The hyperref-generic module
A generic driver for hyperref

The LaTeX Project*

Version 0.95g, released 2021-07-21

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

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

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

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

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

1 Avoiding transition problems

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

- Package options are processed at the end of the driver, Class options are ignored. But not every option already works as package options, in some cases hyperref interferes. So it is recommended for most options —with the exception of a few mentioned below in section 9—to set them in `\hypersetup`, not as package option.

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

- This driver uses 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 \DeclareDocumentMetadata should be used:

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

  The value can be retrieved as document/lang.

4 Dates

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

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

5 PDF page size (mediabox)

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

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

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

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

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

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

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

Some main changes are

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

7.1 Background information

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

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

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

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

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

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

7.2 New Keys

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

- Beside `pdfborder` there are also `linkborder`, `urlborder` etc
- Beside `pdfhighlight` there are also `linkhighlight`, `urlhighlight` etc
- Beside `pdfborderstyle` there are also `linkborderstyle`, `urlborderstyle` etc
- Beside `colorlinks` there are also `colorlink`, `colorurl` etc
- Beside `ocgcolorlinks` there are also `ocgcolorlink`, `ocgcolorurl`, etc TODO
- Beside `hidelinks` there are also `hidelink`, `hideurl`, etc
- `bordercolormodel` allows to set the model used in annotations, the allowed values are `rgb` or `cmyk`. `rgb` is the default. It does *not* change the model of text colors. Be aware that while the PDF format allows cmyk (4 numbers) in the `/C` key of an annotation, this is often ignored by pdf viewers and the colors can be wrong.

- The boolean keys `url`, `link`, `run`, `menu`, `file` allow to deactivate locally the link types.

`colorscheme` (*setup key*) The new key `colorscheme` allows to switch the colors (both for text and borders) with a key word. It takes one of the values `orginal` (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_colorrun_bool,
\l_hyp_annot_colormenu_bool,
\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,
```

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

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

frenchlinks The option `frenchlinks` does nothing at all.

cite colors As mentioned above the support for `citecolor` and `citebordercolor` has been reduced. A package like hyperref can't keep track of such semantic contexts like cite, acronym, glossaries and special references and maintain keys for them. The keys are not completly 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 (undocumentated) 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
- hypertexnames
- 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*)
url (*setup key*)
file (*setup key*)
run (*setup key*)
menu (*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 `\DeclareDocumentMetadata`

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

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

unicode This is always true.

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

```
\DeclareDocumentMetadata{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*)
pdfstartview (*setup key*)
pdfremotestartview (*setup key*)

So with this driver the options pdfview, pdfstartview and pdfremotestartview take the following options:

- 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 is it is quite unpredicable how viewers which support FitR handles zero sizes.
  - pdfstartview and pdfremotestartview will pass the values as absolute coordinates.

14 Assorted key descriptions

The following gives a few details to some keys that are perhaps not completly 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 necessarly 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)>>
```

Typically PDF viewer can handle only GoToR annotions pointing to a PDF. So normally the default value `pdf` of this key should not be changed.

`nesting` (*setup key*) 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*)
`linkborder` (*setup key*)
`urlborder` (*setup key*)
`runborder` (*setup key*)
`menuborder` (*setup key*)

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

`pdfborderstyle` (*setup key*)
`linkborderstyle` (*setup key*)
`urlborderstyle` (*setup key*)
`fileborderstyle` (*setup key*)
`runborderstyle` (*setup key*)
`menuborderstyle` (*setup key*)

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

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

`pdfcreationdate` (*setup key*)
`pdfmoddate` (*setup key*)

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

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

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

File I
hyperref-generic driver implementation

```
⟨*package⟩
⟨@@=hyp⟩
\ProvidesFile{hgeneric-testphase.def}[2021-07-21 v0.95g %
  generic Hyperref driver for the LaTeX PDF management testphase bundle]

\RequirePackage{etoolbox} %why?
\ExplSyntaxOn
```

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

1 messages

Redirect the message name:

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

At first a message for the testing of the resource management

```
\msg_new:nnnn
  { hyp }
  { missing-resource-management }
  { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
  {
     Activate~it~with \\
     \tl_to_str:n{\RequirePackage{pdfmanagement-testphase}}\\
     \tl_to_str:n{\DeclareDocumentMetadata{<options>}}\\
     before~\tl_to_str:n{\documentclass}
  }
```

The pdfversion should be set in `\DeclareDocumentMetadata`

```
\msg_new:nnnn
  { hyp }
  { pdfversion-disabled }
  {
    This~hyperref~driver~ignores~the~pdfversion~key!\\
    Set~the~pdfversion~in~\token_to_str:N \DeclareDocumentMetadata
  }
  {
    For example:\\
    \tl_to_str:n
     {
      \DeclareDocumentMetadata { pdfversion=1.7 }
     }
  }
```

A generic message for ignored keys.

```
\msg_new:nnn
  { hyp }
  { key-dropped }
  {
    This~hyperref~driver~ignores~the~key~#1!\\
    Please~check~the~documentation.
  }
```

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

```
\msg_new:nnn
  { hyp }
  { pdfa-no-push-button }
  { PDF/A:~Push~button~with~JavaScript~is~prohibited }

\msg_new:nnn
  { hyp }
  { pdfa-no-reset-button }
  { PDF/A:~Reset~action~is~prohibited }
```

pdf/A message for not allowed Named actions

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

A message if the destination name is empty.

```
\msg_new:nnn
  { hyp }
  { empty-destination-name }
  {
    Empty~destination~name,\\
            using~‘#1’
  }
```

A message if the destination check fails

```
\msg_new:nnn
  { hyp }
  { invalid-destination-value }
  {
    Invalid~value~‘#1’~of~‘#2’  \\
    is~replaced~by~‘Fit’~\msg_line_context:.
  }
```

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

```
\msg_new:nnn
  { hyp }
  { ignore-deprecated-or-unknown-option-in-pdf-version }
  {
    Option~‘#1’~is~unknown~or~deprecated~in\\
    pdf~version~#2.~Ignored.
  }
\msg_new:nnn
  { hyp }
  { ignore-deprecated-or-unknown-value-in-pdf-version }
  {
    Value~‘#1’~is~unknown~or~deprecated~in\\
    pdf~version~#2.~Ignored.
  }
\msg_new:nnn
  { hyp }
  { replace-deprecated-or-unknown-value-in-pdf-version }
  {
    Value~‘#1’~is~unknown~or~deprecated~in\\
    pdf~version~#2. Value~‘#3’~is used instead.
  }
```

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:

```
\msg_new:nnn
  { hyp }
  { unknown-key }
  {
    unknown~key~#2~of~module~’#1’~set~to~’#3’.
  }
```

```
\msg_new:nnn
  { hyp }
  { unknown-key-to-Hyp }
  {
    ignored~in~family~Hyp~unknown~key~#1.
  }
```

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

```
\cs_new:Npn \__hyp_clist_display:n #1 {*~#1\\}
\msg_new:nnn
  { hyp }
  { unknown-choice }
  {
    Value~'#3'~is~invalid~for~key~'#1'.\\
    The~key~accepts~only~the~choices\\
    \clist_map_function:nN { #2 }\__hyp_clist_display:n
  }

\msg_new:nnn
  { hyp }
  { unknown-choice+empty }
  {
    Value~'#3'~is~invalid~for~key~'#1'.\\
    The~key~accepts~only~the~choices\\
    \clist_map_function:nN { #2 }\__hyp_clist_display:n
    An~empty~value~removes~the~setting.
  }

\msg_new:nnn
  { hyp }
  { no-bool }
  {
    Value~'#2'~is~invalid~for~key~'#1'.\\
    The~key~accepts~only~the~choices\\
    *~true\\
    *~false \\
    *~and~an~empty~value~which~removes~the~setting.\\
    No~value~is~equivalent~to~using~`true`.
  }
```

2 Variants

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

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

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

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

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

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

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

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

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

```
\cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
\cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:
```

```
\legacy_if:nT { Hy@setpdfversion }
 {
    \msg_warning:nn { hyp }{ pdfversion-disabled }
 }
\Hy@DisableOption{pdfversion}
```

`\Acrobatmenu` should use the new internal link command

```
\RenewDocumentCommand \Acrobatmenu { m m }
  {
    \hyper@linknamed {#1} {#2}
  }
```

`\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!

```
%  TODO should go at some time ...
% \kv@set@family@handler{Hyp}
%  { \msg_warning:nnx {hyp}{unknown-key-to-Hyp}{#1} }
\cs_set_protected:Npn \hypersetup #1
  {
     %\kvsetkeys{Hyp} {#1}
     \keys_set:nn { hyp / setup }{ #1 }
  }
% TODO for now unknown keys should only give warnings.
\keys_define:nn { hyp / setup }
  {
    unknown .code:n =
    {
      \msg_warning:nnxxx { hyp } { unknown-key }
      { hyp~/~setup }{ \l_keys_key_str } { #1 }
    }
  }
```

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`)

```

\bool_new:N \l__hyp_href_url_encode_bool
\bool_new:N \l__hyp_href_url_ismap_bool
```

```
\tl_new:N   \l__hyp_href_url_protocol_tl
\tl_new:N   \l__hyp_href_pdf_destination_tl
\tl_new:N   \l__hyp_href_pdf_page_tl
\tl_new:N   \l__hyp_href_run_parameter_tl
\cs_new_protected:Npn \__hyp_href_url_format {\begingroup\Url}


\keys_define:nn { hyp / href }
  {
    ,urlencode  .bool_set:N =  \l__hyp_href_url_encode_bool
    ,format     .code:n = { \cs_set_eq:NN \__hyp_href_url_format #1 },
    ,protocol  .tl_set:N   =  \l__hyp_href_url_protocol_tl
    ,destination .tl_set:N = \l__hyp_href_pdf_destination_tl
    ,pdfremotestartview .code:n =
      {
        \keys_set:nn { hyp / setup }
          { pdfremotestartview = #1 }
      }
    ,page .code:n =
      {
       \tl_set:Nn \l__hyp_href_pdf_page_tl {#1}
       \tl_set:Nn \Hy@href@page {#1}
      }
    ,ismap  .bool_set:N  = \l__hyp_href_url_ismap_bool
    ,run-parameter .tl_set:N = \l__hyp_href_run_parameter_tl
    ,nextactionraw .code:n =
     { %perhaps some safety match later, see hyperref code
        \tl_if_empty:nTF {#1}
         {
           \pdfdict_remove:nn{l_hyp/annot/A}{Next}
         }
         {
           \pdfdict_put:nnn{l_hyp/annot/A}{Next}{#1}
           \tl_set:Nn \Hy@href@nextactionraw {/Next~#1}
           \keys_set:nn {hyp / setup }{ pdfnewwindow = true}
         }
     }
    ,afrelationship .code:n =
      {
        \pdfdict_put:nnx
          { l_pdffile/Filespec}{AFRelationship}{ \pdf_name_from_unicode_e:n {#1}}
      }

  }

\keys_define:nn { hyp / setup }
  {
    ,href / urlencode .bool_set:N =  \l__hyp_href_url_encode_bool
    ,href / urlencode .default:n  = {true}
    ,href / urlencode .initial:n  = {false}
    ,href / protocol  .tl_set:N   =   \l__hyp_href_url_protocol_tl
    ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
    ,href / format  .code:n = { \cs_set_eq:NN \__hyp_href_url_format #1 }
  }
```

```

\hook_new_pair:nn{cmd/href/before}{cmd/href/after}

\DeclareRobustCommand*{\href}[1][]{%
  \mode_leave_vertical:
  \hook_use:n{cmd/href/before}
  \group_begin:
  \keys_set:nn { hyp / href } {#1}
  \let\protect\@unexpandable@protect
  \bool_if:NTF \l__hyp_href_url_encode_bool
    {
      \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
    }
    {
      \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
    }
  \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
}

\begingroup
  \catcode‘\$=6 %
  \catcode‘\#=12 %
  \gdef\href@$1{\expandafter\href@split$1##\\}%
  \gdef\href@split$1#$2#$3\\$4{%
    \hyper@@link{$1}{$2}{$4}%<---__hyp-docstrip doubling!
    \endgroup
    \hook_use:n{cmd/href/after}
  }%
\endgroup

\hook_new_pair:nn{cmd/hrefurl/before}{cmd/hrefurl/after}

\DeclareRobustCommand*{\hrefurl}[1][]
 {
   \mode_leave_vertical:
   \hook_use:n{cmd/href/before}
   \group_begin:
   \keys_set:nn { hyp / href } {#1}
   \let\protect\@unexpandable@protect
   \bool_if:NTF \l__hyp_href_url_encode_bool
    {
      \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
    }
    {
      \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
    }
  \hyper@normalise\__hyp_href_url_aux:nn}

\cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
  {
    \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
    \group_end:
    \hook_use:n{cmd/href/after}
  }
```

```

\hook_new_pair:nn{cmd/hrefpdf/before}{cmd/hrefpdf/after}
\DeclareRobustCommand*{\hrefpdf}[1][]
 {
  \mode_leave_vertical:
  \hook_use:n{cmd/hrefpdf/before}
  \group_begin:
  \keys_set:nn { hyp / href } {#1}
  \let\protect\@unexpandable@protect
  \hyper@normalise\__hyp_href_pdf_aux:nn
 }

\cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
  {
    \exp_args:Nnno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
    \group_end:
    \hook_use:n{cmd/hrefpdf/after}
  }

\hook_new_pair:nn{cmd/hrefrun/before}{cmd/hrefrun/after}
\DeclareRobustCommand*{\hrefrun}[1][]
 {
  \mode_leave_vertical:
  \hook_use:n{cmd/hrefrun/before}
  \group_begin:
  \keys_set:nn { hyp / href } {#1}
  \let\protect\@unexpandable@protect
  \hyper@normalise\__hyp_href_run_aux:nn
 }

\cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
  {
    \exp_args:Nnno\hyper@linklaunch{#1}{#2}{\l__hyp_href_run_parameter_tl}
    \group_end:
    \hook_use:n{cmd/hrefrun/after}
  }


\hook_new_pair:nn{cmd/url/before}{cmd/url/after}

\DeclareRobustCommand*{\url}[1][]
  {
   \mode_leave_vertical:
   \hook_use:n{cmd/url/before}
   \group_begin:
   \keys_set:nn {hyp / href } {#1}
   \let\protect \@unexpandable@protect
    \bool_if:NTF \l__hyp_href_url_encode_bool
    {
      \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
    }
    {
      \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
    }
   \hyper@normalise\__hyp_href_url_aux:n
  }

\cs_new_protected:Npn \__hyp_href_url_aux:n #1
 {
   \exp_args:Nno
     \hyper@linkurl{\__hyp_href_url_format {#1}}
     {\l__hyp_href_url_protocol_tl#1}
   \group_end:
   \hook_use:n{cmd/url/after}
 }

```

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

```

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

```

make the new commands compatible with `\pdfstringdef`:

```
\NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
\pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
\pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}
\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.

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

```
\hook_new:n{hyp/link/cite}
\color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
\color_set:nn  {hyp/color/citeborder}{hyp/color/cite!60!white}
\keys_define:nn { hyp / setup }
  {
    ,citecolor .code:n = {\__hyp_color_set:ne {hyp/color/cite}{#1}}
    ,citebordercolor
```

```
      .code:n = {\__hyp_color_set:ne {hyp/color/citeborder}{#1}}
  }
\hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
  {
    \keys_set:nn { hyp / setup }
      {
        ,linkbordercolor= hyp/color/citeborder
        ,linkcolor      = hyp/color/cite
      }
  }
```

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

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

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

```
\legacy_if:nT { Hy@bookmarks }
  {
    \AddToHook{begindocument/before}[hyperref/bookmark]{\RequirePackage{bookmark}}
  }
\legacy_if:nT {Hy@draft}
  {
    \PassOptionsToPackage{draft}{bookmark}
  }
```

6 Reference and label commands

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

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

__hyp_ref_label:nn
__hyp_ref_if_exist:nn
__hyp_ref_check:nn
__hyp_ref_value:nn

```
%
\cs_new_protected:Npn \__hyp_ref_label:nn #1 #2 %label/attributes
  {
    \@bsphack
      \ref_label:nn{#1}{#2}
    \@esphack
  }
\cs_generate_variant:Nn \__hyp_ref_label:nn {en}
```

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

```
\prg_new_eq_conditional:NNn \__hyp_ref_if_exist:nn \ref_if_exist:nn { p , T , F,  TF }
\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.

```
\cs_new_protected:Npn \__hyp_ref_check:nn #1 #2 %label/attribute
  {
    \__hyp_ref_if_exist:nnF {#1}{#2}
      {
        \protect\G@refundefinedtrue
        \@latex@warning
          {
            Reference~`\tl_to_str:n {#1}'~with~attribute~`\tl_to_str:n {#2}'~
            on~page~\thepage~\space undefined
          }
      }
  }
\cs_generate_variant:Nn \__hyp_ref_check:nn {en}
```

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

```
\cs_new:Npn \__hyp_ref_value:nn #1 #2
  {
    \ref_value:nn{#1}{#2}
  }
\cs_generate_variant:Nn \__hyp_ref_value:nn {en}
```

(*End definition for* `\__hyp_ref_label:nn` *and others.*)

7 Variables

7.1 Private temporary variables

At first a few generic tmp variables

\l__hyp_tmpa_tl
\l__hyp_tmpa_seq
\l__hyp_tmpa_int
\l__hyp_tmpa_box
\l__hyp_tmpa_str

```
\box_new:N \l__hyp_tmpa_box
\tl_new:N  \l__hyp_tmpa_tl
\seq_new:N \l__hyp_tmpa_seq
\int_new:N \l__hyp_tmpa_int
\str_new:N \l__hyp_tmpa_str
```

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

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

\l__hyp_dest_name_tmpa_tl
\l__hyp_uri_tmpa_tl
\l__hyp_filename_tmpa_tl
__hyp_text_tmpa_str␣␣␣\g__hyp_text_tmpa_str

TODO: document and check use!

```
\tl_new:N  \l__hyp_dest_name_tmpa_tl
\tl_new:N  \l__hyp_uri_tmpa_tl
\tl_new:N  \l__hyp_filename_tmpa_tl
\tl_new:N  \l__hyp_para_tmpa_tl
\str_new:N \l__hyp_text_tmpa_str
\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.

```
\tl_const:Nn \c__hyp_dest_undefined_tl {UNDEFINED}
```

(*End definition for* `\c__hyp_dest_undefined_tl`*.*)

\c__hyp_annot_types_seq
\c__hyp_map_annot_hyp_prop
\c__hyp_map_hyp_annot_prop

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

```
\seq_const_from_clist:Nn \c__hyp_annot_types_seq
 {url,link,file,menu,run}
\prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
  {
    URI   = url,
    GoTo  = link,
    GoToR = file,
    Named = menu,
    Launch= run
  }
\prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
  {
    url   = URI,
    link  = GoTo,
    file  = GoToR,
    menu  = Named,
    run   = Launch
  }

```

(*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
\g__hyp_dest_pdfstartview_tl
\l__hyp_dest_pdfremotestartview_tl

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

```
\tl_new:N  \g__hyp_dest_pdfstartpage_tl
\tl_new:N  \g__hyp_dest_pdfstartview_tl
\tl_new:N  \l__hyp_dest_pdfremotestartview_tl
```

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

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

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

```
\tl_new:N  \l__hyp_text_enc_uri_print_tl
\tl_new:N  \l__hyp_text_enc_info_print_tl
\tl_new:N  \l__hyp_text_enc_dest_tl
\tl_new:N  \l__hyp_text_enc_dest_print_tl
\tl_new:N  \l__hyp_text_enc_file_print_tl
\tl_new:N  \l__hyp_text_enc_para_print_tl
```

```
\tl_set:Nn \l__hyp_text_enc_uri_print_tl  {utf8/URI}
\tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
\tl_set:Nn \l__hyp_text_enc_dest_tl       {utf8/string-raw}
\tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
\tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
\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.

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

(*End definition for* `\l__hyp_dest_pdfview_tl`.)

`hyp/annot/link` (*color name*)
`hyp/annot/url` (*color name*)
`hyp/annot/file` (*color name*)
`hyp/annot/run` (*color name*)
`hyp/annot/menu` (*color name*)

These color names are used for the annotations (colorlinks). They are initialized at the end when the color scheme is used

`\g__hyp_bordercolormodel_str`

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`

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

(*End definition for* `\g__hyp_bordercolormodel_str`.)

7.4 Booleans

`\l_hyp_annot_colorlink_bool`
`\l_hyp_annot_colorurl_bool`
`\l_hyp_annot_colorfile_bool`
`\l_hyp_annot_colorrun_bool`
`\l_hyp_annot_colormenu_bool`

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

```
\seq_map_inline:Nn \c__hyp_annot_types_seq
  {
    \bool_new:c {l_hyp_annot_color#1_bool}
  }
```

(*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.

```
\seq_map_inline:Nn \c__hyp_annot_types_seq
  {
    \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
  }
```

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

not_Named_bool␣␣␣␣\l__hyp_annot_Launch_bool

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

```
\seq_map_inline:Nn \c_pdfannot_link_types_seq
  {
    \bool_new:c {l__hyp_annot_#1_bool}
    \bool_set_true:c {l__hyp_annot_#1_bool}
  }
```

(*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.

```
\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.

```
\regex_const:Nn \c__hyp_dest_startview_regex
  {
    \A\ *
     (?:
      (?:XYZ (?:\ +(?:(?:\d+|\d*\.\d+)|null)){3}\ )
      |
      (?:Fit\b|FitB\b)
      |
      (?:(?:FitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1})
      |
      (?:FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
     )
  }
```

(*End definition for* `\c__hyp_dest_startview_regex`*.*)

7.7 PDF dictionaries

l__hyp_page/Trans

This dictionary is used for page transitions.

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

```
\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

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

(*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

```
\cs_new_protected:Npn \__hyp_text_cleanup:N #1
 {

 }
```

(*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 l3pdftools.

#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

```
\cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
  {
     \pdf_string_from_unicode:nVN { #1 } #2 #2
  }
```

(*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

```
\cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
  {
    \group_begin:
    \Hy@pdfstringtrue
    \hook_use:n {hyp/text/pdfstring}
    \__hyp_text_purify:nN  { #1 } \l__hyp_text_tmpa_str
    \__hyp_text_cleanup:N         \l__hyp_text_tmpa_str
    \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
    \str_gset_eq:NN \g__hyp_text_tmpa_str\l__hyp_text_tmpa_str
    \group_end:
    \str_set_eq:NN #3 \g__hyp_text_tmpa_str
 }
\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

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

This is a special version for info keys:

__hyp_text_pdfstring_info:nN

```
\cs_new_protected:Npn \__hyp_text_pdfstring_info:nN #1 #2
  {
    \__hyp_text_pdfstring:noN  { #1 }{ \l__hyp_text_enc_info_print_tl } #2
  }
```

(*End definition for* `\__hyp_text_pdfstring_info:nN`.)

9 Pagelabels

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

The current implementation in hyperref/hyperref drivers:

xetex: hxetex.def, line 80-110
`\HyPL@StorePageLabel` writes to the aux-file at begin document (after reading the aux) `\HyPL@SetPageLabels` is called (defined in hyperref.sty after the driver loading) which calls `\Hy@PutCatalog{/PageLabels<</Nums[\HyPL@Labels]>>}`

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

dvipdfm: identical to xetex

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

luatex identical to pdftex

The code in hyperref inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-28 is correctly split. If the counter can not be identified hyperref generates only /P entries with the whole content.

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

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

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

__hyp_PageLabels_gpush:
\Hy@PutCatalog
\HyPL@StorePageLabel

```
\cs_new_protected:Npn\__hyp_PageLabels_gpush:
  {
    \pdfmanagement_add:nnx {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
  }

\def\Hy@PutCatalog #1 {}


\legacy_if:nT { Hy@pdfpagelabels }
  {
    \cs_set_protected:Npn \HyPL@StorePageLabel #1
      {
        \tl_gput_right:Nx \HyPL@Labels { \the\Hy@abspage<<#1>> }
        \__hyp_PageLabels_gpush:
      }
  }
```

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

10 Core Hyperref Commands

Every hyperref has to define eight core command:

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

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

10.1 Anchors / destinations

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

`\__hyp_destination:nn`

`\__hyp_destination:nn {⟨destination name⟩} {⟨location⟩}`

The ⟨*destination name*⟩ is encoded with the method stored in 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`

```
\Hy@WrapperDef \__hyp_destination:nn #1 #2
  {
    \mode_if_horizontal:T { \@savsf\spacefactor }
    \Hy@SaveLastskip      %defined in hyperref
    \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
    \__hyp_text_pdfstring:xoN
      { \HyperDestNameFilter{#1}  }
      { \l__hyp_text_enc_dest_tl }
      \l__hyp_tmpa_tl
    \str_if_eq:nnTF  {#2} {fitrbox}
      {
        \exp_args:NV
        \pdf_destination:nnnn \l__hyp_tmpa_tl
         { \box_wd:N \l__hyp_dest_box }
         { \box_ht:N \l__hyp_dest_box }
         { \box_dp:N \l__hyp_dest_box }
      }
      {
        \exp_args:NV
          \pdf_destination:nf
          { \l__hyp_tmpa_tl }
          { #2 }
      }
    \Hy@RestoreLastskip   %defined in hyperref
    \mode_if_horizontal:T { \spacefactor\@savsf }
  }
```

(*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`
`\hyper@anchorend`
`hyp/anchor`
`\l_hyp_current_dest_name_tl`

```
\tl_new:N\l_hyp_current_dest_name_tl
\hook_new:n{hyp/anchor}
\cs_new_protected:Npn \hyper@anchor #1
  {
    \exp_args:NnV
      \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
    \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
    \hook_use:n{hyp/anchor}
  }

\cs_new_protected:Npn \hyper@anchorstart #1
  {
    \Hy@activeanchortrue
    \exp_args:NnV
      \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
    \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
    \hook_use:n{hyp/anchor}
  }

\cs_new_protected:Npn \hyper@anchorend
  {
    \Hy@activeanchorfalse
  }
```

(*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:

```
\cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
  {
    \mode_leave_vertical:
    \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
    \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
      {
        \msg_warning:nnx
          { hyp }
          { empty-destination-name }
          { \c__hyp_dest_undefined_tl }
        \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
      }
      {
        \__hyp_text_pdfstring:xoN
          { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
          { \l__hyp_text_enc_dest_tl }
          \l__hyp_dest_name_tmpa_tl
      }
    \exp_args:No
      \pdfannot_link_goto_begin:nw  { \l__hyp_dest_name_tmpa_tl }
  }

\cs_new_protected:Npn \__hyp_link_goto_end:
  {
    \pdfannot_link_goto_end:
  }
```

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

`\hyper@link` `\hyper@link{⟨context⟩}{⟨destination name⟩}{⟨link text⟩}`

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

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

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

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

\hyper@linkstart
\hyper@linkend

`\hyper@linkstart{⟨context⟩}{⟨destination name⟩}`
`\hyper@linkend`

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

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

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

```

\cs_new_protected:Npn \hyper@link #1 #2 #3 %#1 context, #2=destination name, #3 content
  {
    \bool_if:NTF \l__hyp_annot_GoTo_bool
      {
        \Hy@VerboseLinkStart{#1}{#2}
        \group_begin:
        \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
        \hook_use:n {hyp/link/#1}
        \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
        \__hyp_link_goto_end:
        \group_end:
        \Hy@VerboseLinkStop
      }
      {{\let\protect\relax#3}}
  }

\cs_new_protected:Npn \hyper@linkstart #1 #2 %#1 context, #2=destination name
  {
    \bool_if:NT \l__hyp_annot_GoTo_bool
      {
        \Hy@VerboseLinkStart{#1}{#2}% only for debug
        \group_begin:
        \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
        \hook_use:n {hyp/link/#1}
        \__hyp_link_goto_begin:nw {#2}
      }
  }

\cs_new_protected:Npn \hyper@linkend
  {
    \bool_if:NT \l__hyp_annot_GoTo_bool
      {
        \__hyp_link_goto_end:
        \group_end:
        \Hy@VerboseLinkStop
      }
  }
```

10.3 URI links

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

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

```
\pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
\pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}

\cs_new_protected:Npn \hyper@linkurl #1 #2  %#1:link text #2: URI,
  {
    \bool_if:NTF \l__hyp_annot_URI_bool
      {
        \group_begin:
         \__hyp_text_pdfstring:xoN
           { #2}
           { \l__hyp_text_enc_uri_print_tl }
           \l__hyp_uri_tmpa_tl
         \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
         \bool_if:NT \l__hyp_href_url_ismap_bool
           {
            \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
           }
         \cs_set_eq:NN \# \c_hash_str
         \cs_set_eq:NN \% \c_percent_str
         \Hy@safe@activestrue
         \mode_leave_vertical:
         \pdfannot_dict_put:nnx {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
         \pdfannot_link:nxn { URI }
           {
           }
           {
             \let\protect\relax
             #1
             \Hy@xspace@end
             \Hy@VerboseLinkStop %where is the start??
           }
         \group_end:
      }
      {{\let\protect\relax#1}}
  }

```

10.4 GoToR Links files

```
\pdfdict_new:n   {l_hyp/annot/A/GoToR}
\pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
\pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}


\cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
  {
   \bool_if:NTF \l__hyp_annot_GoToR_bool
     {
        \group_begin:
        \tl_set:Nx \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
        \exp_args:Nx
          \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
            {
              \pdfdict_put:nnx { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n {ap
              \exp_args:Nnox
              \pdffile_embed_file:nnn
                {}
                {\l__hyp_filename_tmpa_tl }
                {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
            }
        \pdfdict_put:nnx
           {l_hyp/annot/A/GoToR}
           {F}
           {\pdf_object_ref:e {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
        \__hyp_text_pdfstring:nnN
          { #3 }
          { \l__hyp_text_enc_dest_print_tl }
          \l__hyp_dest_name_tmpa_tl
        \tl_if_blank:eTF {#3}
          {
            \pdfdict_put:nnx {l_hyp/annot/A/GoToR}{D}
              {
               [
                 \int_eval:n
                  { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
                 /\l__hyp_dest_pdfremotestartview_tl
               ]
              }
          }
          {
            \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
          }
        \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

```
        \pdf_object_unnamed_write:nx{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
        \pdfannot_dict_put:nnx {link/GoToR}{A}{\pdf_object_ref_last:}
        \pdfannot_link:nxn %expansion??
          { GoToR }
          {
          }
          {
           \let\protect\relax
            #1\Hy@xspace@end
            \Hy@VerboseLinkStop %where is the start??
          }
        \group_end:
     }
     {{\let\protect\relax#1}}
  }
```

10.5 Launch links

We define \hyper@linklaunch for naming consistency

```
\pdfdict_new:n   {l_hyp/annot/A/Launch}
\pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
\pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}

```

```
\cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
  {
   \bool_if:NTF \l__hyp_annot_Launch_bool
     {
        \group_begin:
          \__hyp_text_pdfstring:nnN
            { #1 }
            { \l__hyp_text_enc_file_print_tl }
            \l__hyp_filename_tmpa_tl
          \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
          \__hyp_text_pdfstring:noN
            { #3 }
            { \l__hyp_text_enc_para_print_tl }
            \l__hyp_para_tmpa_tl
          \bool_if:nTF
            {
              \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {()}
              ||
              \pdf_version_compare_p:Nn > {1.9}
            }
            {
              \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
            }
            {
              \pdfdict_put:nnx {l_hyp/annot/A/Launch}{Win}
                {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
            }
          \mode_leave_vertical:
          \pdfannot_dict_put:nnx {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}>>}
          \pdfannot_link:nxn
            { Launch }
            {
  %            /A
%              <<
%                \pdfdict_use:n {l_hyp/annot/A/Launch}
%              >>
            }
            {
              \let\protect\relax
              #2\Hy@xspace@end
              \Hy@VerboseLinkStop %where is the start??
            }
        \group_end:
      }
      {{\let\protect\relax#2}}
  }
```

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

```
\use:x
  { % filename, anchor text, linkname
    \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str ##1 \exp_not:N \\ ##2 #
  }
  {
    \hyper@linklaunch {#1}{#2}{#3}
  }
```

10.6 Named links (menu)

We also define \hyper@linknamed for consistency.

```
\pdfdict_new:n   {l_hyp/annot/A/Named}
\pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
\pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}

\cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
  {
    \bool_if:NTF \l__hyp_annot_Named_bool
      {
         \group_begin:
         \pdfmeta_standard_verify:nnTF {named_actions}{#1}
           {
             \mode_leave_vertical:
             \pdfdict_put:nnx {l_hyp/annot/A/Named}{N}
               {\pdf_name_from_unicode_e:n{#1}}
             \pdfannot_dict_put:nnx {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}>>
             \pdfannot_link:nxn { Named }
              {
%                /A
%                  <<
%                    \pdfdict_use:n { l_hyp/annot/A/Named }
%                  >>
              }
              {
                #2
                \Hy@xspace@end
                \Hy@VerboseLinkStop
              }
           }
           {
             \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
             #2
           }
        \group_end:
      }
      {{\let\protect\relax#2}}
  }

```

11 Link decorations

11.1 Functions to export and select colors

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

__hyp_color_export:nnN

__hyp_color_export:nnN {⟨*color*⟩} {⟨*export format*⟩} ⟨*tlvar*⟩

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

__hyp_color_export:nnN
__hyp_color_export_aux:wnnN

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

```
\cs_new_protected:Npn \__hyp_color_export:nnN #1  #2  #3
 {
   \tl_if_head_eq_charcode:nNTF {#1}[ %]
    {
      \__hyp_color_export_aux:wnnN  #1 {#2} #3
    }
    {
      \color_export:nnN {#1} {#2} #3
    }
 }

\cs_new_protected:Npn \__hyp_color_export_aux:wnnN  [#1] #2 #3 #4
  {
     \color_export:nnnN {#1}{#2}{#3}#4
  }

\cs_generate_variant:Nn \__hyp_color_export:nnN {xVN}
```

(*End definition for* `\__hyp_color_export:nnN` *and* `\__hyp_color_export_aux:wnnN`.)

__hyp_color_select:n

__hyp_color_select:nN {⟨*color*⟩}

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

__hyp_color_select:n
__hyp_color_select_aux:wn

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

```
\cs_new_protected:Npn \__hyp_color_select:n #1
 {
   \tl_if_head_eq_charcode:nNTF {#1}[ %]
    {
      \__hyp_color_select_aux:wn  #1
    }
    {
      \color_select:n {#1}
    }
 }

\cs_new_protected:Npn \__hyp_color_select_aux:wn  [#1] #2
  {
     \color_select:nn {#1}{#2}
  }

\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`
`\__hyp_color_set_aux:nwn`

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

```
\cs_new_protected:Npn \__hyp_color_set:nn #1 #2
 {
   \tl_if_head_eq_charcode:nNTF {#2}[ %]
    {
      \__hyp_color_set_aux:nwn  { #1 } #2
    }
    {
      \color_set:nn {#1} {#2}
    }
 }

\cs_new_protected:Npn \__hyp_color_set_aux:nwn #1 [#2] #3
  {
     \color_set:nnn {#1}{#2}{#3}
  }

\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.

```
\prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
  {
     \hook_gput_code:nnn
       {pdfannot/link/#2/begin}
       {hyp/color}
       {
         \bool_if:cT { l_hyp_annot_color#1_bool }
           {
             \group_begin:
             \color_select:n { hyp/color/#1}
           }
       }
     \hook_gput_code:nnn
       {pdfannot/link/#2/end}
       {hyp/color}
       {
         \bool_if:cT { l_hyp_annot_color#1_bool }
           {
             \group_end:
           }
```

```
      }
  }
```

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

```
\keys_define:nn { hyp / setup }
  {
    ,colorlinks .meta:n =
      {
        ,pdfborder={0~0~0}
        ,pdfborderstyle=
        ,colorurl  =#1
        ,colorlink =#1
        ,colorrun  =#1
        ,colormenu =#1
        ,colorfile =#1
        }
    ,colorlinks .default:n = {true}
  }
```

colorurl (*setup key*)
colorlink (*setup key*)
colorrun (*setup key*)
colormenu (*setup key*)
colorfile (*setup key*)
urlcolor (*setup key*)
linkcolor (*setup key*)
runcolor (*setup key*)
menucolor (*setup key*)
filecolor (*setup key*)
allcolor (*setup key*)

```
\seq_map_inline:Nn \c__hyp_annot_types_seq
  {
     \keys_define:nn { hyp / setup }
       {
         ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
         ,#1color .code:n =     { \__hyp_color_set:ne {hyp/color/#1}{##1} }
       }
  }

\keys_define:nn { hyp / setup }
  {
    ,allcolors .meta:n =
      {
        ,urlcolor=#1
        ,linkcolor=#1
        ,runcolor=#1
        ,filecolor=#1
        ,menucolor=#1
      }
    ,allcolors .value_required:n = true
  }
```

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*)

```
\keys_define:nn { hyp / setup }
  {
    ,bordercolormodel .choices:nn =
     {rgb,cmyk}
     { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
    ,bordercolormodel .initial:n ={rgb}
  }
\prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
  {
    \keys_define:nn { hyp / setup }
      {
        #1bordercolor .code:n =
          {
             \tl_if_empty:nTF { ##1 }
               {
                 \pdfannot_dict_remove:nn
                  {link/#2}
                  { C }
               }
               {
                 \__hyp_color_export:xVN {##1}\g__hyp_bordercolormodel_str \l__hyp_tmpa_tl
                 \pdfannot_dict_put:nnx
                   {link/#2}
                   { C }
                   { [\l__hyp_tmpa_tl] }
               }
          }
      }
  }

\keys_define:nn { hyp / setup }
  {
    ,allbordercolors .meta:n =
      {
        ,linkbordercolor=#1
        ,urlbordercolor =#1
        ,filebordercolor=#1
        ,menubordercolor=#1
        ,runbordercolor =#1
      }
    ,allbordercolors .value_required:n = true
  }

```

11.3.2 Borderwidth and -arc

```
\prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
  {
    \keys_define:nn { hyp / setup }
      {
        #1border .code:n =
          {
             \tl_if_empty:nTF { ##1 }
               {
                 \pdfannot_dict_remove:nn
                  {link/#2}
                  { Border }
               }
               {
                 \pdfannot_dict_put:nnn
                   {link/#2}
                   { Border }
                   { [##1] }
               }
          }
      }
  }
\keys_define:nn { hyp / setup }
  {
    ,pdfborder  .code:n =
      {
        \tl_if_empty:nTF { #1 }
         {
           \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
             {
               \pdfannot_dict_remove:nn
                  {link/##2}
                  { Border }
             }
         }
         {
           \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
             {
               \pdfannot_dict_put:nnn
                 {link/##2}
                 { Border }
                 { [#1] }
             }
         }
      }
    ,pdfborder .initial:n = {0~0~1},
  }
```

11.3.3 Borderstyle

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

`pdfborderstyle` (*setup key*)
`urlborderstyle` (*setup key*)
`linkborderstyle` (*setup key*)
`runborderstyle` (*setup key*)
`fileborderstyle` (*setup key*)
`menuborderstyle` (*setup key*)

```
\prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
  {
    \keys_define:nn { hyp / setup }
      {
        #1borderstyle .code:n =
          {
             \tl_if_empty:nTF { ##1 }
               {
                 \pdfannot_dict_remove:nn
                  {link/#2}
                  { BS }
               }
               {
                 \pdfannot_dict_put:nnn
                   {link/#2}
                   { BS }
                   { <<##1>> }
               }
          }
      }
  }
\keys_define:nn { hyp / setup }
  {
    ,pdfborderstyle  .code:n =
      {
        \tl_if_empty:nTF { #1 }
         {
           \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
             {
               \pdfannot_dict_remove:nn
                  {link/##2}
                  { BS }
             }
         }
         {
           \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
             {
               \pdfannot_dict_put:nnn
                 {link/##2}
                 { BS }
                 { <<#1>> }
             }
         }
      }
    ,pdfborderstyle .initial:n = {},
  }
```

11.4 ocgcolorlinks

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

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

```
\cs_new_protected:Npn \__hyp_ocg_init:
  {
     \pdf_object_new:nn   { __hyp/OCG/View }   { dict }
     \pdf_object_new:nn   { __hyp/OCG/Print }  { dict }
     \pdf_object_new:nn   { __hyp/OCG/config } { dict }
     \pdf_object_new:nn   { __hyp/OCG/refarray }   { array }
        \pdf_object_write:nx { __hyp/OCG/refarray }
          {
            \pdf_object_ref:n { __hyp/OCG/View }
            \c_space_tl
            \pdf_object_ref:n { __hyp/OCG/Print }
          }
        \pdf_object_write:nn { __hyp/OCG/View }
          {
            /Type/OCG
            /Name(View)
            /Usage
             <<
              /Print <</PrintState/OFF>>~
              /View  <</ViewState/ON  >>~
             >>
          }
        \pdf_object_write:nn { __hyp/OCG/Print }
          {
            /Type/OCG
            /Name(Print)
            /Usage
             <<
              /Print <</PrintState/ON>>~
              /View  <</ViewState/OFF>>~
             >>
          }
       \pdfmanagement_add:nnx { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OC
       \pdfmanagement_add:nnx { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OC
       \pdf_object_write:nx { __hyp/OCG/config }
         {
           /OFF[\pdf_object_ref:n { __hyp/OCG/Print }]
           /AS[
             <<
              /Event/View
              /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
              /Category[/View]
             >>
             <<
              /Event/Print
              /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
              /Category[/Print]
             >>
             <<
              /Event/Export
              /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
              /Category[/Print]
             >>
              ]
         }
       \pdfmanagement_add:nnx { Catalog / OCProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG
       \cs_gset:Npn \__hyp_ocg_init: {}
  }
```

(*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.

```
\prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
  {
     \hook_gput_code:nnn
       {pdfannot/link/#2/begin}
       {hyp/ocg}
       {
         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
```

```
           {
             \__hyp_ocg_init:
             \group_begin:
             \hbox_set:Nw \l__hyp_tmpa_box
           }
       }
     \hook_gput_code:nnn
       {pdfannot/link/#2/end}
       {hyp/ocg}
       {
         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
           {
             \hbox_set_end:
               \mbox
                 {
                   \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
                   \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
                   \pdf_emc:
                   \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
                   \group_begin:
                   \color_select:n { hyp/color/#1 }
                   \box_use_drop:N \l__hyp_tmpa_box
                   \group_end:
                   \pdf_emc:
                 }
             \group_end:
           }
       }
  }
```

ocgcolorlinks (*setup key*)
ocgcolorlink (*setup key*)
ocgcolorurl (*setup key*)
ocgcolorfile (*setup key*)
ocgcolormenu (*setup key*)
ocgcolorrun (*setup key*)

These are the keys for ocgcolors. We try to disable it for pdf version below 1.5

```
\bool_lazy_or:nnTF
  { \pdf_version_compare_p:Nn > {1.4} }
  { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
  {
     \keys_define:nn { hyp / setup }
       {
         ,_ocgcolorlinks .meta:n =
            {
              ocgcolorlink=#1,
              ocgcolorurl=#1,
              ocgcolorfile=#1,
              ocgcolorrun=#1,
              ocgcolormenu=#1
            }
         ,_ocgcolorlinks .default:n = true
       }
  }
  {
    \keys_define:nn { hyp / setup }
      {
        ,_ocgcolorlinks .code:n =
          {
            \msg_warning:nnxx
```

```
              { hyp }
              { ignore-deprecated-or-unknown-option-in-pdf-version }
              { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
          }
      }
  }

\keys_define:nn { hyp / setup }
  {
    ,ocgcolorlinks .choice:
    ,ocgcolorlinks / true .meta:n =
      {
        pdfborder      ={0~0~0},
        pdfborderstyle ={},
        colorlinks     = false,
        _ocgcolorlinks = true
      }
    ,ocgcolorlinks / false .meta:n =
      {
        _ocgcolorlinks = false
      }
    ,ocgcolorlinks .default:n = {true}
  }

\seq_map_inline:Nn \c__hyp_annot_types_seq
  {
    \bool_lazy_or:nnTF
      { \pdf_version_compare_p:Nn > {1.4} }
      { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
      {
        \keys_define:nn { hyp / setup }
          {
            ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
          }
      }
      {
        \keys_define:nn { hyp / setup }
          {
            ,ocgcolor#1 .code:n=
              {
                \msg_warning:nnxx
                  { hyp }
                  { ignore-deprecated-or-unknown-option-in-pdf-version }
                  { ocgcolor#1 }
                  { \pdf_version_major:.\pdf_version_minor: }
              }
          }
      }
  }
```

11.5 Highlighting

This keys set what happens if you click on a link

```
\prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
```

```
  {
    \keys_define:nn { hyp / setup }
      {
        ,#1highlight .choices:nn =
          { /I, /N, /O, /P}
          {
                \pdfannot_dict_put:nnn
                  {link/#2}
                  { H }
                  { ##1 }

          }
        ,#1highlight / .code:n =
         {
           \pdfannot_dict_remove:nn
             {link/#2}
             { H }

          }
        ,#1highlight / unknown .code:n =
          {
           \msg_warning:nnxxx { hyp } { unknown-choice+empty }
             { pdfhighlight }
             { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
             { \exp_not:n {##1} }
          }
      }
  }


\keys_define:nn { hyp / setup }
  {
    ,pdfhighlight .choices:nn =
      { /I, /N, /O, /P}
      {
        \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
          {
            \pdfannot_dict_put:nnn
              {link/####2}
              { H }
              { #1  }
          }
      }
    ,pdfhighlight / .code:n =
     {
        \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
          {
            \pdfannot_dict_remove:nn
              {link/##2}
              { H }
          }
      }
    ,pdfhighlight .initial:n = {/I},
    ,pdfhighlight / unknown .code:n =
      {
       \msg_warning:nnxxx { hyp } { unknown-choice+empty }
         { pdfhighlight }
         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
         { \exp_not:n {#1} }
      }
  }
```

11.6 Hiding links

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

`hidelinks` (*setup key*)
`hidelink` (*setup key*)
`hideurl` (*setup key*)
`hidefile` (*setup key*)
`hiderun` (*setup key*)
`hidemenu` (*setup key*)

```
\keys_define:nn { hyp / setup }
  {
    hidelinks .meta:n =
      {
        ,colorlinks    = false
        ,ocgcolorlinks = false
        ,pdfborder     = { 0~0~0 }
        ,pdfborderstyle=
      }
  }

\seq_map_inline:Nn \c__hyp_annot_types_seq
  {
    \keys_define:nn { hyp / setup }
      {
        hide#1 .meta:n =
          {
             ,color#1       = false
             ,ocgcolor#1    = false
             ,#1border      = { 0~0~0 }
             ,#1borderstyle =
          }
      }
  }
```

11.7 color schemes and settings

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

`colorscheme` (*setup key*)

```
\keys_define:nn { hyp / setup }
  {
    colorscheme .code:n =
      {
        \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
          {
            \keys_set:nn { hyp /setup }
              {
                 ##1 = ##2
              }
          }
      }
  }
\keys_set:nn { hyp / setup } {colorscheme=phelype}
```

12 Keys

12.1 Ignored keys

The following are ignored (with or without warnings)

unicode (*setup key*)
pdfencoding (*setup key*)
pdfversion (*setup key*)

```
\keys_define:nn { hyp / setup }
  {
    ,unicode     .code:n = {}
    ,pdfencoding .code:n = {}
    ,pdfversion .code:n =
      {
         \msg_warning:nn { hyp }{ pdfversion-disabled }
      }
   }
%
```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

verbose (*setup key*)
debug (*setup key*)
draft (*setup key*)
final (*setup key*)

```
\keys_define:nn { hyp / setup }
 {
   ,verbose .choice:
   ,verbose / true  .code:n = { \Hy@verbosetrue}
   ,verbose / false .code:n = { \Hy@verbosefalse}
   ,verbose .default:n = {true}
   ,debug .meta:n = {verbose=#1}
   ,debug .default:n = {true}
 }
\keys_define:nn { hyp / setup }
  {
   ,draft .code:n =
     {
       \Hy@drafttrue
       \PassOptionsToPackage{draft}{bookmark}
     }
   ,final .code:n =
     {
       \Hy@finaltrue
       \PassOptionsToPackage{final}{bookmark}
     }
  }
```

extension (*setup key*)
hypertexnames (*setup key*)
naturalnames (*setup key*)
pageanchor (*setup key*)
linktoc (*setup key*)
linktocpage (*setup key*)
plainpages (*setup key*)
localanchorname (*setup key*)
linkfileprefix (*setup key*)

```
\keys_define:nn { hyp / setup }
 {
   ,extension .tl_set:N = \XR@ext
   ,extension .initial:n= pdf
   ,hypertexnames .choice:
   ,hypertexnames / true  .code:n = { \Hy@hypertexnamestrue}
   ,hypertexnames / false .code:n = { \Hy@hypertexnamesfalse}
   ,hypertexnames .default:n = {true}
   ,linkfileprefix .tl_set:N  = \Hy@linkfileprefix
   ,localanchorname .choice:
   ,localanchorname / true  .code:n = { \Hy@localanchornametrue }
   ,localanchorname / false .code:n = { \Hy@localanchornamefalse }
   ,localanchorname .default:n = {true}
   ,naturalnames .choice:
   ,naturalnames / true  .code:n = { \Hy@naturalnamestrue}
   ,naturalnames / false .code:n = { \Hy@naturalnamesfalse}
   ,naturalnames .default:n = {true}
   ,pageanchor .choice:
   ,pageanchor / true  .code:n = { \Hy@pageanchortrue}
   ,pageanchor / false .code:n = { \Hy@pageanchorfalse}
   ,pageanchor .default:n = {true}
   ,plainpages .choice:
   ,plainpages / true  .code:n = { \Hy@plainpagestrue}
   ,plainpages / false .code:n = { \Hy@plainpagesfalse}
   ,plainpages .default:n = {true}
 }

\keys_define:nn { hyp / setup }
 {
   ,linktoc .choices:nn = { none, section, all, page }
     {
       \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
     }
   ,linktoc / unknown .code:n =
     {
       \msg_warning:nnxxx { hyp } { unknown-choice }
         { linktoc }
         { none, section, all, page }
         { \exp_not:n {#1} }
     }
   ,linktocpage .choice:
   ,linktocpage / true  .meta:n = {linktoc=page}
   ,linktocpage / false .meta:n = {linktoc=section}
   ,linktocpage .default:n = true
 }

```

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

url (*setup key*)
file (*setup key*)
menu (*setup key*)
run (*setup key*)

```
\prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
  {
    \keys_define:nn { hyp / setup }
     {
       ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
     }
  }
```

```
\keys_define:nn { hyp / setup }
 {
   ,baseurl       .code:n =
     {
       \__hyp_text_pdfstring:ooN  { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
        \tl_if_empty:NTF \l__hyp_tmpa_tl
          {
             \pdfmanagement_remove:nn {Catalog} { URI }
          }
          {
             \pdfmanagement_add:nnx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
          }
     }
   %only false does something ...
   ,bookmarks .choice:
   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
   ,bookmarks / true .code:n = {}
   ,bookmarks .default:n = {true}
   ,bookmarksnumbered .choice:
   ,bookmarksnumbered / false .code:n = { \Hy@bookmarksnumberedfalse }
   ,bookmarksnumbered / true .code:n  = { \Hy@bookmarksnumberedtrue }
   ,bookmarksnumbered .default:n      = {true}
   ,bookmarksopen .choice:
   ,bookmarksopen / false .code:n = { \Hy@bookmarksopenfalse }
   ,bookmarksopen / true .code:n  = { \Hy@bookmarksopentrue }
   ,bookmarksopen .default:n      = {true}
   ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
   ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
   ,pdfcenterwindow    .choice:
   ,pdfcenterwindow / false .code:n =
     {
       \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
     }
   ,pdfcenterwindow / true .code:n =
     {
       \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
     }
   ,pdfcenterwindow /  .code:n =
     {
       \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
     }
   ,pdfcenterwindow / unknown .code:n =
     {
       \msg_warning:nnxx { hyp } { no-bool }
         { pdfcenterwindow }
         { \exp_not:n {#1} }
     }
   ,pdfcenterwindow    .default:n = true
   ,pdfdirection    .choice:
   ,pdfdirection / L2R .code:n =
     {
       \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
     }
   ,pdfdirection / R2L .code:n =
     {
       \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
     }
   ,pdfdirection /  .code:n =
     {
       \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
     }
   ,pdfdirection / unknown .code:n =
     {
       \msg_warning:nnxxx { hyp } { unknown-choice+empty }
         { pdfdirection }
         { L2R , R2L }
         { \exp_not:n {#1} }
     }
   ,pdfdisplaydoctitle    .choice:
   ,pdfdisplaydoctitle / false .code:n =
     {
       \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
     }
   ,pdfdisplaydoctitle / true .code:n =
     {
       \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
     }
   ,pdfdisplaydoctitle    .default:n = true
   ,pdfduplex    .choices:nn =
     {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
     {
       \pdf_version_compare:NnTF > {1.6}
         {
           \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
             { PrintDuplex } { /#1 }
         }
         {
           \msg_warning:nnxx
             {hyp}
             {ignore-deprecated-or-unknown-option-in-pdf-version}
             {pdfduplex}
             {\pdf_version:}
         }
     }%
   ,pdfduplex /  .code:n =
     {
       \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
     }
   ,pdfduplex / unknown .code:n =
     {
       \msg_warning:nnxxx { hyp } { unknown-choice+empty }
         { pdfduplex }
         { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
         { \exp_not:n {#1} }
     }
   ,pdffitwindow    .choice:
   ,pdffitwindow / false .code:n =
     {
       \pdfmanagement_remove:nn {Catalog / ViewerPreferences }  { FitWindow }
     }
   ,pdffitwindow / true .code:n =
     {
       \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
     }
   ,pdffitwindow /  .code:n =
     {
       \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
     }
   ,pdffitwindow    .default:n = true
   ,pdffitwindow / unknown .code:n =
     {
       \msg_warning:nnxx { hyp } { no-bool }
         { pdffitwindow }
         { \exp_not:n {#1} }
     }
   ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
   ,pdflinkmargin .initial:n = {1pt}
   ,pdfmenubar    .choice:
   ,pdfmenubar / true .code:n =
     {
       \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
     }
   ,pdfmenubar / false .code:n =
     {
       \pdfmanagement_add:nn {Catalog / ViewerPreferences }
          { HideMenubar } { true }
     }
   ,pdfmenubar /  .code:n =
     {
       \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
     }
   ,pdfmenubar    .default:n = true
   ,pdfmenubar / unknown .code:n =
     {
       \msg_warning:nnxx { hyp } { no-bool }
         { pdfmenubar }
         { \exp_not:n {#1} }
     }
   ,pdfnewwindow .choice:
   ,pdfnewwindow / true .code:n =
     {
       \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
       \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
     }
   ,pdfnewwindow / false .code:n =
     {
       \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
       \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
     }
   ,pdfnewwindow /  .code:n =
     {
       \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
       \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
     }
   ,pdfnonfullscreenpagemode   .choices:nn =
     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
     {
       \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
         { NonFullScreenPageMode} {/#1}
     }
   ,pdfnonfullscreenpagemode / UseAttachments .code:n =
     {
       \pdf_version_compare:NnTF < {1.6}
         {
           %message
         }
         {
           \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
             {NonFullScreenPageMode}{/UseAttachments}
         }
     }
   ,pdfnonfullscreenpagemode /  .code:n =
     {
       \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
     }
   ,pdfnonfullscreenpagemode / unknown .code:n =
     {
       \msg_warning:nnxxx { hyp } { unknown-choice+empty }
         { pdfnonfullscreenpagemode }
         { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
         { \exp_not:n {#1} }
     }
   ,pdfnumcopies .code:n =
     {
       \pdf_version_compare:NnTF > {1.6}
         {
           \tl_if_empty:nTF {#1}
             {
               \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
             }
             {
               \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
                 {NumCopies}{#1}
             }
         }
         {
           \msg_warning:nnxx
             {hyp}
             {ignore-deprecated-or-unknown-option-in-pdf-version}
             {pdfnumcopies}
             {\pdf_version:}
         }
     }
   ,pdfpagelayout .choices:nn =
     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
     { \pdfmanagement_add:nnx {Catalog} { PageLayout }{ /#1 } }
```

```
,pdfpagelayout / .code:n =
  { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
,pdfpagelayout / unknown .code:n =
  {
    \msg_warning:nnxxx { hyp } { unknown-choice+empty }
      { pdfpagelayout }
      { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
      { \exp_not:n {#1} }
  }
,pdfpagemode   .choices:nn =
  { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
  { \pdfmanagement_add:nnx {Catalog} { PageMode }{ /#1 } }
,pdfpagemode / UseAttachments .code:n =
  {
    \pdf_version_compare:NnTF > {1.5}
      {
        \pdfmanagement_add:nnx {Catalog} { PageMode }{ /UseAttachments }
      }
      {
        \msg_warning:nnxx
          {hyp}
          {ignore-deprecated-or-unknown-value-in-pdf-version}
          {UseAttachments}
          {\pdf_version:}
      }
  }
,pdfpagemode   .initial:n  = { UseOutlines } %for now ...
,pdfpagemode / unknown .code:n =
  {
    \msg_warning:nnxxx { hyp } { unknown-choice+empty }
      { pdfpagemode }
      { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
      { \exp_not:n {#1} }
  }
,pdfpagescrop  .code:n =
  {
    \tl_if_empty:nTF {#1} %or blank?
      {
        \pdfmanagement_remove:nn {Pages} { CropBox }
      }
      {
        \pdfmanagement_add:nnx {Pages} { CropBox } { [#1] }
      }
  }
,pdfpicktraybypdfsize    .choice:
,pdfpicktraybypdfsize / true .code:n =
  {
    \pdf_version_compare:NnTF > {1.6}
      {
        \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
          { PickTrayByPDFSize } { true }
      }
      {
        \msg_warning:nnxx
```

```
            {hyp}
            {ignore-deprecated-or-unknown-option-in-pdf-version}
            {pdfpicktraybypdfsize}
            {\pdf_version:}
        }
    }
  ,pdfpicktraybypdfsize / false .code:n =
    {
      \pdf_version_compare:NnTF > {1.6}
        {
          \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
            { PickTrayByPDFSize } { false }
        }
        {
          \msg_warning:nnxx
            {hyp}
            {ignore-deprecated-or-unknown-option-in-pdf-version}
            {pdfpicktraybypdfsize}
            {\pdf_version:}
        }
    }
  ,pdfpicktraybypdfsize /  .code:n =
    {
      \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
    }
  ,pdfpicktraybypdfsize / unknown .code:n =
    {
      \msg_warning:nnxx { hyp } { no-bool }
        { picktraybypdfsize }
        { \exp_not:n {#1} }
    }
  ,pdfprintarea    .choices:nn =
    { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
    {
      \pdf_version_compare:NnTF < {2.0}
        {
          \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
            { PrintArea } { /#1 }
        }
        {
          \msg_warning:nnxx
            {hyp}
            {ignore-deprecated-or-unknown-option-in-pdf-version}
            {pdfprintarea}
            {\pdf_version:}
        }
    }%
  ,pdfprintarea /  .code:n =
    { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
  ,pdfprintarea / unknown .code:n =
    {
      \msg_warning:nnxxx { hyp } { unknown-choice+empty }
        { pdfprintarea }
        { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
```

```
          { \exp_not:n {#1} }
      }
    ,pdfprintclip    .choices:nn =
      { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
      {
        \pdf_version_compare:NnTF < {2.0}
          {
            \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
              { PrintClip } { /#1 }
          }
          {
            \msg_warning:nnxx
              {hyp}
              {ignore-deprecated-or-unknown-option-in-pdf-version}
              {pdfprintclip}
              {\pdf_version:}
          }
      }%
    ,pdfprintclip /  .code:n =
      {
        \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
      }
    ,pdfprintclip / unknown .code:n =
      {
        \msg_warning:nnxxx { hyp } { unknown-choice+empty }
          { pdfprintclip }
          { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
          { \exp_not:n {#1} }
      }
    ,pdfprintpagerange .code:n =
      {
        \pdf_version_compare:NnTF > {1.6}
          {
            \tl_if_empty:nTF { #1}
              {
                \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
                  { PrintPageRange }
              }
              {
               \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
                  {PrintPageRange}{[#1]}
              }
          }
          {
             \msg_warning:nnxx
               {hyp}
               {ignore-deprecated-or-unknown-option-in-pdf-version}
               {pdfprintpagerange}
               {\pdf_version:}
          }
      }
    ,pdfprintscaling    .choices:nn =
      { None, AppDefault }
      {
```

```
      \pdf_version_compare:NnTF > {1.5}
        {
          \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
            { PrintScaling } { /#1 }
        }
        {
          \msg_warning:nnxx
            {hyp}
            {ignore-deprecated-or-unknown-option-in-pdf-version}
            {pdfprintscaling}
            {\pdf_version:}
        }
    }%
  ,pdfprintscaling /  .code:n =
    {
      \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
    }
  ,pdfprintscaling / unknown .code:n =
    {
      \msg_warning:nnxxx { hyp } { unknown-choice+empty }
        { pdfprintarea }
        { None, AppDefault }
        { \exp_not:n {#1} }
    }
  ,pdfremotestartview .code:n =
    {
      \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
      \exp_args:NNV
      \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl  \l__hyp_tmpa_
        {
          \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
        }
        {
          \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
          \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
        }
    }
  ,pdfremotestartview .initial:n = {Fit}
  % pdfstartpage is special as it shares code with pdfstartview
  ,pdfstartpage .code:n =
    {
      \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
      \bool_if:nTF
        { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
        {
          \pdfmanagement_remove:nn {Catalog} { OpenAction }
        }
        {
          \pdfmanagement_add:nnx {Catalog} { OpenAction }
            {
              [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
            }
        }
    }
  ,pdfstartpage .initial:n =1
  ,pdfstartview .code:n =
    {
      \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
      \exp_args:NNV
      \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl  \l__hyp_tmpa_
        {
          \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
        }
        {
          \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
          \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
        }
      \bool_if:nTF
        { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
        {
          \pdfmanagement_remove:nn {Catalog} { OpenAction }
        }
        {
          \pdfmanagement_add:nnx {Catalog} { OpenAction }
            {
              [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
            }
        }
    }
  ,pdfstartview .initial:n = Fit
  ,pdftoolbar    .choice:
  ,pdftoolbar / true .code:n =
    {
      \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
    }
  ,pdftoolbar / false .code:n =
    {
      \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
        { HideToolbar } { true }
    }
  ,pdftoolbar / true .code:n =
    {
      \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
    }
  ,pdftoolbar    .default:n = true
  ,pdftoolbar / unknown .code:n =
    {
      \msg_warning:nnxx { hyp } { no-bool }
        { pdftoolbar }
        { \exp_not:n {#1} }
    }
  % pdfview see below.
  ,pdfviewarea    .choices:nn =
    { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
    {
      \pdf_version_compare:NnTF < {2.0}
        {
          \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
          { ViewArea } { /#1 }
        }
        {
          \msg_warning:nnxx
            {hyp}
            {ignore-deprecated-or-unknown-option-in-pdf-version}
            {pdfviewarea}
            {\pdf_version:}
        }
  }%
,pdfviewarea /  .code:n =
  {
    \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
  }
,pdfviewarea / unknown .code:n =
  {
    \msg_warning:nnxxx { hyp } { unknown-choice+empty }
      { pdfviewarea }
      { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
      { \exp_not:n {#1} }
  }
,pdfviewclip    .choices:nn =
  { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
  {
    \pdf_version_compare:NnTF < {2.0}
      {
        \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
          { ViewClip } { /#1 }
      }
      {
        \msg_warning:nnxx
          {hyp}
          {ignore-deprecated-or-unknown-option-in-pdf-version}
          {pdfviewclip}
          {\pdf_version:}
      }
  }%
,pdfviewclip /  .code:n =
  {
    \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
  }
,pdfviewclip / unknown .code:n =
  {
    \msg_warning:nnxxx { hyp } { unknown-choice+empty }
      { pdfviewclip }
      { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
      { \exp_not:n {#1} }
  }
,pdfwindowui    .choice:
,pdfwindowui / true .code:n =
  {
    \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
  }
,pdfwindowui / false .code:n =
     {
       \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
          { HideWindowUI } { true }
     }
   ,pdfwindowui /  .code:n =
     {
       \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
     }
   ,pdfwindowui / unknown .code:n =
     {
       \msg_warning:nnxx { hyp } { no-bool }
         { pdfwindowui }
         { \exp_not:n {#1} }
     }
   ,pdfwindowui    .default:n = true
 }
```

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

```
\keys_define:nn { hyp / setup }
  {
    ,pdfview .code:n =
      {
        \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
        \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
           {
             { xyz }
               {
                 \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
                   {
                     \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
                     \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
                       {
                         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
                       }
                       {
                         \tl_set:Nx \l__hyp_dest_pdfview_tl
                            {
                               \fp_eval:n { \l__hyp_tmpa_tl * 100 }
                            }
                       }
                   }
                   {
                     \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
                   }
               }
             { fit }   { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}  }
             { fitb }  { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
             { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh}}
             { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv}}
             { fith }  { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
             { fitv }  { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
             { fitr }
               {
                 \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
                   {
                     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
                   }
                   {
                      %ensure 4 values ...
                     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
                     \seq_put_right:Nn \l__hyp_tmpa_seq  {0}
                     \seq_put_right:Nn \l__hyp_tmpa_seq  {0}
                     \seq_put_right:Nn \l__hyp_tmpa_seq  {0}
                     \hbox_set_to_wd:Nnn \l__hyp_dest_box
                       {
                         \fp_eval:n
                          {
                            round
                             (
                               abs
                                (
                                  \seq_item:Nn\l__hyp_tmpa_seq{4}
                                  -
                                  (\seq_item:Nn\l__hyp_tmpa_seq{2})
                                 ),
                                3
                             )
                          }bp
                       }{}
                     \box_set_dp:Nn \l__hyp_dest_box
                       {
                         \fp_eval:n
                           {
                             round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
                           }bp
                       }
                     \box_set_ht:Nn \l__hyp_dest_box
                       {
                         \seq_item:Nn\l__hyp_tmpa_seq{5}bp
                       }
                   }
               }
           }
           {
              \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
              \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
           }
      }
    ,pdfview .initial:n  = {xyz}
  }
```

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.

```
\keys_define:nn { hyp / setup }
```

```
  {
   ,pdflang       .code:n =
          {
            \tl_if_empty:nTF { #1 }
              {
                \pdfmanagement_remove:nn {Catalog} { Lang }
              }
              {
                \pdfmanagement_add:nnx {Catalog} { Lang } { (#1) }
              }
            \__hyp_store_metadata:nn {pdflang}{#1}
          }
   }
```

12.3.1 “info keys”

pdfauthor (*setup key*)
pdftitle (*setup key*)
pdfcreator (*setup key*)
pdfsubject (*setup key*)
pdfproducer (*setup key*)
pdfkeywords (*setup key*)

The keys store their value also in the metadate container, so that hyperxmp can use them.

```
\cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
  {
    \keys_define:nn { hyp / setup }
      {
        pdf#1  .code:n =
          {
            \tl_if_blank:nTF {##1}
              {
                \pdfmanagement_remove:nn {Info}{#2}
              }
              {
                \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
                \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
                  {
                    \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
                  }
              }
             \__hyp_store_metadata:nn {pdf#1}{##1}
          }
      }
    \keys_define:nn { hyp / info }
      {
        #2  .code:n =
          {
            \tl_if_blank:nTF {##1}
              {
                \pdfmanagement_remove:nn {Info}{#2}
              }
              {
                \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
                \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
                  {
                    \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
                  }
              }
            \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
          }
        ,unknown .code:n =
         {
            \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
            \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
              {
                \exp_args:Nno
                \pdfmanagement_add:nnx {Info}
                    { \l_keys_key_str } {\l__hyp_tmpa_str}
              }
         }
      }
  }
\__hyp_setup_info_key:nn {author}   {Author}
\__hyp_setup_info_key:nn {title}    {Title}
\__hyp_setup_info_key:nn {producer} {Producer}
\__hyp_setup_info_key:nn {creator}  {Creator}
% ignored key: addtopdfcreator
\__hyp_setup_info_key:nn {subject}  {Subject}
\__hyp_setup_info_key:nn {keywords}  {Keywords}
```

`pdfcreationdate` (*setup key*)
`pdfmoddate` (*setup key*)

These keys are not really needed. We store them too in the container. CreationDate and ModDate should not use the hex encoding.

```
\cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
  {
    \keys_define:nn { hyp / setup }
      {
        pdf#1  .code:n =
          {
            \tl_if_blank:nTF {##1}
              {
                \pdfmanagement_remove:nn {Info}{#2}
              }
              {
                \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
              }
             \__hyp_store_metadata:nn {pdf#1}{##1}
          }
      }
    \keys_define:nn { hyp / info }
      {
        #2  .code:n =
          {
            \tl_if_blank:nTF {##1}
              {
                \pdfmanagement_remove:nn {Info}{#2}
              }
              {
                \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
              }
            \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
          }
      }
  }
```

```

\__hyp_setup_info_date_key:nn {creationdate}  {CreationDate}
\__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 ...

```
\keys_define:nn { hyp / setup }
  {
    ,pdftrapped .code:n =
      {
        \exp_args:Nne
          \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
      }
    ,_pdftrapped  .choices:nn = {TRUE,FALSE,UNKNOWN}
       {
         \pdfmanagement_add:nnx {Info}{Trapped}
           {/
             \str_uppercase:f { \str_head:n { #1 } }
             \str_lowercase:f { \str_tail:n { #1 } }
           }
         \__hyp_store_metadata:nx {pdftrapped}
           {
             \str_uppercase:f { \str_head:n { #1 } }
             \str_lowercase:f { \str_tail:n { #1 } }
           }
       }
    ,_pdftrapped / unknown .code:n =
       {
         \msg_warning:nnxxx { hyp } { unknown-choice }
           { pdftrapped }
           { true~(case~insensitive), false~(case~insensitive), unknown~(case~insensitive) }
           { \exp_not:n {#1} }
       }
  }
```

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

```
\keys_define:nn { hyp / setup }
  {
    pdfinfo .code:n =
      {
        \keys_set:nn { hyp / info } { #1 }
      }
  }
```

Now we set some default values

```
\keys_set:nn { hyp / setup} {pdfcreator = LaTeX~with~hyperref}
\keys_set:nn { hyp / setup} {pdfauthor  = }
\keys_set:nn { hyp / setup} {pdftitle   = }
\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:

```
\clist_map_inline:nn
  {
   ,pdfcopyright
   ,pdftype
   ,pdflicenseurl
   ,pdfauthortitle
   ,pdfcaptionwriter
   ,pdfmetalang
   ,pdfapart
   ,pdfaconformance
   ,pdfuapart
   ,pdfxstandard
   ,pdfsource
   ,pdfdocumentid
   ,pdfinstanceid
   ,pdfversionid
   ,pdfrendition
   ,pdfpublication
   ,pdfpubtype
   ,pdfbytes
   ,pdfnumpages
   ,pdfissn
   ,pdfeissn
   ,pdfisbn
   ,pdfbookedition
   ,pdfpublisher
   ,pdfvolumenum
   ,pdfissuenum
   ,pdfpagerange
   ,pdfdoi
   ,pdfurl
   ,pdfidentifier
   ,pdfsubtitle
   ,pdfpubstatus
   ,pdfcontactaddress
   ,pdfcontactcity
   ,pdfcontactregion
   ,pdfcontactpostcode
   ,pdfcontactcountry
   ,pdfcontactphone
   ,pdfcontactemail
   ,pdfcontacturl
   ,pdfdate
  }
  {
    \keys_define:nn { hyp / setup }
      {
        #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
      }
  }

\str_if_eq:eeT {A}{\pdfmeta_standard_item:n{type}}
  {
    \exp_args:Nnx
```

```
      \keys_set:nn {hyp / setup}
        {
          ,pdfapart        = \pdfmeta_standard_item:n {level}
          ,pdfaconformance = \pdfmeta_standard_item:n {conformance}
        }
  }
```

12.5 Transitions

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

```
\keys_define:nn { hyp / setup }
  {
    pdfpageduration .code:n =
     {
        \tl_if_blank:nTF { #1 }
          {
            \pdfmanagement_remove:nn {Page}{Dur}
          }
          {
            \pdfmanagement_add:nnn {Page}{Dur}{#1}
          }
     }
  }
```

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

```
\keys_define:nn { hyp / setup }
  {
    pdfpagetransition .code:n =
     {
       \tl_if_blank:nTF {#1}
         {
           \pdfmanagement_remove:nn {Page}{Trans}
         }
         {
           \group_begin:
           \keys_set:nn { hyp / trans }{style=R,#1}
           \pdf_object_unnamed_write:nx { dict }
             {
               \pdfdict_use:n {l__hyp_page/Trans}
             }
           \pdfmanagement_add:nnx {Page}{Trans}{\pdf_object_ref_last:}
           \group_end:
         }
     }
  }
\keys_define:nn { hyp / trans }
  {
    ,style .choices:nn =
      {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
      { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
    ,style / unknown .code:n =
      {
        \msg_warning:nnxxx { hyp } { unknown-choice }
           { trans / style }
           { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
           { \exp_not:n {#1} }
      }
    ,duration .code:n  =
      {
        \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{#1}
      }
    ,direction .choices:nn =
      {H,V}
      { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
    ,direction .choices:nn =
      {0,90,180,270,315}
      { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ #1 } }
    ,direction / None .code:n =
      { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ /None } }
    ,direction / unknown .code:n =
      {
        \msg_warning:nnxxx { hyp } { unknown-choice }
           { trans / 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)
           }
           { \exp_not:n {#1} }
      }
    ,motion .choices:nn =
     {I,O}
     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
    ,motion / unknown .code:n =
      {
        \msg_warning:nnxxx { hyp } { unknown-choice }
           { trans / motion }
           { I~(inwards) , O~(outwards) }
           { \exp_not:n {#1} }
      }
     ,scale .code:n =
       { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
     ,opaque .choices:nn = {true,false}
       { \pdfdict_put:nnn { l__hyp_page/Trans }{ B } { #1} }
     ,opaque / unknown .code:n =
       {
          \msg_warning:nnxxx { hyp } { unknown-choice }
           { trans / B }
           { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
           { \exp_not:n {#1} }
       }
     % try to set unknown keys as style
     ,unknown .code:n =
       {
         % warning ...
         \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
       }
  }
```

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

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

Unfinished Form field code

```
\NewDocumentCommand \MakeFieldObject { m m }
  {
    \pdfxform_new:nnn { #2 }{} { #1 }
  }


\prop_new:N   \g__hyp_AcroForm_CoFields_prop
\prop_new:N   \g__hyp_AcroForm_Fields_prop

\let\HyField@afields\ltx@empty
\let\HyField@cofields\ltx@empty
%% UF test for old pdftex removed
%\let\HyField@AuxAddToFields\ltx@gobble
%\let\HyField@AuxAddToCoFields\ltx@gobbletwo
\def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%

% the value doesn't matter, but with a prop we avoid duplicates and it is
% clearly faster than removing them from a sequence
```

```
\def\HyField@AuxAddToFields#1
  {
    \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
  }%

%fields with empty key get a value too -- lets hope that
%this give the expected behaviour
\def\HyField@AuxAddToCoFields #1 #2
  {
    \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
  }

\Hy@AtBeginDocument
  {
    \if@filesw
      \immediate\write\@mainaux{%
        \string\providecommand\string\HyField@AuxAddToFields[1]{}%
       }%
      \immediate\write\@mainaux{%
        \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
      }%
    \fi
    \let\HyField@AfterAuxOpen\@firstofone
  }%

\def\HyField@AddToFields
  {
    \exp_args:Nx\HyField__hypAddToFields
      {
        \pdfannot_box_ref_last:
      }
    \ifx\Fld@calculate@code\ltx@empty
    \else
      \begingroup
        \Hy@safe@activestrue
        \edef\Hy@temp{%
          \endgroup
          \if@filesw
            \write\@mainaux
             {
              \string\HyField@AuxAddToCoFields
               {
                \Fld@calculate@sortkey
               }
               {
                 \pdfannot_box_ref_last:
               }
            }
          \fi
        }%
      \Hy@temp
    \fi
  }%
```

```
\def\HyField__hypAddToFields#1{
  \HyField@AfterAuxOpen{%
    \if@filesw
      \write\@mainaux{%
        \string\HyField@AuxAddToFields{#1}%
        }%
      \fi
    }%
  }%

\ExplSyntaxOff
\ExplSyntaxOn
\tl_new:N  \l__hyp_CheckmarkYes_tl
\tl_set:Nn \l__hyp_CheckmarkYes_tl {  __hyp_xform_CheckMarkYes }
\tl_new:N  \l__hyp_CheckmarkOff_tl
\tl_set:Nn \l__hyp_CheckmarkOff_tl {  __hyp_xform_CheckMarkOff }

\def\@Form[#1]
  {
    \@ifundefined{textcolor}{\let\textcolor\@gobble}{}
    \kvsetkeys{Form}{#1}
    \pdf@ifdraftmode{}
      {
        \Hy@FormObjects
        \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
          {
            \pdfmanagement_add:nnx { Catalog / AcroForm } { Fields }{##1}
            %\pdfmanagement_show:n { Catalog / AcroForm }
          }
        \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
          {
            \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
              {
                \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
              }
            \seq_sort:Nn \l__hyp_tmpa_seq
              {
                \int_compare:nNnTF { \pdf@strcmp{##1}{##2} } > { 0 }
                %\str_compare:nNnTF {##1} > {##2}
                  { \sort_return_swapped: }
                  { \sort_return_same: }
              }
            \seq_map_inline:Nn \l__hyp_tmpa_seq
             {
                \pdfmanagement_add:nnx { Catalog / AcroForm }
                  { CO }
                  {
                    \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
                  }
             }
          }
       \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
         {ZaDb} {\pdf_object_ref:n {__hyp/Font/ZaDb} }
       \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
```

```
     {Helv} {\pdf_object_ref:n {__hyp/Font/Helv} }
   \pdfmanagement_add:nnx {Catalog /AcroForm}
     {DA}{(/Helv~10~Tf~0~g)}
   \pdfmeta_standard_verify:nTF {form_no_NeedAppearance}
     {
       \legacy_if:nT { HyField@NeedAppearances }
         {
           \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
         }
     }
     {
       \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
     }
   }
 \MakeFieldObject
   {
     \group_begin:
     \fontfamily{pzd}
     \fontencoding{U}
     \fontseries{m}
     \fontshape{n}
     \selectfont
     \char123
     \group_end:
   }
   {__hyp_xform_Ding}
\MakeFieldObject
   {
     \group_begin:
     \fontfamily{pzd}
     \fontencoding{U}
     \fontseries{m}
     \fontshape{n}
     \selectfont
     \phantom{\char123}
     \group_end:
   }
   {__hyp_xform_DingOff}
 \MakeFieldObject
   {
     \group_begin:
     \fontfamily{pzd}
     \fontencoding{U}
     \fontseries{m}
     \fontshape{n}
     \selectfont
     \char51
     \group_end:
   }
   {__hyp_xform_CheckMarkYes}
 \MakeFieldObject
   {
    \group_begin:
     \fontfamily{pzd}
```

```
         \fontencoding{U}
         \fontseries{m}
         \fontshape{n}
         \selectfont
         \phantom{\char51} %perhaps xetex needs some small glyph ..
         \group_end:
       }
       {__hyp_xform_CheckMarkOff}
     \MakeFieldObject
       {
         \fbox{\textcolor{yellow}{\textsf{Submit}}} %color?
       }
       {__hyp_xform_Submit}
     \MakeFieldObject
       {
         \fbox{\textcolor{yellow}{\textsf{SubmitP}}} %color?
       }
       {__hyp_xform_SubmitP}
  }
\ExplSyntaxOff
\let\@endForm\ltx@empty
\let\HyAnn@AbsPageLabel\ltx@empty
\let\Fld@pageobjref\ltx@empty

\ExplSyntaxOn
\newcount\HyAnn@Count
\HyAnn@Count=\ltx@zero
\def\HyAnn@AbsPageLabel
  {
    \global\advance\HyAnn@Count by\ltx@one
    %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
    %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
    %\zref@refused{HyAnn@\the\HyAnn@Count}%
    \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
    \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
  }%
\def\Fld@pageobjref
  {
    \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
      {
        /P~\pdf_pageobject_ref:n
          {
            \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
          }
      }
  }
\ExplSyntaxOff
\ExplSyntaxOn
%% check if the attr should be set through
%% hooks.
%% check if options are missing.
\def\@TextField[#1]#2{% parameters, label
  \def\Fld@name{#2}%
  \let\Fld@default\ltx@empty
  \let\Fld@value\@empty
  \def\Fld@width{\DefaultWidthofText}%
  \def\Fld@height{%
    \ifFld@multiline
      \DefaultHeightofTextMultiline
    \else
      \DefaultHeightofText
    \fi
  }%
  \begingroup
    \expandafter\HyField@SetKeys\expandafter{%
      \DefaultOptionsofText,#1%
    }%
    \PDFForm@Name
    \HyField@FlagsText
    \ifFld@hidden\def\Fld@width{1sp}\fi
    \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
    \LayoutTextField{#2}{%
      \leavevmode
      \HyAnn@AbsPageLabel
      \Hy@escapeform\PDFForm@Text
      \pdfannot_box:nnnn
          {\Fld@width}
          {\Fld@height}
          {0pt} %is this correct?
          {\PDFForm@Text}
      \MakeTextField{\Fld@width}{\Fld@height}
      \HyField@AddToFields
    }%
  \endgroup
}
\providecommand\@curropt{}
\def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
  \def\Fld@name{#2}
  \let\Fld@default\relax
  \let\Fld@value\relax
  \def\Fld@width{\DefaultWidthofChoiceMenu}
  \def\Fld@height{\DefaultHeightofChoiceMenu}
  \begingroup
    \Fld@menulength=0 %
    \@tempdima\z@
    \clist_map_variable:nNn { #3 } \@curropt
    %\@for\@curropt:=#3\do
      {%
        \expandafter\Fld@checkequals\@curropt==\\%
        \Hy@StepCount\Fld@menulength
        \settowidth{\@tempdimb}{\@currDisplay}%
        \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
      }%
    \advance\@tempdima by~15\p@
    \begingroup
      \HyField@SetKeys{#1}
    \edef\x{\endgroup
      \noexpand\expandafter
      \noexpand\HyField@SetKeys
      \noexpand\expandafter{%
        \expandafter\noexpand\csname DefaultOptionsof%
          \ifFld@radio
            Radio%
          \else
            \ifFld@combo
              \ifFld@popdown
                PopdownBox%
              \else
                ComboBox%
              \fi
            \else
              ListBox%
            \fi
          \fi
        \endcsname
      }%
    }\x
    \HyField@SetKeys{#1}%
    \PDFForm@Name
    \ifFld@hidden\def\Fld@width{1sp}\fi
    \ifx\Fld@value\relax
      \let\Fld@value\Fld@default
    \fi
    \LayoutChoiceField{#2}{%
      \ifFld@radio
        \HyField@FlagsRadioButton
        \__hypRadio{#3}%
      \else
        \begingroup
          \HyField@FlagsChoice
          \ifdim\Fld@width<\@tempdima
            \ifdim\@tempdima<1cm\@tempdima1cm\fi
            \edef\Fld@width{\the\@tempdima}%
          \fi
          \ifFld@combo
          \else
            \@tempdima=\the\Fld@menulength\Fld@charsize
            \advance\@tempdima by~\Fld@borderwidth bp %
            \advance\@tempdima by~\Fld@borderwidth bp %
            \edef\Fld@height{\the\@tempdima}%
          \fi
          \__hypListbox{#3}%
        \endgroup
      \fi
    }%
  \endgroup
}
\tl_new:N  \l__hyp_RadioYes_tl
\tl_set:Nn \l__hyp_RadioYes_tl { __hyp_xform_Ding }
\def\__hypRadio#1{%
  \Fld@listcount=0~%
  %\show\Fld@default
  \EdefEscapeName\Fld@default{\Fld@default}%
  \clist_map_variable:nNn { #1 } \@curropt
  %\@for\@curropt:=#1\do
   {%
    \expandafter\Fld@checkequals\@curropt==\\%
    \EdefEscapeName\@currValue{\@currValue}%
    \Hy@StepCount\Fld@listcount
    \@currDisplay\space
    \leavevmode
    \HyAnn@AbsPageLabel
    \Hy@escapeform\PDFForm@Radio
    \pdfannot_box:nnnn
      {\Fld@width}
      {\Fld@height}
      {0pt} %is this correct?
      {
        \PDFForm@Radio
        /AP
         <<
         /N
          <<
          /\@currValue\c_space_tl \pdfxform_ref:o {__hyp_xform_Ding}
          %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
          >>
         >>
      }
      {\fbox{  \MakeRadioField{\Fld@width}{\Fld@height}} }
     \int_compare:nNnT { \Fld@listcount} = { 1 }
      { \HyField@AddToFields }
    \c_space_tl % deliberate space between radio buttons
                % to do: --> should be configurable
  }%
}

\newcount\Fld@listcount
\def\__hypListbox#1
 {
  \HyField@PDFChoices{#1}
  \mode_leave_vertical:
  \HyAnn@AbsPageLabel
  \Hy@escapeform\PDFForm@List
  \pdf_link_user:nnn
       {widget} %perhaps we need more types??
       {\PDFForm@List}
       {\MakeChoiceField{\Fld@width}{\Fld@height}}
  \HyField@AddToFields
 }


\def\@PushButton[#1]#2{% parameters, label
  \def\Fld@name{#2}%
  \group_begin:
    \exp_args:No\HyField@SetKeys
     {
      \DefaultOptionsofPushButton,#1
     }
    \PDFForm@Name
    \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
      {
        \HyField@FlagsPushButton
        \legacy_if:nT {Fld@hidden}
          {
            \def\Fld@width{1sp}
          }
        \LayoutPushButtonField
          {
            \mode_leave_vertical:
            \HyAnn@AbsPageLabel
            \Hy@escapeform\PDFForm@Push
            \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
            \pdfannot_box:nnnn
              {\box_wd:N\l_tmpa_box}
              {\box_ht:N\l_tmpa_box}
              {\box_dp:N\l_tmpa_box} %is this correct?
              {\PDFForm@Push}
             {\box_use:N\l_tmpa_box}
            \HyField@AddToFields
          }
      }
      {
        \msg_error:nn { hyp }{ pdfa-no-push-button }
        \LayoutPushButtonField
          {
            \mode_leave_vertical:
            \MakeButtonField{#2}
          }
      }
   \group_end:
}

\def\@Submit[#1]#2
 {
   \def\Fld@width {\DefaultWidthofSubmit}
   \def\Fld@height{\DefaultHeightofSubmit}
   \group_begin:
     \exp_args:No\HyField@SetKeys
       {
         \DefaultOptionsofSubmit,#1
       }
   \HyField@FlagsPushButton
   \HyField@FlagsSubmit
   \legacy_if:nT { Fld@hidden }
     {
       \def\Fld@width{1sp}
     }
    \mode_leave_vertical:
    \HyAnn@AbsPageLabel
    \Hy@escapeform\PDFForm@Submit
    \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
    \pdfannot_box:nnnn
      {\box_wd:N\l_tmpa_box}
      {\box_ht:N\l_tmpa_box}
      {\box_dp:N\l_tmpa_box} %is this correct?
      {
        \PDFForm@Submit
        /AP<<
          /N~\pdfxform_ref:n {__hyp_xform_Submit}~
          /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
         >>
       }
      \HyField@AddToFields
      \box_use:N\l_tmpa_box

    \group_end:
 }

\def\@Reset[#1]#2
 {
   \def\Fld@width {\DefaultWidthofReset}
   \def\Fld@height{\DefaultHeightofReset}
   \group_begin:
     \exp_args:No\HyField@SetKeys
       {
         \DefaultOptionsofReset,#1
       }
     \mode_leave_vertical:
     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
       {
         \HyField@FlagsPushButton
         \legacy_if:nT { Fld@hidden }
           { \def\Fld@width{1sp} }
         \HyAnn@AbsPageLabel
         \Hy@escapeform\PDFForm@Reset
         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
         \pdfannot_box:nnnn
           {\box_wd:N\l_tmpa_box}
           {\box_ht:N\l_tmpa_box}
           {\box_dp:N\l_tmpa_box} %is this correct?
           { \PDFForm@Reset }
         \HyField@AddToFields
         \box_use:N \l_tmpa_box
       }
       {
         \msg_error:nn { hyp }{ pdfa-no-reset-button }
         \MakeButtonField{#2}
       }
    \group_end:
  }

\def\@CheckBox[#1]#2
  {% parameters, label
    \def\Fld@name{#2}
    \def\Fld@default{0}
    \group_begin:
      \def\Fld@width {\DefaultWidthofCheckBox}
      \def\Fld@height{\DefaultHeightofCheckBox}
      \exp_args:No\HyField@SetKeys
        {
          \DefaultOptionsofCheckBox,#1
        }
      \PDFForm@Name
      \HyField@FlagsCheckBox
      \legacy_if:nT { Fld@hidden }
        {
          \def\Fld@width{1sp}
        }
      \LayoutCheckField{#2}
        {
          \mode_leave_vertical:
          \HyAnn@AbsPageLabel
          \Hy@escapeform\PDFForm@Check
          \pdfannot_box:nnnn
            {\Fld@width}
            {\Fld@height}
            {0pt} %is this correct?
            {\PDFForm@Check}
          \HyField@AddToFields %check if this works with xelatex ...
        }
    \group_end:
  }
\ExplSyntaxOff

%hm. Should a luatex driver use type1 fonts in fields????
\ExplSyntaxOn
\def\Hy@FormObjects
  {
    \pdf_object_new:nn   {__hyp/Encoding/pdfdoc }   { dict }
    \pdf_object_new:nn   {__hyp/Font/ZaDb } { dict }
    \pdf_object_new:nn   {__hyp/Font/Helv }    { dict }
    \pdf_object_write:nx {__hyp/Encoding/pdfdoc }
      {
        /Type/Encoding
        /Differences[
          24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
          /ring/tilde
          \c_space_tl
          39/quotesingle
          \c_space_tl
          96/grave %
          \iow_newline:
          128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
          /fraction/guilsinglleft/guilsinglright/minus/perthousand
          /quotedblbase/quotedblleft/quotedblright/quoteleft
          /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
          /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
          \iow_newline:
```

```
          164/currency
          \c_space_tl
          166/brokenbar
          \c_space_tl
          168/dieresis/copyright/ordfeminine
          \c_space_tl
          172/logicalnot/.notdef/registered/macron/degree/plusminus
          /twosuperior/threesuperior/acute/mu
          \c_space_tl
          183/periodcentered/cedilla/onesuperior/ordmasculine
          \c_space_tl
          188/onequarter/onehalf/threequarters
          \iow_newline:
          192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
          /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
          /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
          /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
          /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
          /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
          /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
          /iacute/icircumflex/idieresis/eth/ntilde/ograve/oacute
          /ocircumflex/otilde/odieresis/divide/oslash/ugrave
          /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
         ]
      }
    \pdf_object_write:nn {__hyp/Font/ZaDb }
      {
        /Type/Font
        /Subtype/Type1
        /Name/ZaDb
        /BaseFont/ZapfDingbats
      }
    \pdf_object_write:nx {__hyp/Font/Helv }
      {
        /Type/Font
        /Subtype/Type1
        /Name/Helv
        /BaseFont/Helvetica
        /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
      }
  \global\let\Hy@FormObjects\relax
  }
\ExplSyntaxOff
\providecommand*{\Fld@pageobjref}{}
\ifcsname pdf@escapestring\endcsname
  \def\Hy@escapeform#1{%
    \ifHy@pdfescapeform
      \let\Hy@escapestring\pdfescapestring
    \else
      \let\Hy@escapestring\@firstofone
    \fi
  }%
  \Hy@escapeform{}%
\else
```

```
  \let\Hy@escapestring\@firstofone
  \def\Hy@escapeform#1{%
    \ifHy@pdfescapeform
      \def\Hy@escapestring##1{%
        \noexpand\Hy@escapestring{\noexpand##1}%
      }%
      \edef\Hy@temp{#1}%
      \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{}\@nil
      \def\Hy@escapestring##1{%
        \@ifundefined{Hy@esc@\string##1}{%
          ##1%
          \ThisShouldNotHappen
        }{%
          \csname Hy@esc@\string##1\endcsname
        }%
      }%
    \else
      \let\Hy@escapestring\@firstofone
    \fi
  }%
  \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
    \ifx\\#3\\%
    \else
      \expandafter
      \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
      \ltx@ReturnAfterFi{%
        \Hy__hypescapeform#3\@nil
      }%
    \fi
  }%
\fi
\def\PDFForm@Name{%
  \PDFForm__hypName\Fld@name
  \ifx\Fld@altname\relax
  \else
    \PDFForm__hypName\Fld@altname
  \fi
  \ifx\Fld@mappingname\relax
  \else
    \PDFForm__hypName\Fld@mappingname
  \fi
}
\def\PDFForm__hypName#1{%
  \begingroup
    \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
      \ifHy@unicode
        \Hy@unicodefalse
      \fi
    \fi
    \pdfstringdef\Hy@gtemp#1%
  \endgroup
  \let#1\Hy@gtemp
}
\def\Fld@X@additionalactions{%
```

```
  \ifx\Fld@keystroke@code\@empty
  \else
    /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
  \fi
  \ifx\Fld@format@code\@empty
  \else
    /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
  \fi
  \ifx\Fld@validate@code\@empty
  \else
    /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
  \fi
  \ifx\Fld@calculate@code\@empty
  \else
    /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
  \fi
  \ifx\Fld@onfocus@code\@empty
  \else
    /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
  \fi
  \ifx\Fld@onblur@code\@empty
  \else
    /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
  \fi
  \ifx\Fld@onmousedown@code\@empty
  \else
    /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
  \fi
  \ifx\Fld@onmouseup@code\@empty
  \else
    /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
  \fi
  \ifx\Fld@onenter@code\@empty
  \else
    /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
  \fi
  \ifx\Fld@onexit@code\@empty
  \else
    /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
  \fi
}
\ExplSyntaxOn
\def\Fld@additionalactions
 {%
  \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{}
    {
      \pdfmeta_standard_verify:nT {annot_widget_no_AA}
        {/AA<<\Fld@X@additionalactions>>}
    }
 }
\ExplSyntaxOff
\def\Fld@annotnames{%
  /T(\Fld@name)%
  \ifx\Fld@altname\relax
  \else
    /TU(\Fld@altname)%
  \fi
  \ifx\Fld@mappingname\relax
  \else
    /TM(\Fld@mappingname)%
  \fi
}
\ExplSyntaxOn
\def\PDFForm@Check
  {
    /Subtype/Widget
    ~\Fld@annotflags
    ~\Fld@pageobjref
    ~\Fld@annotnames
    /FT/Btn
   \Fld@flags
   /Q~\Fld@align
   /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
   /AP
     <<
      /N
        <<
          /Yes~\pdfxform_ref:o{\l__hyp_CheckmarkYes_tl}
          /Off~\pdfxform_ref:o{\l__hyp_CheckmarkOff_tl}
        >>
    >>
  /MK<<
    \int_compare:nNnF {\Fld@rotation}={0}
      {
        /R~\Fld@rotation
      }
    \tl_if_empty:NF\Fld@bordercolor
      {
        /BC[\Fld@bordercolor]
      }
    \tl_if_empty:NF\Fld@bcolor
      {
        /BG[\Fld@bcolor]
      }
    /CA(\Hy@escapestring{\Fld@cbsymbol})%
  >>
  /DA
    (
      /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
      \tl_if_empty:NF \Fld@color
        {
          \c_space_tl \Fld@color
        }
    )
  /H/P
  \legacy_if:nTF {Fld@checked}
    {
      /V/Yes /AS/Yes
    }
    {
      /V/Off /AS/Off
    }
  \Fld@additionalactions
}
\ExplSyntaxOff
\ExplSyntaxOn
   \def\PDFForm@Push
     {
       /Subtype/Widget
       ~\Fld@annotflags
       ~\Fld@pageobjref
       ~\Fld@annotnames
       /FT/Btn
       ~\Fld@flags
       /H/P
       /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
       \bool_if:nT
         {
           !\int_compare_p:nNn {\Fld@rotation} = {0}
          ||
           \tl_if_exist_p:N \Fld@bordercolor
         }
         {
           /MK
             <<
               \int_compare:nNnF {\Fld@rotation} = {0}
                 {
                   /R~\Fld@rotation
                 }
               \tl_if_exist:NT \Fld@bordercolor
                 {
                   /BC[\Fld@bordercolor]
                 }
             >>
          }
       /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
       \Fld@additionalactions
     }

\ExplSyntaxOff
\def\PDFForm@List{%
  /Subtype/Widget%
  \Fld@annotflags
  \Fld@pageobjref
  \Fld@annotnames
  /FT/Ch%
  \Fld@flags
  /Q \Fld@align
  /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
  \ifcase0\ifnum\Fld@rotation=\z@   \else 1\fi
          \ifx\Fld@bordercolor\relax\else 1\fi
          \ifx\fld@bcolor\relax     \else 1\fi
          \space
  \else
    /MK<<%
      \ifnum\Fld@rotation=\z@
      \else
        /R \Fld@rotation
      \fi
      \ifx\Fld@bordercolor\relax
      \else
        /BC[\Fld@bordercolor]%
      \fi
      \ifx\Fld@bcolor\relax
      \else
        /BG[\Fld@bcolor]%
      \fi
    >>%
  \fi
  /DA(/Helv \strip@pt\Fld@charsize\space Tf%
      \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
  \Fld@choices
  \Fld@additionalactions
}
\ExplSyntaxOn
\def\PDFForm@Radio
  {
    /Subtype/Widget
    ~\Fld@annotflags
    ~\Fld@pageobjref
    ~\Fld@annotnames
    /FT/Btn
   \Fld@flags
   /H/P
   /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
   /MK<<
    \ifnum\Fld@rotation=\z@
    \else
      /R~\Fld@rotation
    \fi
    \ifx\Fld@bordercolor\relax
    \else
      /BC[\Fld@bordercolor]%
    \fi
    \ifx\Fld@bcolor\relax
    \else
      /BG[\Fld@bcolor]%
    \fi
     /CA(\Hy@escapestring{\Fld@radiosymbol})%
    >>
    /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
      \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
   \ifx\Fld@default\@empty
    /V/Off%
    /DV/Off%
   \else
```

```
    /V/\Fld@default
    /DV/\Fld@default
   \fi
   \Fld@additionalactions
 }
\ExplSyntaxOff
\ExplSyntaxOn
% Does an appeareance dict make sense here?
\def\PDFForm@Text
  {
    /Subtype/Widget
    ~\Fld@annotflags
    ~\Fld@pageobjref
    ~\Fld@annotnames
    /FT/Tx
    ~\Fld@flags
    /Q~\Fld@align
    /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
   \bool_if:nT
     {
        !\int_compare_p:nNn {\Fld@rotation} = {0}
       ||
        \tl_if_exist_p:N \Fld@bordercolor
       ||
        \tl_if_exist_p:N \Fld@bcolor
     }
     {
       /MK
         <<
           \int_compare:nNnF {\Fld@rotation} = {0}
             {
               /R~\Fld@rotation
             }
           \tl_if_exist:NT \Fld@bordercolor
             {
               /BC[\Fld@bordercolor]
             }
           \tl_if_exist:NT \Fld@bcolor
             {
               /BG[\Fld@bcolor]
             }
        >>
     }
   /DA
     (
       /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
       \tl_if_empty:NF {\c_space_tl\Fld@color}
     )
   /DV(\Hy@escapestring{\Fld@default})
   /V(\Hy@escapestring{\Fld@value})
   ~\Fld@additionalactions
   \int_compare:nNnT { \Fld@maxlen}>{0}
     {
       /MaxLen~\Fld@maxlen
     }
  }
\ExplSyntaxOff

\def\PDFForm@Submit{%
  /Subtype/Widget%
  \Fld@annotflags
  \Fld@pageobjref
  \Fld@annotnames
  /FT/Btn%
  \Fld@flags
  /H/P%
  /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
  \ifcase0\ifnum\Fld@rotation=\z@   \else 1\fi
          \ifx\Fld@bordercolor\relax\else 1\fi
          \space
  \else
    /MK<<%
      \ifnum\Fld@rotation=\z@
      \else
        /R \Fld@rotation
      \fi
      \ifx\Fld@bordercolor\relax
      \else
        /BC[\Fld@bordercolor]%
      \fi
    >>%
  \fi
  /A<<%
    /S/SubmitForm%
    /F<<%
      /FS/URL%
      /F(\Hy@escapestring{\Form@action})%
    >>%
    \Fld@submitflags
  >>%
  \Fld@additionalactions
}
\ExplSyntaxOn
       \def\PDFForm@Reset{%
         /Subtype/Widget%
         \Fld@annotflags
         \Fld@pageobjref
         \Fld@annotnames
         /FT/Btn%
         \Fld@flags
         /H/P%
         /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
         \ifcase0\ifnum\Fld@rotation=\z@   \else 1\fi
                 \ifx\Fld@bordercolor\relax\else 1\fi
                 \space
         \else
           /MK<<%
             \ifnum\Fld@rotation=\z@
             \else
               /R~\Fld@rotation
             \fi
             \ifx\Fld@bordercolor\relax
             \else
               /BC[\Fld@bordercolor]%
             \fi
           >>%
         \fi
         /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
         /A<</S/ResetForm>>%
         \Fld@additionalactions
       }%


 %these patterns are used in hyperref checks.
%it is unclear if they are really useful and if a backend support is
%needed.
\str_case:VnF \c_sys_backend_str
 {
   { pdfmode }
   {
     \def\HyPat@ObjRef
      {
       [0-9]*[1-9][0-9]*~0~R
      }
   }
   { dvipdfmx }
   {
     \def\HyPat@ObjRef
      {
        @[^~]+
      }
   }
   { xdvipdfmx }
   {
     \def\HyPat@ObjRef
      {
        @[^~]+
      }
   }
 }
 { %also set in hyperref sty, so probably not needed.
   \def\HyPat@ObjRef/{.+}
 }


\ExplSyntaxOff
% UF: removed Hy@writebookmark
%     \Hy@currentbookmarklevel{0}
%     \Hy@numberline
%     \__hypwritetorep
%     counter{bookmark@seq@number}
% removed \HyPsd@SanitizeForOutFile, not needed
% removed \currentpdfbookmark, defined by bookmark,
% should use \newcommand there
% removed \subpdfbookmark, defined by bookmark,
% should use \newcommand there
% removed \belowpdfbookmark, defined by bookmark,
% should use \newcommand there
% removed \pdfbookmark, defined by bookmark,
% \BOOKMARK
% \@BOOKMARK
%% \RequirePackage{rerunfilecheck}[2009/12/10]
%% removed \Hy@OutlineRerunCheck, unneeded with bookmark
%% removed \ReadBookmarks / unneeded with bookmark.
%% removed \Hy@OutlineName
%% removed \check@bm@number
%% removed \calc@bm@number

\ifHy@implicit
\else
  \expandafter\endinput
\fi
\newlength\Hy@SectionHShift
\def\Hy@SectionAnchorHref#1{%
  \ifx\protect\@typeset@protect
    \Hy__hypSectionAnchor{#1}%
  \fi
}
\DeclareRobustCommand*{\Hy__hypSectionAnchor}[1]{%
  \leavevmode
  \hbox to 0pt{%
    \kern-\Hy@SectionHShift
    \Hy@raisedlink{%
      \hyper@anchorstart{#1}\hyper@anchorend
    }%
    \hss
  }%
}
\let\H@old@ssect\@ssect
\def\@ssect#1#2#3#4#5{%
  \Hy@MakeCurrentHrefAuto{section*}%
  \setlength{\Hy@SectionHShift}{#1}%
  \begingroup
    \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
    \toks\tw@\expandafter{%
      \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
      #5%
    }%
  \edef\x{\endgroup
    \the\toks@{\the\toks\tw@}%
  }\x
}
\let\H@old@schapter\@schapter
\def\@schapter#1{%
  \begingroup
    \let\@mkboth\@gobbletwo
    \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
    \Hy@raisedlink{%
      \hyper@anchorstart{\@currentHref}\hyper@anchorend
    }%
  \endgroup
  \H@old@schapter{#1}%
}
\ltx@IfUndefined{@chapter}{}{%
  \let\Hy@org@chapter\@chapter
  \def\@chapter{%
    \def\Hy@next{%
      \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
      \Hy@raisedlink{%
        \hyper@anchorstart{\@currentHref}\hyper@anchorend
      }%
    }%
    \ifnum\c@secnumdepth>\m@ne
      \ltx@IfUndefined{if@mainmatter}%
      \iftrue{\csname if@mainmatter\endcsname}%
        \let\Hy@next\relax
      \fi
    \fi
    \Hy@next
    \Hy@org@chapter
  }%
}
\let\H@old@part\@part
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname chapter\endcsname\relax
  \let\Hy@secnum@part\z@
\else
  \let\Hy@secnum@part\m@ne
\fi
\def\@part{%
  \ifnum\Hy@secnum@part>\c@secnumdepth
    \phantomsection
  \fi
  \H@old@part
}
\let\H@old@spart\@spart
\def\@spart#1{%
  \Hy@MakeCurrentHrefAuto{part*}%
  \Hy@raisedlink{%
    \hyper@anchorstart{\@currentHref}\hyper@anchorend
  }%
  \H@old@spart{#1}%
}
\let\H@old@sect\@sect
\def\@sect#1#2#3#4#5#6[#7]#8{%
  \ifnum #2>\c@secnumdepth
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
  {%
    \Hy@MakeCurrentHrefAuto{section*}%
    \setlength{\Hy@SectionHShift}{#3}%
    \begingroup
      \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[{#7}]}%
      \toks\tw@\expandafter{%
        \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
        #8%
      }%
    \edef\x{\endgroup
      \the\toks@{\the\toks\tw@}%
    }\x
  }{%
    \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[{#7}]{#8}%
  }%
}
\expandafter\def\csname Parent-4\endcsname{}
\expandafter\def\csname Parent-3\endcsname{}
\expandafter\def\csname Parent-2\endcsname{}
\expandafter\def\csname Parent-1\endcsname{}
\expandafter\def\csname Parent0\endcsname{}
\expandafter\def\csname Parent1\endcsname{}
\expandafter\def\csname Parent2\endcsname{}
\expandafter\def\csname Parent3\endcsname{}
\expandafter\def\csname Parent4\endcsname{}
%%
%% End of file ‘hgeneric-testphase.def’.
⟨/package⟩
```

```
⟨*colorscheme⟩
% collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
% cite color ignored, as it doesn’t fit ... should be done by cite packages ?
% linkcolor=
%,filecolor=
%,urlcolor=
%,menucolor=
%,runcolor=
%,linkbordercolor=
%,filebordercolor=
%,urlbordercolor=
%,menubordercolor=
%,runbordercolor=

\prop_const_from_keyval:Nn \c__hyp_colorscheme_original_prop
 {
   linkcolor       = [rgb]{1,0,0}, %red
   filecolor       = [rgb]{0,1,1}, %cyan
   urlcolor        = [rgb]{1,0,1}, %magenta
   menucolor       = [rgb]{1, 0, 0}, %red
   runcolor        = [rgb]{0,1,1}, %cyan
 %------------
   linkbordercolor = [rgb]{1, 0 ,0 },
   filebordercolor = [rgb]{0, .5, .5},
   urlbordercolor  = [rgb]{0, 1, 1},
   menubordercolor = [rgb]{1, 0, 0},
   runbordercolor  = [rgb]{0, .7, .7}
 }

\prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
 {
   linkcolor       = [rgb]{0,0.2,0.6},
   filecolor       = [rgb]{0.8,0,0.8},
   urlcolor        = [rgb]{0.8,0,0.8},
   menucolor       = [rgb]{0,0.2,0.6},
   runcolor        = [rgb]{0.8,0,0.8},
 %-------------     %--------
   linkbordercolor = [rgb]{0,0.2,0.6},
   filebordercolor = [rgb]{0.8,0,0.8},
   urlbordercolor  = [rgb]{0.8,0,0.8},
   menubordercolor = [rgb]{0,0.2,0.6},
   runbordercolor  = [rgb]{0.8,0,0.8}
 }

\prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
 { %two colors: intern/extern
   linkcolor       = [rgb]{0.79216, 0, 0.12549},
   filecolor       = [rgb]{0.01961, 0.44314, 0.6902},
   urlcolor        = [rgb]{0.01961, 0.44314, 0.6902},
   menucolor       = [rgb]{0.79216, 0, 0.12549 },
   runcolor        = [rgb]{0.01961, 0.44314, 0.6902 },
 %-------------     %--------
   linkbordercolor = [rgb]{0.79216, 0, 0.12549},
   filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
   urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
   menubordercolor = [rgb]{0.79216, 0, 0.12549 },
   runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
 }

\prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
 { %all darkgray
   linkcolor       = [rgb]{0.4 ,0.4 ,0.4 },
   filecolor       = [rgb]{0.4 ,0.4 ,0.4 },
   urlcolor        = [rgb]{0.4 ,0.4 ,0.4 },
   menucolor       = [rgb]{0.4 ,0.4 ,0.4 },
   runcolor        = [rgb]{0.4 ,0.4 ,0.4 },
 %-------------     %--------
   linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
   filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
   urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
   menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
   runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
 }

\prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
 { %dvipsnam.def
   linkcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
   filecolor       = [rgb]{1, 0, 0}, %Red
   urlcolor        = [rgb]{0.06, 0.46, 1}, %NavyBlue
   menucolor       = [rgb]{1, 0, 0}, %Red
   runcolor        = [rgb]{1, 0, 0}, %Red
 %-------------     %------------------
   linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
   filebordercolor = [rgb]{1, 0, 0}, %Red
   urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
   menubordercolor = [rgb]{1, 0, 0}, %Red
   runbordercolor  = [rgb]{1, 0, 0} %Red
 }

\prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
 { %dvipsnam.def
   linkcolor       = [rgb]{0.72, 0, 0}, %BrickRed
   filecolor       = [rgb]{0, 1, 0},    %Green
   urlcolor        = [rgb]{0.64, 0.08, 0.98}, %Mulberry
   menucolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
   runcolor        = [rgb]{0.64, 0.08, 0.98}, %Mulberry
 %-------------     %------------------
   linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
   filebordercolor = [rgb]{0, 1, 0},    %Green
   urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
   menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
   runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
 }


\prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
 {
   linkcolor       = [rgb]{0.50196, 0, 0.02353},
   filecolor       = [rgb]{0.07451, 0.09412, 0.46667},
   urlcolor        = [rgb]{0.54118, 0, 0.52941},
   menucolor       = [rgb]{0.44706, 0.45882, 0},
   runcolor        = [rgb]{0.07451, 0.46667, 0.46275},
 %-------------      %-------------
   linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
   filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
   urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
   menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
   runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
 }

\prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
 {
   linkcolor       = [rgb]{0,0,0},
   filecolor       = [rgb]{0,0,0},
   urlcolor        = [rgb]{0,0,0},
   menucolor       = [rgb]{0,0,0},
   runcolor        = [rgb]{0,0,0},
 %-------------     %--------
   linkbordercolor = [rgb]{0,0,0},
   filebordercolor = [rgb]{0,0,0},
   urlbordercolor  = [rgb]{0,0,0},
   menubordercolor = [rgb]{0,0,0},
   runbordercolor  = [rgb]{0,0,0}
 }
```

```
⟨/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.