 804, 854, 892	\ifHy@pdfescapeform	3036, 3046
\Hy@verbostrue	1414	\ifHy@unicode	3089
\Hy@VersionChecked	157	\kv@set@family@handler	177
\Hy@WrapperDef	601	\ltx@empty	2453, 2454, 2493, 2580, 2581, 2582, 2613
\Hy@xspace@end	685, 743, 803, 853, 891	\ltx@gobble	2456
\HyAnn@AbsPageLabel	2581, 2587, 2633, 2730, 2774, 2802, 2841, 2888, 2926	\ltx@gobbletwo	2457
\HyAnn@Count	2585, 2586, 2589, 2590, 2591, 2592, 2593, 2594, 2598, 2602	\ltx@ifUndefined	3537, 3547
\HyField@AddToFields	2487, 2641, 2763, 2780, 2811, 2867, 2896, 2960	\ltx@one	2589
\HyField@afields	2453	\ltx@ReturnAfterFi	3069
\HyField@AfterAuxOpen	2458, 2484, 2517	\ltx@zero	2586
\HyField@AuxAddToCoFields	2457, 2469, 2481, 2502	\m@ne	3546, 3561
\HyField@AuxAddToFields	2456, 2462, 2478, 2520	\OBJ@OCG@view	163
\HyField@cofields	2454	\p@	2663
\HyField@FlagsCheckBox	2918	\pdf@ifdraftmode	2532
\HyField@FlagsChoice	2699	\PDF@SetupDoc	158
\HyField@FlagsPushButton	2794, 2834, 2885	\PDFForm@Check	2927, 2959, 3161
\HyField@FlagsRadioButton	2695	\PDFForm@List	2775, 2778, 3248
\HyField@FlagsSubmit	2835	\PDFForm@Name	2627, 2688, 2791, 2917, 3075
\HyField@FlagsText	2628	\PDFForm@Push	2803, 2809, 3214
\HyField@PDFChoices	2772	\PDFForm@Radio	2731, 2751, 3283
		\PDFForm@Reset	2889, 2895, 3407
		\PDFForm@Submit	2842, 2861, 3372
		\PDFForm@Text	2634, 2639, 3322
		\protected@edef	653

