

The hyperref-generic module

A generic driver for hyperref

The L^AT_EX Project*

Version 0.96c, released 2023-11-17

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

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

The driver *requires* the new PDF management code, so documents wanting to use it should start like this (this requires L^AT_EX-2022-06-01 or newer):

```
\DocumentMetadata %loads the PDF management and 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 are added by code from `l3pdfmeta`. (without the `pdfmanagement` the XMP-metadata can be added with packages 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 a 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}}
```

- It is possible to store titles in more than one language. If the value begins with an “optional argument” which represents a language tag, the value is taken as a comma list and splitted. The first value is used for the Info dictionary, the others are used in the XMP-metadata. Commas in a title must then be protected with braces:

```
pdftitle = {[en]English Title,[de] Deutscher Titel,[fr]{titre français, avec com
```

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

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

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

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

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

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

4 Dates

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

5 PDF page size (mediabox)

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

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

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

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

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

6 Commands to create “external” references

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

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

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

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

(Expandable) commands are expanded and special chars can also be input by commands but beside this no conversion is done: for all input `hyperref` basically assumes that the input is already a valid percent encoded url or a valid file name. `hyperref` also doesn't extend or add protocols.

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

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

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

6.1 Special problem: links to files

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

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

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

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

6.2 Splits

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

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

For this reason three new commands have been create:

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

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

6.3 Options

All `\href` commands and `\url` have an option argument for keyval syntax. It accepts the following keys. Not all keys make sense for all keys, but they don't error, they are silently ignored. The optional argument can currently not be used together with the `\urldef` command.

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

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

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

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

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

Some main changes are

- The default colors have been changed.

- Citations have by default no special color, they are colored like other internal links. You can use `citecolor` and `citebordercolor` to assign them a special color. This color is not reset if you use `allcolors` or switch to another color scheme. If you want the colors to follow `linkcolor` again you should remove the label `hyp/cite` and/or `hyp/citeborder` from the hook `hyp/link/cite`.
- a number of color schemes have been predefined.

7.1 Background information

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

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

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

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

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

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

7.2 New Keys

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

- Beside `pdfborder` there are also `linkborder`, `urlborder` etc

- Beside `pdfhighlight` there are also `linkhighlight`, `urlhighlight` etc
- Beside `pdfborderstyle` there are also `linkborderstyle`, `urlborderstyle` etc
- Beside `colorlinks` there are also `colorlink`, `colorurl` etc
- Beside `ocgcolorlinks` there are also `ocgcolorlink`, `ocgcolorurl`, etc TODO
- Beside `hidelinks` there are also `hidelink`, `hideurl`, etc
- `bordercolormodel` allows to set the model used in annotations, the allowed values are `rgb` or `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 `primary-colors` (the colors as `hyperref` uses normally), `phelype`, `daleif`, `szabolcsA`, `szabolcsB`, `tivv`, `julian`, `henryford`.

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

The default is `phelype`.

7.3 Public interfaces

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

```
\l_hyp_annot_colorlink_bool,
\l_hyp_annot_colorurl_bool,
\l_hyp_annot_colorfile_bool,
\l_hyp_annot_colorruncolor_bool,
\l_hyp_annot_colormenu_bool,
\l_hyp_annot_ocgcolorlink_bool,
\l_hyp_annot_ocgcolorurl_bool,
\l_hyp_annot_ocgcolorfile_bool,
\l_hyp_annot_ocgcolorruncolor_bool,
\l_hyp_annot_ocgcolormenu_bool,
```

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

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

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

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

7.4 Changed behaviour

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

Color keys accept the following input syntax:

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

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

frenchlinks The option `frenchlinks` does nothing at all.

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

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

8 PDF strings

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

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

The new method is under heavy development!

Important differences here are

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

9 Package options from hyperref

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

Options that still must be set as package options are

- `backref`
- `CJKbookmarks` this key should not be used anymore. At some time it will be removed.
- `destlabel` (destination names are taken from `\label` if possible)
- `encap`
- `hyperfigures` (according to the `hyperref` manual it makes figures hyper links, but actually is a no-op for most drivers, and it does nothing with this driver either.)
- `hyperfootnotes`
- `hyperindex`
- `implicit` (redefine `LATEX` internals)
- `nesting` unneeded key, see comment below in 14. At some time it will be either removed or extended (if some use can be found).
- `pagebackref`
- `pdfpagelabels` (set PDF page labels)
- `psdextra` this loads some extra definitions used by `\pdfstringdef`. The new driver uses `\pdfstringdef` only for the bookmarks, for other strings it is not relevant.

Options that can be without problems set as package options are

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

Ignored options:

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

10 Disabling links

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

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

11 Draftmode

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

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

12 Dropped options

A number of options are ignored by this driver

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

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

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

unicode This is always true.

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

```
\DocumentMetadata{pdfstandard=A-2b}
```

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

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

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

13 Destinations

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

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

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

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

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

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

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

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

`pdfremotestartview` (*setup key*)

- `Fit`, `FitB`, `FitH`, `FitV`, `FitBH`, `FitBV` which can be followed by a positive integer (separated by a space) or the keyword `null`. The number can be gives as a *<dimension expression>* surrounded with `\hypercalcbp`. The driver redefines this command to use `\dim_to_decimal_in_bp:n`.

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

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

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

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

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

13.1 Names of destinations

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

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

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

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

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

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

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

14 Assorted key descriptions

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

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

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

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

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

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

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

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

will create

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

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

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

linkborder (*setup key*)

urlborder (*setup key*)

runborder (*setup key*)

menuborder (*setup key*)

according to the reference set round corners, but PDF viewer seem to ignore it. The third number is the line width of the border. Settings done with `pdfborderstyle` should take precedence.

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

File I

hyperref-generic driver implementation

```

1 <*package>
2 <@@=hyp>
3 \ProvidesFile{hggeneric-testphase.def}[2023-11-17 v0.96c %
4   generic Hyperref driver for the LaTeX PDF management testphase bundle]
5
6 \RequirePackage{etoolbox} %why?
```

Temporary command definition, can be remove when hyperref is update too.

```

7 \long\def\Hy@ReturnAfterFi#1\fi{\fi#1}
8 \ExplSyntaxOn
9 \file_input:n {hyperref-colorschemes.def}
10 \ExplSyntaxOff
```

1 messages

Redirect the message name:

```

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

At first a message for the testing of the resource management

```
13 \cs_if_exist:NTF \DocumentMetadata
14 {
15   \msg_new:nnnn
16     { hyp }
17     { missing-resource-management }
18     { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
19     {
20       Activate~it~with ~\
21       \tl_to_str:n{\DocumentMetadata{<options>}}~\
22       before~\tl_to_str:n{\documentclass}
23     }
24 }
25 {
26   \msg_new:nnnn
27     { hyp }
28     { missing-resource-management }
29     { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
30     {
31       Activate~it~with ~\
32       \tl_to_str:n{\RequirePackage{pdfmanagement-testphase}}~\
33       \tl_to_str:n{\DocumentMetadata{<options>}}~\
34       before~\tl_to_str:n{\documentclass}
35     }
36 }
```

The pdfversion should be set in \DocumentMetadata

```
37 \msg_new:nnnn
38 { hyp }
39 { pdfversion-disabled }
40 {
41   This~hyperref~driver~ignores~the~pdfversion~key!~\
42   Set~the~pdfversion~in~\token_to_str:N \DocumentMetadata
43 }
44 {
45   For example~\
46   \tl_to_str:n
47   {
48     \DocumentMetadata { pdfversion=1.7 }
49   }
50 }
```

A generic message for ignored keys.

```
51 \msg_new:nnn
52 { hyp }
53 { key-dropped }
54 {
55   This~hyperref~driver~ignores~the~key~#1!~\
56   Please~check~the~documentation.
57 }
```

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

```
58 \msg_new:nnn
59 { hyp }
60 { pdfa-no-push-button }
```

```

61 { PDF/A:~Push-button-with-JavaScript-is-prohibited }
62
63 \msg_new:nnn
64 { hyp }
65 { pdfa-no-reset-button }
66 { PDF/A:~Reset-action-is-prohibited }

```

pdf/A message for not allowed Named actions

```

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

```

A message if the destination name is empty.

```

71 \msg_new:nnn
72 { hyp }
73 { empty-destination-name }
74 {
75   Empty-destination-name,\
76   using~'#1'
77 }

```

A message if the destination check fails

```

78 \msg_new:nnn
79 { hyp }
80 { invalid-destination-value }
81 {
82   Invalid-value~'#1'~of~'#2'  \
83   is-replaced-by~'Fit'~\msg_line_context:.
84 }

```

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

```

85 \msg_new:nnn
86 { hyp }
87 { ignore-deprecated-or-unknown-option-in-pdf-version }
88 {
89   Option~'#1'~is-unknown-or-deprecated-in\
90   pdf-version~#2.~Ignored.
91 }
92 \msg_new:nnn
93 { hyp }
94 { ignore-deprecated-or-unknown-value-in-pdf-version }
95 {
96   Value~'#1'~is-unknown-or-deprecated-in\
97   pdf-version~#2.~Ignored.
98 }
99 \msg_new:nnn
100 { hyp }
101 { replace-deprecated-or-unknown-value-in-pdf-version }
102 {
103   Value~'#1'~is-unknown-or-deprecated-in\
104   pdf-version~#2. Value~'#3'~is used instead.
105 }

```

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:


```

106 \msg_new:nnn
107 { hyp }
108 { unknown-key }
109 {
110     unknown-key~#2-of~module~'#1'~set-to~'#3'.
111 }
112 \msg_new:nnn
113 { hyp }
114 { unknown-key-to-Hyp }
115 {
116     ignored-in~family~Hyp~unknown-key~#1.
117 }

```

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

```

118 \cs_new:Npn \__hyp_clist_display:n #1 {*~#1\\}
119 \msg_new:nnn
120 { hyp }
121 { unknown-choice }
122 {
123     Value~'#3'~is~invalid~for~key~'#1'.\\
124     The~key~accepts~only~the~choices\\
125     \clist_map_function:nN { #2 }\__hyp_clist_display:n
126 }
127
128 \msg_new:nnn
129 { hyp }
130 { unknown-choice+empty }
131 {
132     Value~'#3'~is~invalid~for~key~'#1'.\\
133     The~key~accepts~only~the~choices\\
134     \clist_map_function:nN { #2 }\__hyp_clist_display:n
135     An~empty~value~removes~the~setting.
136 }
137
138 \msg_new:nnn
139 { hyp }
140 { no-bool }
141 {
142     Value~'#2'~is~invalid~for~key~'#1'.\\
143     The~key~accepts~only~the~choices\\
144     *~true\\
145     *~false \\
146     *~and~an~empty~value~which~removes~the~setting.\\
147     No~value~is~equivalent~to~using~'true'.
148 }

```

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

```

149 \msg_new:nnn
150 { hyp }
151 { empty-info-value }
152 {
153     Empty-value~for~key~#1.\\
154     This~isn't~honored~by~all~backends.
155 }

```

2 Variants

```

156 \cs_generate_variant:Nn\pdf_destination:nn {nf}
157 \cs_generate_variant:Nn\pdf_object_ref:n {e}
158 \cs_generate_variant:Nn\pdf_pageobject_ref:n {e}

```

3 Overwriting/providing commands from hyperref

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

```

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

160 %\cs_set_protected:Npn \PDF@SetupDoc{}
161 %\PDF@FinishDoc{}% dummy needed for hyperref ...

```

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

`\hypercalcbp`

```

162 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n

```

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

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

```

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

```

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

```

167 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
168 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:

169 \legacy_if:nT { Hy@setpdfversion }
170 {
171   \msg_warning:nn { hyp }{ pdfversion-disabled }
172 }
173 \Hy@DisableOption{pdfversion}

```

`\Acrobatmenu` should use the new internal link command

```

174 \RenewDocumentCommand \Acrobatmenu { m m }
175 {
176   \hyper@linknamed {#1} {#2}
177 }

```

`\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!

```

178 % TODO should go at some time ...
179 % \kv@set@family@handler{Hyp}
180 % { \msg_warning:nne {hyp}{unknown-key-to-Hyp}{#1} }
181 \cs_set_protected:Npn \hypersetup #1
182 {
183   \kvsetkeys{Hyp} {#1}
184   \keys_set:nn { hyp / setup }{ #1 }
185 }

```

```

186 % TODO for now unknown keys should only give warnings.
187 \keys_define:nn { hyp / setup }
188 {
189   unknown .code:n =
190   {
191     \msg_warning:nneee { hyp } { unknown-key }
192     { hyp~/~setup }{ \l_keys_key_str } { #1 }
193   }
194 }

```

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

```

195 \keys_define:nn { hyp / setup }
196 {
197   next-anchor .code:n =
198   {
199     \AddToHookNext{__hyp/dest/make}
200     {\Hy@MakeCurrentHref{#1}}
201   }
202 }

```

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`)

```

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

```

```

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

```

```

285     \hook_use:n{cmd/href/after}
286   }%
287 \endgroup
288
289 \hook_new_pair:nn{cmd/hrefurl/before}{cmd/hrefurl/after}
290
291 \DeclareRobustCommand*{\hrefurl}[1] []
292 {
293   \mode_leave_vertical:
294   \hook_use:n{cmd/href/before}
295   \group_begin:
296   \keys_set:nn { hyp / href } {#1}
297   \bool_if:NTF \l__hyp_href_url_encode_bool
298   {
299     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
300   }
301   {
302     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
303   }
304   \hyper@normalise\__hyp_href_url_aux:nn}
305
306 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
307 {
308   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
309   \group_end:
310   \hook_use:n{cmd/href/after}
311 }
312
313 \hook_new_pair:nn{cmd/hrefpdf/before}{cmd/hrefpdf/after}
314 \DeclareRobustCommand*{\hrefpdf}[1] []
315 {
316   \mode_leave_vertical:
317   \hook_use:n{cmd/hrefpdf/before}
318   \group_begin:
319   \keys_set:nn { hyp / href } {#1}
320   \hyper@normalise\__hyp_href_pdf_aux:nn
321 }
322
323 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
324 {
325   \exp_args:Nno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
326   \group_end:
327   \hook_use:n{cmd/hrefpdf/after}
328 }
329
330 \hook_new_pair:nn{cmd/hrefrun/before}{cmd/hrefrun/after}
331 \DeclareRobustCommand*{\hrefrun}[1] []
332 {
333   \mode_leave_vertical:
334   \hook_use:n{cmd/hrefrun/before}
335   \group_begin:
336   \keys_set:nn { hyp / href } {#1}
337   \hyper@normalise\__hyp_href_run_aux:nn
338 }

```

```

339
340 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
341 {
342   \exp_args:Nno\hyper@linklaunch{#1}{#2}{\l__hyp_href_run_parameter_tl}
343   \group_end:
344   \hook_use:n{cmd/hrefrun/after}
345 }
346
347
348 \hook_new_pair:nn{cmd/url/before}{cmd/url/after}
349
350 \DeclareRobustCommand*\url{[1] []}
351 {
352   \mode_leave_vertical:
353   \hook_use:n{cmd/url/before}
354   \group_begin:
355   \keys_set:nn {hyp / href } {#1}
356   \bool_if:NTF \l__hyp_href_url_encode_bool
357   {
358     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
359   }
360   {
361     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
362   }
363   \hyper@normalise\__hyp_href_url_aux:n
364 }
365
366 \cs_new_protected:Npn \__hyp_href_url_aux:n #1
367 {
368   \exp_args:Nno
369   \hyper@linkurl{\__hyp_href_url_format: {#1}}
370   {\l__hyp_href_url_protocol_tl#1}
371   \group_end:
372   \hook_use:n{cmd/url/after}
373 }
374

```

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

```

375
376 \def\urldef#1#2{\begingroup\def\url{\hyper@normalise\url@}\setbox\z@\hbox\bgroup
377   \def\Url@HyperHook##1\endgroup{\Url@def{#1}{#2}}}%
378   % Because hyperref breaks \urldef and does not define its own (Grrrr!)...
379   \def\url@##1{\egroup\endgroup\DeclareRobustCommand#1{#2{##1}}}%
380   #2}
381

```

make the new commands compatible with `\pdfstringdef`:

```

382 \NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
383 \pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
384 \pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}
385 \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.

```
386 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
387 {
388   %\tl_set:cn {@#1}{#2}
389   \AddToDocumentProperties[hyperref]{#1}{#2}
390 }
391 \cs_generate_variant:Nn \__hyp_store_metadata:nn {en,ne,ee}
```

4.2 citecolor

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

```
392 \hook_new:n{hyp/link/cite}
393 %\color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
394 %\color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
395 \keys_define:nn { hyp / setup }
396 {
397   ,citecolor .code:n = {\__hyp_color_set:ne {hyp/color/cite}{#1}\__hyp_citecolor_hook_init
398   ,citebordercolor
399   .code:n = {\__hyp_color_set:ne {hyp/color/citeborder}{#1}\__hyp_citebordercolor_hook_i
400 }
401 \cs_new_protected:Npn \__hyp_citecolor_hook_init:
402 {
403   \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
404   {
405     \keys_set:nn { hyp / setup }
406     {
407       linkcolor      = hyp/color/cite
408     }
409   }
410   \cs_gset_eq:NN \__hyp_citecolor_hook_init: \prg_do_nothing:
411 }
412 \cs_new_protected:Npn \__hyp_citebordercolor_hook_init:
413 {
414   \hook_gput_code:nnn { hyp/link/cite }{hyp/citeborder}
415   {
416     \keys_set:nn { hyp / setup }
417     {
418       linkbordercolor = hyp/color/citeborder
419     }
420   }
421   \cs_gset_eq:NN \__hyp_citebordercolor_hook_init: \prg_do_nothing:
422 }
423
```

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

```

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

```

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

```

427 \AddToHook { package/bookmark/after}
428 {
429   \define@key{BKM}{color}
430   {
431     \__hyp_color_set:ne {__hyp/tmpa}{#1}
432     \color_export:nVN
433       {__hyp/tmpa}
434       \g__hyp_bordercolormodel_str
435       \BKM@color
436   }
437 }
438 \legacy_if:nT { Hy@bookmarks }
439 {
440   \AddToHook{begindocument/before}[hyperref/bookmark]
441   {
442     \RequirePackage{bookmark}
443   }
444 }
445 \legacy_if:nT { Hy@draft}
446 {
447   \PassOptionsToPackage{draft}{bookmark}
448 }

```

6 Reference and label commands

This uses the in-built property module.

`__hyp_property_record:nn`

```

449 %

```

A label command which adds the space commands from LaTeX:

```

450 \cs_new_protected:Npn \__hyp_property_record:nn #1 #2 %label/attributes
451 {
452   \@bsphack
453   \property_record:nn{#1}{#2}
454   \@esphack
455 }

```

we generate a few variants. We use ee-variants as they already exist in the module and once this is there it can go here.

```

456 \cs_generate_variant:Nn \__hyp_property_record:nn {ee}

```


(End of definition for `__hyp_property_record:nn`.)

7 Variables

7.1 Private temporary variables

At first a few generic tmp variables

```

\l__hyp_tmpa_tl
\l__hyp_tmpa_seq 457 \box_new:N \l__hyp_tmpa_box
\l__hyp_tmpa_int 458 \tl_new:N \l__hyp_tmpa_tl
\l__hyp_tmpa_box 459 \seq_new:N \l__hyp_tmpa_seq
\l__hyp_tmpa_str 460 \int_new:N \l__hyp_tmpa_int
461 \str_new:N \l__hyp_tmpa_str

```

(End of 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
462 \tl_new:N \l__hyp_dest_name_tmpa_tl
463 \tl_new:N \l__hyp_uri_tmpa_tl
464 \tl_new:N \l__hyp_filename_tmpa_tl
465 \tl_new:N \l__hyp_para_tmpa_tl
466 \str_new:N \l__hyp_text_tmpa_str
467 \str_new:N \g__hyp_text_tmpa_str

```

TODO: document and check use!

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

7.2 Constants

This variable is used if a destination name is empty.

```

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

```

(End of definition for `\c__hyp_dest_undefined_tl`.)

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

```

\c__hyp_annot_types_seq
\c__hyp_map_annot_hyp_prop
\c__hyp_map_hyp_annot_prop
469 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
470 {url,link,file,menu,run}
471 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
472 {
473   URI    = url,
474   GoTo   = link,
475   GoToR  = file,
476   Named  = menu,
477   Launch = run
478 }
479 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
480 {
481   url    = URI,
482   link   = GoTo,
483   file   = GoToR,
484   menu   = Named,
485   run    = Launch

```

```
486 }
```

```
487
```

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

7.3 Variables

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

```
488 \tl_new:N \g__hyp_dest_pdfstartpage_tl
489 \tl_new:N \g__hyp_dest_pdfstartview_tl
490 \tl_new:N \l__hyp_dest_pdfremotestartview_tl
```

(End of 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

491 \tl_new:N \l__hyp_text_enc_uri_print_tl
492 \tl_new:N \l__hyp_text_enc_info_print_tl
493 \tl_new:N \l__hyp_text_enc_dest_tl
494 \tl_new:N \l__hyp_text_enc_dest_print_tl
495 \tl_new:N \l__hyp_text_enc_file_print_tl
496 \tl_new:N \l__hyp_text_enc_para_print_tl
497
498 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
499 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
500 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
501 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
502 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
503 \tl_set:Nn \l__hyp_text_enc_para_print_tl {utf8/string}
```

(End of 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.

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

(End of definition for `\l__hyp_dest_pdfview_tl`.)

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

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

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

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

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

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

(End of definition for `\g__hyp_bordercolormodel_str`.)

7.4 Booleans

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

```
\l_hyp_annot_colorlink_bool
\l_hyp_annot_colorurl_bool
\l_hyp_annot_colorfile_bool
\l_hyp_annot_colorrund_bool
\l_hyp_annot_colormenu_bool
506 \seq_map_inline:Nn \c__hyp_annot_types_seq
507 {
508   \bool_new:c {l_hyp_annot_color#1_bool}
509 }
```

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

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

```
\l_hyp_annot_ocgcolorlink_bool
\l_hyp_annot_ocgcolorurl_bool
\l_hyp_annot_ocgcolorfile_bool
\l_hyp_annot_ocgcolorrund_bool
\l_hyp_annot_ocgcolormenu_bool
510 \seq_map_inline:Nn \c__hyp_annot_types_seq
511 {
512   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
513 }
```

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

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

```
514 \seq_map_inline:Nn \c_pdfannot_link_types_seq
515 {
516   \bool_new:c {l__hyp_annot_#1_bool}
517   \bool_set_true:c {l__hyp_annot_#1_bool}
518 }
```

(End of 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.

```
519 \box_new:N \l__hyp_dest_box
```

(End of 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.

```
520 \regex_const:Nn \c__hyp_dest_startview_regex
521 {
522   \A\ *
523   (?
524     (?:XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
525     |
526     (?:Fit\b|FitB\b)
527     |
528     (?:\bFitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1}
529     |
530     (?:FitR (?:\ +\d+|\ +\d*\.\d+)){4}\ )
531   )
532 }
```

(End of definition for `\c__hyp_dest_startview_regex`.)

7.7 PDF dictionaries

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

```
533 \pdfdict_new:n {l__hyp_page/Trans}
534 \pdfdict_put:nnn {l__hyp_page/Trans}{Type}{/Trans}
```

(End of 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`

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

(End of 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`

```
536 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
537 {
538   \str_set:Ne #2 {\text_purify:n { #1 } }
539 }
```

(End of 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`

```
540 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
541 {
542
543 }
```

(End of definition for `__hyp_text_cleanup:N`.)

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

#2 is str variable, #1 should be one of	
utf8/string	(lit) (utf8/string)
utf8/string-raw	lit (utf8/string)
utf8/URI	(percent encoded url)
utf8/URI-raw	percent encoded url
utf16/hex	<HEX> (utf16/hex)
utf16/hex-raw	HEX (utf16/hex)
utf16/string	(lit) (utf16/string)
utf16/string-raw	lit (utf16/string)

`__hyp_text_string_from_unicode:nN`

```

544 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
545 {
546   \pdf_string_from_unicode:nVN { #1 } #2 #2
547 }

```

(End of 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`

```

548 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
549 {
550   \group_begin:
551   \Hy@pdfstringtrue
552   \hook_use:n {hyp/text/pdfstring}
553   \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
554   \__hyp_text_cleanup:N      \l__hyp_text_tmpa_str
555   \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
556   \str_gset_eq:NN \g__hyp_text_tmpa_str\l__hyp_text_tmpa_str
557   \group_end:
558   \str_set_eq:NN #3 \g__hyp_text_tmpa_str
559 }
560 \cs_generate_variant:Nn \__hyp_text_pdfstring:nnN {enN,onN,eoN,ooN,noN}

```

(End of definition for __hyp_text_pdfstring:nnN.)

!!! temporary until all instances are gone

```

561 \cs_new_protected:Npn \Hy@pstringdef #1 #2
562 { \__hyp_text_pdfstring:enN {#2} {utf8/string-raw}#1 }

```

This is a special version for info keys:

`__hyp_text_pdfstring_info:nN`

```

563 \cs_new_protected:Npn \__hyp_text_pdfstring_info:nN #1 #2
564 {
565   \__hyp_text_pdfstring:noN { #1 }{ \l__hyp_text_enc_info_print_tl } #2
566 }

```

(End of definition for __hyp_text_pdfstring_info:nN.)

9 Pagelabels

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

The current implementation in hyperref/hyperref drivers:

xetex: hxtex.def, line 80-110

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

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

dvipdfm: identical to xetex

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

luatex identical to pdftex

The code in `hyperref` inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-30 is correctly split. If the counter can not be identified `hyperref` generates only /P entries with the whole content.

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

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

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

```
\__hyp_PageLabels_gpush:
  \Hy@PutCatalog
  \HyPL@StorePageLabel
567 \cs_new_protected:Npn \__hyp_PageLabels_gpush:
568   {
569     \pdfmanagement_add:nne {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
570   }
571
572 \def\Hy@PutCatalog #1 {}
573
574
575 \legacy_if:nT { Hy@pdfpagelabels }
576   {
577     \cs_set_protected:Npn \HyPL@StorePageLabel #1
578       {
579         \tl_gput_right:Ne \HyPL@Labels { \the\Hy@abspage<<#1>> }
580         \__hyp_PageLabels_gpush:
581       }
582   }
```

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

10 Core Hyperref Commands

Every `hyperref` has to define eight core command:

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

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

10.1 Anchors / destinations

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

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

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

```
\__hyp_destination:nn

583 \Hy@WrapperDef \__hyp_destination:nn #1 #2
584 {
585   \mode_if_horizontal:T { \@savs\spacefactor }
586   \Hy@SaveLastskip      %defined in hyperref
587   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
588   \__hyp_text_pdfstring:eoN
589   { \HyperDestNameFilter{#1} }
590   { \l__hyp_text_enc_dest_tl }
591   \l__hyp_tmpa_tl
592   \str_if_eq:nnTF {#2} {fitrbx}
593   {
594     \exp_args:NV
595     \pdf_destination:nnnn \l__hyp_tmpa_tl
596     { \box_wd:N \l__hyp_dest_box }
597     { \box_ht:N \l__hyp_dest_box }
598     { \box_dp:N \l__hyp_dest_box }
599   }
600   {
601     \exp_args:NV
602     \pdf_destination:nf
603     { \l__hyp_tmpa_tl }
604     { #2 }
605   }
606   \Hy@RestoreLastskip %defined in hyperref
607   \mode_if_horizontal:T { \spacefactor\@savs }
608 }
```

(End of definition for `__hyp_destination:nn`.)

These are the three destinations commands. They are modelled along the xetex version. It is not quite clear if really all three are needed for the backends supported by this driver, but changing the hyperref code would be difficult. 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 609 \tl_new:N\l_hyp_current_dest_name_tl
\hyper@anchorend
hyp/anchor
\l_hyp_current_dest_name_tl
```

```

610 \hook_new:n{hyp/anchor}
611 \cs_new_protected:Npn \hyper@anchor #1
612 {
613   \exp_args:NnV
614     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
615   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
616   \hook_use:n{hyp/anchor}
617 }
618
619 \cs_new_protected:Npn \hyper@anchorstart #1
620 {
621   \Hy@activeanchortrue
622   \exp_args:NnV
623     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
624   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
625   \hook_use:n{hyp/anchor}
626 }
627
628 \cs_new_protected:Npn \hyper@anchorend
629 {
630   \Hy@activeanchorfalse
631 }

```

(End of 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:

```

632 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
633 {
634   \mode_leave_vertical:
635   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
636   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
637   {
638     \msg_warning:nne
639       { hyp }
640       { empty-destination-name }
641       { \c__hyp_dest_undefined_tl }
642     \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
643   }
644   {
645     \__hyp_text_pdfstring:eoN
646     { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
647     { \l__hyp_text_enc_dest_tl }
648     \l__hyp_dest_name_tmpa_tl
649   }
650   \exp_args:No
651   \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
652 }
653

```



```

654 \cs_new_protected:Npn \__hyp_link_goto_end:
655 {
656   \pdfannot_link_goto_end:
657 }

```

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

```

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

```

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

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

```

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

```

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

```

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

```

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

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

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

```

658
659 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
660 {
661   \bool_if:NTF \l__hyp_annot_GoTo_bool
662   {
663     \Hy@VerboseLinkStart{#1}{#2}
664     \group_begin:
665     \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
666     \hook_use:n {hyp/link/#1}
667     \__hyp_link_goto_begin:nw {#2} #3 \Hy@xspace@end
668     \__hyp_link_goto_end:
669     \group_end:
670     \Hy@VerboseLinkStop
671   }
672   {\let\protect\relax#3}
673 }
674
675 \cs_new_protected:Npn \hyper@linkstart #1 #2 % #1 context, #2=destination name
676 {
677   \bool_if:NT \l__hyp_annot_GoTo_bool
678   {
679     \Hy@VerboseLinkStart{#1}{#2}% only for debug
680     \group_begin:
681     \tl_set:Nn \l_hyp_current_dest_name_tl {#2}

```

```

682     \hook_use:n {hyp/link/#1}
683     \__hyp_link_goto_begin:nw {#2}
684   }
685 }
686
687 \cs_new_protected:Npn \hyper@linkend
688 {
689   \bool_if:NT \l__hyp_annot_GoTo_bool
690   {
691     \__hyp_link_goto_end:
692     \group_end:
693     \Hy@VerboseLinkStop
694   }
695 }

```

10.3 URI links

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

```

696 \pdfdict_new:n {l_hyp/annot/A/URI}
697 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
698 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
699
700 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
701 {
702   \bool_if:NTF \l__hyp_annot_URI_bool
703   {
704     \group_begin:
705     \__hyp_text_pdfstring:eoN
706     { #2}
707     { \l__hyp_text_enc_uri_print_tl }
708     \l__hyp_uri_tmpa_tl
709     \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
710     \bool_if:NT \l__hyp_href_url_ismap_bool
711     {
712       \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
713     }
714     \cs_set_eq:NN \# \c_hash_str
715     \cs_set_eq:NN \% \c_percent_str
716     \Hy@safe@activetrue
717     \mode_leave_vertical:
718     \pdfannot_dict_put:nne {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
719     \pdfannot_link:nen { URI }
720     {
721     }
722     {
723       \let\protect\relax
724       #1
725       \Hy@xspace@end
726       \Hy@VerboseLinkStop %where is the start??
727     }
728     \group_end:
729   }
730   {{\let\protect\relax#1}}
731 }

```

10.4 GoToR Links files

```

733 \pdfdict_new:n {l_hyp/annot/A/GoToR}
734 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
735 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
736
737 \cs_generate_variant:Nn \pdffile_embed_file:nnn {noe}
738 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
739 {
740   \bool_if:NTF \l__hyp_annot_GoToR_bool
741   {
742     \group_begin:
743     \tl_set:Nx \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
744     \exp_args:Nx
745     \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
746     {
747       \pdfdict_put:nne { l_pdffile/Filespec } { Subtype } { \pdf_name_from_unicode_e:n {ap
748       \pdffile_embed_file:noe
749       {}
750       {\l__hyp_filename_tmpa_tl }
751       {\__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
752     }
753     \pdfdict_put:nne
754     {l_hyp/annot/A/GoToR}
755     {F}
756     {\pdf_object_ref:e {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
757     \__hyp_text_pdfstring:nnN
758     { #3 }
759     { \l__hyp_text_enc_dest_print_tl }
760     \l__hyp_dest_name_tmpa_tl
761     \tl_if_blank:eTF {#3}
762     {
763       \pdfdict_put:nne {l_hyp/annot/A/GoToR}{D}
764       {
765         [
766           \int_eval:n
767           { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
768           /\l__hyp_dest_pdfremotestartview_tl
769         ]
770       }
771     }
772     {
773       \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
774     }
775     \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>

```

776   \pdf_object_unnamed_write:ne{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
777   \pdfannot_dict_put:nne {link/GoToR}{A}{\pdf_object_ref_last:}
778   \pdfannot_link:nnn %expansion??
779   { GoToR }

```

```

780     {
781     }
782     {
783     \let\protect\relax
784     #1\Hy@xspace@end
785     \Hy@VerboseLinkStop %where is the start??
786     }
787     \group_end:
788   }
789   {{\let\protect\relax#1}}
790 }

```

10.5 Launch links

We define \hyper@linklaunch for naming consistency

```

791 \pdfdict_new:n {l_hyp/annot/A/Launch}
792 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
793 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
794
795 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
796 {
797   \bool_if:NTF \l__hyp_annot_Launch_bool
798   {
799     \group_begin:
800     \__hyp_text_pdfstring:nnN
801       { #1 }
802       { \l__hyp_text_enc_file_print_tl }
803       \l__hyp_filename_tmpa_tl
804     \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
805     \__hyp_text_pdfstring:noN
806       { #3 }
807       { \l__hyp_text_enc_para_print_tl }
808       \l__hyp_para_tmpa_tl
809     \bool_if:nTF
810       {
811         \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
812         ||
813         \pdf_version_compare_p:Nn > {1.9}
814       }
815       {
816         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
817       }
818       {
819         \pdfdict_put:nne
820           {l_hyp/annot/A/Launch}
821           {Win}
822           {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
823       }
824     \mode_leave_vertical:
825     \pdfannot_dict_put:nne {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}>>}
826     \pdfannot_link:nen
827       { Launch }
828       {
829         % /A
830         % <<

```

```

831 %           \pdfdict_use:n {l_hyp/annot/A/Launch}
832 %           >>
833     }
834     {
835         \let\protect\relax
836         #2\Hy@xspace@end
837         \Hy@VerboseLinkStop %where is the start??
838     }
839     \group_end:
840 }
841 {\let\protect\relax#2}}
842 }

```

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

```

843 \use:e
844 { % filename, anchor text, linkname
845     \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str #1 \exp_not:N \ \ #2 #3
846 }
847 {
848     \hyper@linklaunch {#1}{#2}{#3}
849 }

```

10.6 Named links (menu)

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

```

850 \pdfdict_new:n {l_hyp/annot/A/Named}
851 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
852 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
853
854 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
855 {
856     \bool_if:NTF \l__hyp_annot_Named_bool
857     {
858         \group_begin:
859         \pdfmeta_standard_verify:nnTF {named_actions}{#1}
860         {
861             \mode_leave_vertical:
862             \pdfdict_put:nne {l_hyp/annot/A/Named}{N}
863             {\pdf_name_from_unicode_e:n{#1}}
864             \pdfannot_dict_put:nne {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}>>}
865             \pdfannot_link:nnn { Named }
866             {
867 %           /A
868 %           <<
869 %           \pdfdict_use:n { l_hyp/annot/A/Named }
870 %           >>
871             }
872             {
873                 #2
874                 \Hy@xspace@end
875                 \Hy@VerboseLinkStop
876             }
877         }
878     }
879 }

```

```

878         {
879             \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
880             #2
881         }
882     \group_end:
883 }
884 {{\let\protect\relax#2}}
885 }
886

```

11 Link decorations

11.1 Functions to export and select colors

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

```

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

```

`__hyp_color_select:n` `__hyp_color_select:n {<color>}`

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

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

```

888 \cs_new_protected:Npn \__hyp_color_select:n #1
889 {
890     \tl_if_head_eq_charcode:nNTF {#1}[ %]
891     {
892         \__hyp_color_select_aux:wn #1
893     }
894     {
895         \color_select:n {#1}
896     }
897 }
898
899 \cs_new_protected:Npn \__hyp_color_select_aux:wn [#1] #2
900 {
901     \color_select:nn {#1}{#2}
902 }
903
904 \cs_generate_variant:Nn \__hyp_color_select:n {e}

```

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

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

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

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

```

905 \cs_new_protected:Npn \__hyp_color_set:nn #1 #2
906 {
907   \tl_if_head_eq_charcode:nNTF {#2}[ %]
908   {
909     \__hyp_color_set_aux:nwn { #1 } #2
910   }
911   {
912     \color_set:nn {#1} {#2}
913   }
914 }
915
916 \cs_new_protected:Npn \__hyp_color_set_aux:nwn #1 [#2] #3
917 {
918   \color_set:nnn {#1}{#2}{#3}
919 }
920
921 \cs_generate_variant:Nn \__hyp_color_set:nn {ne}

```

(End of 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.

```

922 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
923 {
924   \hook_gput_code:nnn
925     {pdfannot/link/#2/begin}
926     {hyp/color}
927     {
928       \bool_if:cT { l_hyp_annot_color#1_bool }
929       {
930         \group_begin:
931         \color_select:n { hyp/color/#1}
932       }
933     }
934   \hook_gput_code:nnn
935     {pdfannot/link/#2/end}
936     {hyp/color}
937     {
938       \bool_if:cT { l_hyp_annot_color#1_bool }
939       {
940         \group_end:
941       }
942     }
943 }

```

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

```

944 \keys_define:nn { hyp / setup }
945 {

```

```

946     ,colorlinks .meta:n =
947     {
948         ,pdfborder={0~0~0}
949         ,pdfborderstyle=
950         ,colorurl  =#1
951         ,colorlink  =#1
952         ,colorrun   =#1
953         ,colormenu  =#1
954         ,colorfile  =#1
955     }
956     ,colorlinks .default:n = {true}
957 }

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)
allcolors (setup key)
958 \seq_map_inline:Nn \c__hyp_annot_types_seq
959 {
960     \keys_define:nn { hyp / setup }
961     {
962         ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
963         ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
964     }
965 }
966
967 \keys_define:nn { hyp / setup }
968 {
969     ,allcolors .meta:n =
970     {
971         ,urlcolor=#1
972         ,linkcolor=#1
973         ,runcolor=#1
974         ,filecolor=#1
975         ,menucolor=#1
976     }
977     ,allcolors .value_required:n = true
978 }

```

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)
979 \keys_define:nn { hyp / setup }
980 {
981     ,bordercolormodel .choices:nn =
982     {rgb,cmyk}
983     { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
984     ,bordercolormodel .initial:n = {rgb}
985 }
986 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
987 {

```



```

988 \keys_define:nn { hyp / setup }
989 {
990   #1bordercolor .code:n =
991   {
992     \tl_if_empty:nTF { ##1 }
993     {
994       \pdfannot_dict_remove:nn
995       {link/#2}
996       { C }
997     }
998     {
999       \__hyp_color_set:ne {hyp/color/#1border}{##1}
1000       \color_export:nVN
1001       {hyp/color/#1border}
1002       \g__hyp_bordercolormodel_str
1003       \l__hyp_tmpa_tl
1004       \pdfannot_dict_put:nne
1005       {link/#2}
1006       { C }
1007       { [\l__hyp_tmpa_tl] }
1008     }
1009   }
1010 }
1011 }
1012
1013 \keys_define:nn { hyp / setup }
1014 {
1015   ,allbordercolors .meta:n =
1016   {
1017     ,linkbordercolor=#1
1018     ,urlbordercolor =#1
1019     ,filebordercolor=#1
1020     ,menubordercolor=#1
1021     ,runbordercolor =#1
1022   }
1023   ,allbordercolors .value_required:n = true
1024 }
1025

```

11.3.2 Borderwidth and -arc

```

1026 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1027 {
1028   \keys_define:nn { hyp / setup }
1029   {
1030     #1border .code:n =
1031     {
1032       \tl_if_empty:nTF { ##1 }
1033       {
1034         \pdfannot_dict_remove:nn
1035         {link/#2}
1036         { Border }
1037       }
1038     }

```

```

1039         \pdfannot_dict_put:nnn
1040         {link/#2}
1041         { Border }
1042         { [##1] }
1043     }
1044 }
1045 }
1046 }
1047 \keys_define:nn { hyp / setup }
1048 {
1049     ,pdfborder .code:n =
1050     {
1051         \tl_if_empty:nTF { #1 }
1052         {
1053             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1054             {
1055                 \pdfannot_dict_remove:nn
1056                 {link/##2}
1057                 { Border }
1058             }
1059         }
1060         {
1061             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1062             {
1063                 \pdfannot_dict_put:nnn
1064                 {link/##2}
1065                 { Border }
1066                 { [#1] }
1067             }
1068         }
1069     }
1070     ,pdfborder .initial:n = {0~0~1},
1071 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1072 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1073 {
runborderstyle (setup key) 1074     \keys_define:nn { hyp / setup }
fileborderstyle (setup key) 1075     {
menuborderstyle (setup key) 1076         #1borderstyle .code:n =
1077         {
1078             \tl_if_empty:nTF { ##1 }
1079             {
1080                 \pdfannot_dict_remove:nn
1081                 {link/#2}
1082                 { BS }
1083             }
1084             {
1085                 \pdfannot_dict_put:nnn
1086                 {link/#2}
1087                 { BS }

```

```

1088         { <<##1>> }
1089     }
1090 }
1091 }
1092 }
1093 \keys_define:nn { hyp / setup }
1094 {
1095     ,pdfborderstyle .code:n =
1096     {
1097         \tl_if_empty:nTF { #1 }
1098         {
1099             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1100             {
1101                 \pdfannot_dict_remove:nn
1102                 {link/##2}
1103                 { BS }
1104             }
1105         }
1106         {
1107             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1108             {
1109                 \pdfannot_dict_put:nnn
1110                 {link/##2}
1111                 { BS }
1112                 { <<##1>> }
1113             }
1114         }
1115     }
1116     ,pdfborderstyle .initial:n = {},
1117 }

```

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.

```

1118 \cs_new_protected:Npn \__hyp_ocg_init:
1119 {
1120     \pdf_object_new:n { __hyp/OCG/View }
1121     \pdf_object_new:n { __hyp/OCG/Print }
1122     \pdf_object_new:n { __hyp/OCG/config }
1123     \pdf_object_new:n { __hyp/OCG/refarray }
1124     \pdf_object_write:nne { __hyp/OCG/refarray } { array }
1125     {
1126         \pdf_object_ref:n { __hyp/OCG/View }
1127         \c_space_tl
1128         \pdf_object_ref:n { __hyp/OCG/Print }
1129     }
1130     \pdf_object_write:nnn { __hyp/OCG/View } { dict }
1131     {
1132         /Type/OCG
1133         /Name(View)

```

```

1134         /Usage
1135         <<
1136         /Print <</PrintState/OFF>>~
1137         /View <</ViewState/ON >>~
1138         >>
1139     }
1140     \pdf_object_write:nnn { __hyp/OCG/Print } { dict }
1141     {
1142         /Type/OCG
1143         /Name(Print)
1144         /Usage
1145         <<
1146         /Print <</PrintState/ON>>~
1147         /View <</ViewState/OFF>>~
1148         >>
1149     }
1150     \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n { __hyp/OCG/Print } }
1151     \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n { __hyp/OCG/Print } }
1152     \pdf_object_write:nne { __hyp/OCG/config } { dict }
1153     {
1154         /OFF[\pdf_object_ref:n { __hyp/OCG/Print } ]
1155         /AS[
1156             <<
1157             /Event/View
1158             /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1159             /Category[/View]
1160             >>
1161             <<
1162             /Event/Print
1163             /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1164             /Category[/Print]
1165             >>
1166             <<
1167             /Event/Export
1168             /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1169             /Category[/Print]
1170             >>
1171             ]
1172         }
1173     \pdfmanagement_add:nne { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG/Print } }
1174     \cs_gset:Npn \__hyp_ocg_init: { }
1175 }

```

(End of 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.

```

1176 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1177 {
1178     \hook_gput_code:nnn
1179     {pdfannot/link/#2/begin}
1180     {hyp/ocg}
1181     {
1182         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1183         {

```

```

1184         \__hyp_ocg_init:
1185         \group_begin:
1186         \hbox_set:Nw \l__hyp_tmpa_box
1187     }
1188 }
1189 \hook_gput_code:nnn
1190 {pdfannot/link/#2/end}
1191 {hyp/ocg}
1192 {
1193     \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1194     {
1195         \hbox_set_end:
1196         \mbox
1197         {
1198             \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1199             \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1200             \pdf_emc:
1201             \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1202             \group_begin:
1203             \color_select:n { hyp/color/#1 }
1204             \box_use_drop:N \l__hyp_tmpa_box
1205             \group_end:
1206             \pdf_emc:
1207         }
1208         \group_end:
1209     }
1210 }
1211 }

```

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

```

1212 \bool_lazy_or:nnTF
1213 { \pdf_version_compare_p:Nn > {1.4} }
1214 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1215 {
1216     \keys_define:nn { hyp / setup }
1217     {
1218         ,_ocgcolorlinks .meta:n =
1219         {
1220             ocgcolorlink=#1,
1221             ocgcolorurl=#1,
1222             ocgcolorfile=#1,
1223             ocgcolorruncolor=#1,
1224             ocgcolormenu=#1
1225         }
1226         ,_ocgcolorlinks .default:n = true
1227     }
1228 }
1229 {
1230     \keys_define:nn { hyp / setup }
1231     {
1232         ,_ocgcolorlinks .code:n =
1233         {
1234             \msg_warning:nnee
1235             { hyp }

```

```

1236         { ignore-deprecated-or-unknown-option-in-pdf-version }
1237         { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1238     }
1239 }
1240 }
1241
1242 \keys_define:nn { hyp / setup }
1243 {
1244     ,ocgcolorlinks .choice:
1245     ,ocgcolorlinks / true .meta:n =
1246     {
1247         pdfborder      = {0~0~0},
1248         pdfborderstyle = {},
1249         colorlinks     = false,
1250         _ocgcolorlinks = true
1251     }
1252     ,ocgcolorlinks / false .meta:n =
1253     {
1254         _ocgcolorlinks = false
1255     }
1256     ,ocgcolorlinks .default:n = {true}
1257 }
1258
1259 \seq_map_inline:Nn \c__hyp_annot_types_seq
1260 {
1261     \bool_lazy_or:nnTF
1262     { \pdf_version_compare_p:Nn > {1.4} }
1263     { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1264     {
1265         \keys_define:nn { hyp / setup }
1266         {
1267             ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1268         }
1269     }
1270     {
1271         \keys_define:nn { hyp / setup }
1272         {
1273             ,ocgcolor#1 .code:n=
1274             {
1275                 \msg_warning:nnee
1276                 { hyp }
1277                 { ignore-deprecated-or-unknown-option-in-pdf-version }
1278                 { ocgcolor#1 }
1279                 { \pdf_version_major:.\pdf_version_minor: }
1280             }
1281         }
1282     }
1283 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1284 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1285 {

```

```

1286 \keys_define:nn { hyp / setup }
1287 {
1288   ,#1highlight .choices:nn =
1289   { /I, /N, /O, /P}
1290   {
1291     \pdfannot_dict_put:nnn
1292     {link/#2}
1293     { H }
1294     { ##1 }
1295
1296   }
1297   ,#1highlight / .code:n =
1298   {
1299     \pdfannot_dict_remove:nn
1300     {link/#2}
1301     { H }
1302
1303   }
1304   ,#1highlight / unknown .code:n =
1305   {
1306     \msg_warning:nneee { hyp } { unknown-choice+empty }
1307     { pdfhighlight }
1308     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1309     { \exp_not:n {##1} }
1310   }
1311 }
1312 }
1313
1314
1315 \keys_define:nn { hyp / setup }
1316 {
1317   ,pdfhighlight .choices:nn =
1318   { /I, /N, /O, /P}
1319   {
1320     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1321     {
1322       \pdfannot_dict_put:nnn
1323       {link/####2}
1324       { H }
1325       { #1 }
1326     }
1327   }
1328   ,pdfhighlight / .code:n =
1329   {
1330     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1331     {
1332       \pdfannot_dict_remove:nn
1333       {link/##2}
1334       { H }
1335     }
1336   }
1337   ,pdfhighlight .initial:n = {/I},
1338   ,pdfhighlight / unknown .code:n =
1339   {

```

```

1340     \msg_warning:nneee { hyp } { unknown-choice+empty }
1341     { pdfhighlight }
1342     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1343     { \exp_not:n {#1} }
1344   }
1345 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1346 \keys_define:nn { hyp / setup }
hideurl (setup key) 1347 {
hidefile (setup key) 1348   hidelinks .meta:n =
hiderun (setup key) 1349   {
hidemenu (setup key) 1350     ,colorlinks      = false
1351     ,ocgcolorlinks = false
1352     ,pdfborder      = { 0~0~0 }
1353     ,pdfborderstyle=
1354   }
1355 }
1356
1357 \seq_map_inline:Nn \c__hyp_annot_types_seq
1358 {
1359   \keys_define:nn { hyp / setup }
1360   {
1361     hide#1 .meta:n =
1362     {
1363       ,color#1      = false
1364       ,ocgcolor#1   = false
1365       ,#1border      = { 0~0~0 }
1366       ,#1borderstyle =
1367     }
1368   }
1369 }

```

11.7 color schemes and settings

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

```

colorscheme (setup key)
1370 \keys_define:nn { hyp / setup }
1371 {
1372   colorscheme .code:n =
1373   {
1374     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1375     {
1376       \keys_set:nn { hyp /setup }
1377       {
1378         ##1 = ##2
1379       }
1380     }
1381   }

```



```

1382 }
1383 \keys_set:nn { hyp / setup } {colorscheme=phetype}

```

12 Keys

12.1 Ignored keys

The following are ignored (with or without warnings)

```

unicode (setup key)
pdfencoding (setup key) 1384 \keys_define:nn { hyp / setup }
pdfversion (setup key) 1385 {
1386   ,unicode .code:n = {}
1387   ,pdfencoding .code:n = {}
1388   ,pdfversion .code:n =
1389   {
1390     \msg_warning:nn { hyp }{ pdfversion-disabled }
1391   }
1392 }
1393 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
debug (setup key) 1394 \keys_define:nn { hyp / setup }
draft (setup key) 1395 {
1396   ,verbose .legacy_if_set:n = {Hy@verbose}
1397   ,debug .legacy_if_set:n = {Hy@verbose}
1398 }
1399 \keys_define:nn { hyp / setup }
1400 {
1401   ,draft .code:n =
1402   {
1403     \Hy@drafttrue
1404     \PassOptionsToPackage{draft}{bookmark}
1405   }
1406   ,final .code:n =
1407   {
1408     \Hy@finaltrue
1409     \PassOptionsToPackage{final}{bookmark}
1410   }
1411 }

extension (setup key)
hypertextnames (setup key) 1412 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1413 {
1414   ,extension .tl_set:N = \XR@ext
1415   ,extension .initial:n= pdf
1416   ,hypertextnames .legacy_if_set:n = {Hy@hypertextnames}
1417   ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1418   ,localanchorname .legacy_if_set:n = {Hy@localanchorname}
}

linktocpage (setup key)
plainpages (setup key)
localanchorname (setup key)
linkfileprefix (setup key)

```

```

1419 ,naturalnames .legacy_if_set:n = {Hy@naturalnames}
1420 ,pageanchor .legacy_if_set:n = {Hy@pageanchor}
1421 ,plainpages .legacy_if_set:n = {Hy@plainpages}
1422 }
1423
1424 \keys_define:nn { hyp / setup }
1425 {
1426 ,linktoc .choices:nn = { none, section, all, page }
1427 {
1428 \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1429 }
1430 ,linktoc / unknown .code:n =
1431 {
1432 \msg_warning:nneee { hyp } { unknown-choice }
1433 { linktoc }
1434 { none, section, all, page }
1435 { \exp_not:n {#1} }
1436 }
1437 ,linktocpage .choice:
1438 ,linktocpage / true .meta:n = {linktoc=page}
1439 ,linktocpage / false .meta:n = {linktoc=section}
1440 ,linktocpage .default:n = true
1441 }
1442

```

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

```

url (setup key) 1443 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1444 {
menu (setup key) 1445 \keys_define:nn { hyp / setup }
run (setup key) 1446 {
1447 ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1448 }
1449 }

```

```

1450 \keys_define:nn { hyp / setup }
1451 {
1452 ,baseurl .code:n =
1453 {
1454 \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1455 \tl_if_empty:NTF \l__hyp_tmpa_tl
1456 {
1457 \pdfmanagement_remove:nn {Catalog} { URI }
1458 }
1459 {
1460 \pdfmanagement_add:nne {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1461 }
1462 \__hyp_store_metadata:nn {baseurl}{#1}
1463 }
1464 %only false does something ...
1465 ,bookmarks .choice:
1466 ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1467 ,bookmarks / true .code:n = {}
1468 ,bookmarks .default:n = {true}
1469 ,bookmarksnumbered .legacy_if_set:n = {Hy@bookmarksnumbered}

```

```

1470 ,bookmarksopen .legacy_if_set:n = {Hy@bookmarksopen}
1471 ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1472 ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1473 ,pdfcenterwindow .choice:
1474 ,pdfcenterwindow / false .code:n =
1475 {
1476   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1477 }
1478 ,pdfcenterwindow / true .code:n =
1479 {
1480   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1481 }
1482 ,pdfcenterwindow / .code:n =
1483 {
1484   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1485 }
1486 ,pdfcenterwindow / unknown .code:n =
1487 {
1488   \msg_warning:nnee { hyp } { no-bool }
1489   { pdfcenterwindow }
1490   { \exp_not:n {#1} }
1491 }
1492 ,pdfcenterwindow .default:n = true
1493 ,pdfdirection .choice:
1494 ,pdfdirection / L2R .code:n =
1495 {
1496   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1497 }
1498 ,pdfdirection / R2L .code:n =
1499 {
1500   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1501 }
1502 ,pdfdirection / .code:n =
1503 {
1504   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1505 }
1506 ,pdfdirection / unknown .code:n =
1507 {
1508   \msg_warning:nneee { hyp } { unknown-choice+empty }
1509   { pdfdirection }
1510   { L2R , R2L }
1511   { \exp_not:n {#1} }
1512 }
1513 ,pdfdisplaydoctitle .choice:
1514 ,pdfdisplaydoctitle / false .code:n =
1515 {
1516   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1517 }
1518 ,pdfdisplaydoctitle / true .code:n =
1519 {
1520   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1521 }
1522 ,pdfdisplaydoctitle .default:n = true
1523 ,pdfduplex .choices:nn =

```

```

1524 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1525 {
1526   \pdf_version_compare:NnTF > {1.6}
1527   {
1528     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1529     { PrintDuplex } { /#1 }
1530   }
1531   {
1532     \msg_warning:nnee
1533     {hyp}
1534     {ignore-deprecated-or-unknown-option-in-pdf-version}
1535     {pdfduplex}
1536     {\pdf_version:}
1537   }
1538 }%
1539 ,pdfduplex / .code:n =
1540 {
1541   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1542 }
1543 ,pdfduplex / unknown .code:n =
1544 {
1545   \msg_warning:nneee { hyp } { unknown-choice+empty }
1546   { pdfduplex }
1547   { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1548   { \exp_not:n {#1} }
1549 }
1550 ,pdffitwindow .choice:
1551 ,pdffitwindow / false .code:n =
1552 {
1553   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1554 }
1555 ,pdffitwindow / true .code:n =
1556 {
1557   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1558 }
1559 ,pdffitwindow / .code:n =
1560 {
1561   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1562 }
1563 ,pdffitwindow .default:n = true
1564 ,pdffitwindow / unknown .code:n =
1565 {
1566   \msg_warning:nnee { hyp } { no-bool }
1567   { pdffitwindow }
1568   { \exp_not:n {#1} }
1569 }
1570 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1571 ,pdflinkmargin .initial:n = {1pt}
1572 ,pdfmenubar .choice:
1573 ,pdfmenubar / true .code:n =
1574 {
1575   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1576 }
1577 ,pdfmenubar / false .code:n =

```

```

1578     {
1579         \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1580         { HideMenubar } { true }
1581     }
1582 ,pdfmenubar / .code:n =
1583     {
1584         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1585     }
1586 ,pdfmenubar .default:n = true
1587 ,pdfmenubar / unknown .code:n =
1588     {
1589         \msg_warning:nnee { hyp } { no-bool }
1590         { pdfmenubar }
1591         { \exp_not:n {#1} }
1592     }
1593 ,pdfnewwindow .choice:
1594 ,pdfnewwindow / true .code:n =
1595     {
1596         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1597         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1598     }
1599 ,pdfnewwindow / false .code:n =
1600     {
1601         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1602         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1603     }
1604 ,pdfnewwindow / .code:n =
1605     {
1606         \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1607         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1608     }
1609 ,pdfnonfullscreenpagemode .choices:nn =
1610 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1611 {
1612     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1613     { NonFullScreenPageMode } {/#1}
1614 }
1615 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1616 {
1617     \pdf_version_compare:NnTF < {1.6}
1618     {
1619         %message
1620     }
1621     {
1622         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1623         {NonFullScreenPageMode}{/UseAttachments}
1624     }
1625 }
1626 ,pdfnonfullscreenpagemode / .code:n =
1627 {
1628     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1629 }
1630 ,pdfnonfullscreenpagemode / unknown .code:n =
1631 {

```

```

1632     \msg_warning:nneee { hyp } { unknown-choice+empty }
1633     { pdfnonfullscreenpagemode }
1634     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1635     { \exp_not:n {#1} }
1636 }
1637 ,pdfnumcopies .code:n =
1638 {
1639     \pdf_version_compare:NnTF > {1.6}
1640     {
1641         \tl_if_empty:nTF {#1}
1642         {
1643             \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1644         }
1645         {
1646             \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1647             {NumCopies}{#1}
1648         }
1649     }
1650     {
1651         \msg_warning:nnee
1652         {hyp}
1653         {ignore-deprecated-or-unknown-option-in-pdf-version}
1654         {pdfnumcopies}
1655         {\pdf_version:}
1656     }
1657 }
1658 ,pdfpagelayout .choices:nn =
1659 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1660 { \pdfmanagement_add:nne {Catalog} { PageLayout }{ /#1 } }
1661 ,pdfpagelayout / .code:n =
1662 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1663 ,pdfpagelayout / unknown .code:n =
1664 {
1665     \msg_warning:nneee { hyp } { unknown-choice+empty }
1666     { pdfpagelayout }
1667     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1668     { \exp_not:n {#1} }
1669 }
1670 ,pdfpagemode .choices:nn =
1671 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1672 { \pdfmanagement_add:nne {Catalog} { PageMode }{ /#1 } }
1673 ,pdfpagemode / UseAttachments .code:n =
1674 {
1675     \pdf_version_compare:NnTF > {1.5}
1676     {
1677         \pdfmanagement_add:nne {Catalog} { PageMode }{ /UseAttachments }
1678     }
1679     {
1680         \msg_warning:nnee
1681         {hyp}
1682         {ignore-deprecated-or-unknown-value-in-pdf-version}
1683         {UseAttachments}
1684         {\pdf_version:}
1685     }
1686 }

```

```

1686     }
1687 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1688 ,pdfpagemode / unknown .code:n =
1689 {
1690     \msg_warning:nneee { hyp } { unknown-choice+empty }
1691     { pdfpagemode }
1692     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1693     { \exp_not:n {#1} }
1694 }
1695 ,pdfpagescrop .code:n =
1696 {
1697     \tl_if_empty:nTF {#1} %or blank?
1698     {
1699         \pdfmanagement_remove:nn {Pages} { CropBox }
1700     }
1701     {
1702         \pdfmanagement_add:nne {Pages} { CropBox } { [#1] }
1703     }
1704 }
1705 ,pdfpicktraybypdfsize .choice:
1706 ,pdfpicktraybypdfsize / true .code:n =
1707 {
1708     \pdf_version_compare:NnTF > {1.6}
1709     {
1710         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1711         { PickTrayByPDFSize } { true }
1712     }
1713     {
1714         \msg_warning:nnee
1715         {hyp}
1716         {ignore-deprecated-or-unknown-option-in-pdf-version}
1717         {pdfpicktraybypdfsize}
1718         {\pdf_version:}
1719     }
1720 }
1721 ,pdfpicktraybypdfsize / false .code:n =
1722 {
1723     \pdf_version_compare:NnTF > {1.6}
1724     {
1725         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1726         { PickTrayByPDFSize } { false }
1727     }
1728     {
1729         \msg_warning:nnee
1730         {hyp}
1731         {ignore-deprecated-or-unknown-option-in-pdf-version}
1732         {pdfpicktraybypdfsize}
1733         {\pdf_version:}
1734     }
1735 }
1736 ,pdfpicktraybypdfsize / .code:n =
1737 {
1738     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1739 }

```

```

1740 ,pdfpicktraybypdfsize / unknown .code:n =
1741 {
1742     \msg_warning:nnee { hyp } { no-bool }
1743     { picktraybypdfsize }
1744     { \exp_not:n {#1} }
1745 }
1746 ,pdfprintarea .choices:nn =
1747 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1748 {
1749     \pdf_version_compare:NnTF < {2.0}
1750     {
1751         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1752         { PrintArea } { /#1 }
1753     }
1754     {
1755         \msg_warning:nnee
1756         {hyp}
1757         {ignore-deprecated-or-unknown-option-in-pdf-version}
1758         {pdfprintarea}
1759         {\pdf_version:}
1760     }
1761 }%
1762 ,pdfprintarea / .code:n =
1763 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1764 ,pdfprintarea / unknown .code:n =
1765 {
1766     \msg_warning:nneee { hyp } { unknown-choice+empty }
1767     { pdfprintarea }
1768     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1769     { \exp_not:n {#1} }
1770 }
1771 ,pdfprintclip .choices:nn =
1772 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1773 {
1774     \pdf_version_compare:NnTF < {2.0}
1775     {
1776         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1777         { PrintClip } { /#1 }
1778     }
1779     {
1780         \msg_warning:nnee
1781         {hyp}
1782         {ignore-deprecated-or-unknown-option-in-pdf-version}
1783         {pdfprintclip}
1784         {\pdf_version:}
1785     }
1786 }%
1787 ,pdfprintclip / .code:n =
1788 {
1789     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1790 }
1791 ,pdfprintclip / unknown .code:n =
1792 {
1793     \msg_warning:nneee

```



```

1794     { hyp }
1795     { unknown-choice+empty }
1796     { pdfprintclip }
1797     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1798     { \exp_not:n {#1} }
1799   }
1800 ,pdfprintpagerange .code:n =
1801   {
1802     \pdf_version_compare:NnTF > {1.6}
1803     {
1804       \tl_if_empty:nTF { #1}
1805       {
1806         \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1807         { PrintPageRange }
1808       }
1809       {
1810         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1811         {PrintPageRange}{[#1]}
1812       }
1813     }
1814     {
1815       \msg_warning:nnee
1816       {hyp}
1817       {ignore-deprecated-or-unknown-option-in-pdf-version}
1818       {pdfprintpagerange}
1819       {\pdf_version:}
1820     }
1821   }
1822 ,pdfprintscaling .choices:nn =
1823   { None, AppDefault }
1824   {
1825     \pdf_version_compare:NnTF > {1.5}
1826     {
1827       \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1828       { PrintScaling } { /#1 }
1829     }
1830     {
1831       \msg_warning:nnee
1832       {hyp}
1833       {ignore-deprecated-or-unknown-option-in-pdf-version}
1834       {pdfprintscaling}
1835       {\pdf_version:}
1836     }
1837   }%
1838 ,pdfprintscaling / .code:n =
1839   {
1840     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1841   }
1842 ,pdfprintscaling / unknown .code:n =
1843   {
1844     \msg_warning:nneee { hyp } { unknown-choice+empty }
1845     { pdfprintarea }
1846     { None, AppDefault }
1847     { \exp_not:n {#1} }

```

```

1848     }
1849 ,pdfremotestartview .code:n =
1850 {
1851     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
1852     \exp_args:NNV
1853     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1854     {
1855         \tl_set:Ne \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1856     }
1857     {
1858         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1859         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1860     }
1861 }
1862 ,pdfremotestartview .initial:n = {Fit}
1863 % pdfstartpage is special as it shares code with pdfstartview
1864 ,pdfstartpage .code:n =
1865 {
1866     \tl_gset:Ne \g__hyp_dest_pdfstartpage_tl { #1 }
1867     \bool_if:nTF
1868     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1869     {
1870         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1871     }
1872     {
1873         \pdfmanagement_add:nne {Catalog} { OpenAction }
1874         {
1875             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1876         }
1877     }
1878 }
1879 ,pdfstartpage .initial:n =1
1880 ,pdfstartview .code:n =
1881 {
1882     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
1883     \exp_args:NNV
1884     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1885     {
1886         \tl_gset:Ne \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1887     }
1888     {
1889         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1890         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1891     }
1892     \bool_if:nTF
1893     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1894     {
1895         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1896     }
1897     {
1898         \pdfmanagement_add:nne {Catalog} { OpenAction }
1899         {
1900             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1901         }

```

```

1902     }
1903 }
1904 ,pdfstartview .initial:n = Fit
1905 ,pdftoolbar .choice:
1906 ,pdftoolbar / true .code:n =
1907 {
1908     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1909 }
1910 ,pdftoolbar / false .code:n =
1911 {
1912     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1913     { HideToolbar } { true }
1914 }
1915 ,pdftoolbar / true .code:n =
1916 {
1917     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1918 }
1919 ,pdftoolbar .default:n = true
1920 ,pdftoolbar / unknown .code:n =
1921 {
1922     \msg_warning:nnee { hyp } { no-bool }
1923     { pdftoolbar }
1924     { \exp_not:n {#1} }
1925 }
1926 % pdfview see below.
1927 ,pdfviewarea .choices:nn =
1928 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1929 {
1930     \pdf_version_compare:NnTF < {2.0}
1931     {
1932         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1933         { ViewArea } { /#1 }
1934     }
1935     {
1936         \msg_warning:nnee
1937         {hyp}
1938         {ignore-deprecated-or-unknown-option-in-pdf-version}
1939         {pdfviewarea}
1940         {\pdf_version:}
1941     }
1942 }%
1943 ,pdfviewarea / .code:n =
1944 {
1945     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1946 }
1947 ,pdfviewarea / unknown .code:n =
1948 {
1949     \msg_warning:nnee { hyp } { unknown-choice+empty }
1950     { pdfviewarea }
1951     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1952     { \exp_not:n {#1} }
1953 }
1954 ,pdfviewclip .choices:nn =
1955 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }

```

```

1956 {
1957   \pdf_version_compare:NnTF < {2.0}
1958   {
1959     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1960     { ViewClip } { /#1 }
1961   }
1962   {
1963     \msg_warning:nnee
1964     {hyp}
1965     {ignore-deprecated-or-unknown-option-in-pdf-version}
1966     {pdfviewclip}
1967     {\pdf_version:}
1968   }
1969   }%
1970 ,pdfviewclip / .code:n =
1971 {
1972   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
1973 }
1974 ,pdfviewclip / unknown .code:n =
1975 {
1976   \msg_warning:nneee { hyp } { unknown-choice+empty }
1977   { pdfviewclip }
1978   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1979   { \exp_not:n {#1} }
1980 }
1981 ,pdfwindowui .choice:
1982 ,pdfwindowui / true .code:n =
1983 {
1984   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
1985 }
1986 ,pdfwindowui / false .code:n =
1987 {
1988   \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1989   { HideWindowUI } { true }
1990 }
1991 ,pdfwindowui / .code:n =
1992 {
1993   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
1994 }
1995 ,pdfwindowui / unknown .code:n =
1996 {
1997   \msg_warning:nnee { hyp } { no-bool }
1998   { pdfwindowui }
1999   { \exp_not:n {#1} }
2000 }
2001 ,pdfwindowui .default:n = true
2002 }

```

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

```

2003 \keys_define:nn { hyp / setup }
2004 {
2005   ,pdfview .code:n =
2006   {
2007     \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}

```

```

2008 \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
2009 {
2010   { xyz }
2011   {
2012     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
2013     {
2014       \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2015       \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2016       {
2017         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2018       }
2019       {
2020         \tl_set:Ne \l__hyp_dest_pdfview_tl
2021         {
2022           \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2023         }
2024       }
2025     }
2026     {
2027       \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2028     }
2029   }
2030   { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2031   { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2032   { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2033   { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2034   { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2035   { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2036   { fitr }
2037   {
2038     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2039     {
2040       \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2041     }
2042     {
2043       %ensure 4 values ...
2044       \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2045       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2046       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2047       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2048       \hbox_set_to_wd:Nnn \l__hyp_dest_box
2049       {
2050         \fp_eval:n
2051         {
2052           round
2053           (
2054             abs
2055             (
2056               \seq_item:Nn\l__hyp_tmpa_seq{4}
2057               -
2058               (\seq_item:Nn\l__hyp_tmpa_seq{2})
2059             ),
2060             3
2061           )

```

```

2062         }bp
2063     }{}
2064     \box_set_dp:Nn \l__hyp_dest_box
2065     {
2066         \fp_eval:n
2067         {
2068             round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2069         }bp
2070     }
2071     \box_set_ht:Nn \l__hyp_dest_box
2072     {
2073         \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2074     }
2075 }
2076 }
2077 }
2078 {
2079     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2080     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2081 }
2082 }
2083 ,pdfview .initial:n = {xyz}
2084 }

```

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.

```

2085 \keys_define:nn { hyp / setup }
2086 {
2087     ,pdflang .code:n =
2088     {
2089         \tl_if_empty:nF { #1 }
2090         {
2091             \pdfmanagement_add:nne {Catalog} { Lang } { (#1) }
2092             \AddToDocumentProperties[document]{lang}{#1}
2093         }
2094     }
2095 }

```

12.3.1 “info keys”

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

pdftitle (*setup key*) Creator and Producer can’t be removed with the pdfmanagement, but we allow to set an

pdfcreator (*setup key*) empty value. If the value begin with an optional argument, we assume a multilanguage

pdfsubject (*setup key*) clist and use only the first value.

pdfproducer (*setup key*)

pdfkeywords (*setup key*)

```

2096 \regex_new:N\l__hyp_optlang_regex
2097 \regex_set:Nn\l__hyp_optlang_regex {\A\[([A-Za-z\-\-])\](.*)}
2098 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2099 {
2100     \keys_define:nn { hyp / setup }

```

```

2101 {
2102   pdf#1 .code:n =
2103   {
2104     \tl_if_blank:nTF {##1}
2105     {
2106       \str_case:nnF { #1 }
2107       {
2108         {creator}
2109         {
2110           \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2111           \pdfmanagement_add:nne {Info}{Creator}{()}
2112         }
2113         {producer}
2114         {
2115           \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2116           \pdfmanagement_add:nne {Info}{Producer}{()}
2117         }
2118       }
2119       {
2120         \pdfmanagement_remove:nn {Info}{#2}
2121       }
2122     }
2123     {
2124       \tl_set:Ne\l__hyp_tmpa_tl {\clist_item:nn{##1}{1}}
2125       \exp_args:NNV
2126       \regex_extract_once:NnN \l__hyp_optlang_regex \l__hyp_tmpa_tl\l__hyp_tmpa_s
2127       \seq_if_empty:NTF\l__hyp_tmpa_seq
2128       {
2129         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2130       }
2131       {
2132         \exp_args:Ne
2133         \__hyp_text_pdfstring_info:nN {\seq_item:Nn \l__hyp_tmpa_seq{3}}\l__hyp_t
2134       }
2135       \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2136       {
2137         \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2138       }
2139     }
2140     \__hyp_store_metadata:nn {pdf#1}{##1}
2141   }
2142 }
2143 \keys_define:nn { hyp / info }
2144 {
2145   #2 .code:n =
2146   {
2147     \tl_if_blank:nTF {##1}
2148     {
2149       \pdfmanagement_remove:nn {Info}{#2}
2150     }
2151     {
2152       \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2153       \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2154       {

```

```

2155         \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2156     }
2157 }
2158 \exp_args:Ne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2159 }
2160 ,unknown .code:n =
2161 {
2162     \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2163     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2164     {
2165         \exp_args:Nno
2166         \pdfmanagement_add:nne {Info}
2167         { \l_keys_key_str } {\l__hyp_tmpa_str}
2168     }
2169 }
2170 }
2171 }
2172 \__hyp_setup_info_key:nn {author} {Author}
2173 \__hyp_setup_info_key:nn {title} {Title}
2174 \__hyp_setup_info_key:nn {producer} {Producer}
2175 \__hyp_setup_info_key:nn {creator} {Creator}
2176 % ignored key: addtopdfcreator
2177 \__hyp_setup_info_key:nn {subject} {Subject}
2178 \__hyp_setup_info_key:nn {keywords} {Keywords}

```

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

```

pdfmetadate (setup key) 2179 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2180 {
2181     \keys_define:nn { hyp / setup }
2182     {
2183         pdf#1 .code:n =
2184         {
2185             \tl_if_blank:nTF {##1}
2186             {
2187                 \pdfmanagement_remove:nn {Info}{#2}
2188             }
2189             {
2190                 \pdfmanagement_add:nne {Info}{#2}{(##1)}
2191             }
2192             \__hyp_store_metadata:nn {pdf#1}{##1}
2193             \AddToDocumentProperties[document]{#1}{##1}
2194         }
2195     }
2196     \keys_define:nn { hyp / info }
2197     {
2198         #2 .code:n =
2199         {
2200             \tl_if_blank:nTF {##1}
2201             {
2202                 \pdfmanagement_remove:nn {Info}{#2}
2203             }
2204             {
2205                 \pdfmanagement_add:nne {Info}{#2}{(##1)}

```



```

2206     }
2207     \exp_args:Ne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2208   }
2209 }
2210 }
2211
2212 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2213 \__hyp_setup_info_date_key:nn {moddate} {ModDate}
2214 \keys_define:nn { hyp / setup }
2215 {
2216     pdfmetadate .code:n = { \__hyp_store_metadata:nn {pdfmetadate}{#1} }
2217 }

```

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

```

2218 \keys_define:nn { hyp / setup }
2219 {
2220     ,pdftrapped .code:n =
2221     {
2222         \exp_args:Nne
2223         \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2224     }
2225     ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2226     {
2227         \pdfmanagement_add:nne {Info}{Trapped}
2228         {/
2229             \str_uppercase:f { \str_head:n { #1 } }
2230             \str_lowercase:f { \str_tail:n { #1 } }
2231         }
2232         \__hyp_store_metadata:ne {pdftrapped}
2233         {
2234             \str_uppercase:f { \str_head:n { #1 } }
2235             \str_lowercase:f { \str_tail:n { #1 } }
2236         }
2237     }
2238     ,_pdftrapped / unknown .code:n =
2239     {
2240         \msg_warning:nneee { hyp } { unknown-choice }
2241         { pdftrapped }
2242         { true~(case~insensitive), false~(case~insensitive), unknown~(case~insensitive) }
2243         { \exp_not:n {#1} }
2244     }
2245 }

```

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

```

2246 \keys_define:nn { hyp / setup }
2247 {
2248     pdfinfo .code:n =
2249     {
2250         \keys_set:nn { hyp / info } { #1 }
2251     }
2252 }

```

Now we set some default values

```

2253 \keys_set:nn { hyp / setup } {pdfcreator = LaTeX-with-hyperref}

```

```

2254 \keys_set:nn { hyp / setup} {pdfauthor = }
2255 \keys_set:nn { hyp / setup} {pdftitle = }
2256 \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 most of them, but they are only stored as metadata:

```

2257 \clist_map_inline:nn
2258 {
2259   ,pdfcopyright
2260   ,pdftype
2261   ,pdflicenseurl
2262   ,pdfauthortitle
2263   ,pdfcaptionwriter
2264   ,pdfmetalang
2265   ,pdfsource
2266   ,pdfdocumentid
2267   ,pdfinstanceid
2268   ,pdfversionid
2269   ,pdfrendition
2270   ,pdfpublication
2271   ,pdfpubtype
2272   ,pdfbytes
2273   ,pdfnumpages
2274   ,pdfissn
2275   ,pdfeissn
2276   ,pdfisbn
2277   ,pdfbookedition
2278   ,pdfpublisher
2279   ,pdfvolumenum
2280   ,pdfissuenum
2281   ,pdfpagerange
2282   ,pdfdoi
2283   ,pdfurl
2284   ,pdfidentifier
2285   ,pdfsubtitle
2286   ,pdfpubstatus
2287   ,pdfcontactaddress
2288   ,pdfcontactcity
2289   ,pdfcontactregion
2290   ,pdfcontactpostcode
2291   ,pdfcontactcountry
2292   ,pdfcontactphone
2293   ,pdfcontactemail
2294   ,pdfcontacturl
2295   ,pdfdate
2296 }
2297 {
2298   \keys_define:nn { hyp / setup }
2299   {
2300     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2301   }

```

```

2302 }
2303

```

12.5 Transitions

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

```

2304 \keys_define:nn { hyp / setup }
2305 {
2306   pdfpageduration .code:n =
2307   {
2308     \tl_if_blank:nTF { #1 }
2309     {
2310       \pdfmanagement_remove:nn {Page}{Dur}
2311     }
2312     {
2313       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2314     }
2315   }
2316 }

```

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

<code>style</code>	one of Split, Blinds, Box, Wipe, Dissolve, Glitter, R, Fly, Push, Cover, Uncover, Fade
<code>duration</code>	a number, describes the duration of the transition
<code>direction</code>	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)
<code>motion</code>	one of I, O, only relevant for Split, Box and Fly
<code>scale</code>	a number, only relevant for Fly style
<code>opaque</code>	true or false, only relevant for Fly style

```

2317 \keys_define:nn { hyp / setup }
2318 {
2319   pdfpagetransition .code:n =
2320   {
2321     \tl_if_blank:nTF {#1}

```

```

2322     {
2323         \pdfmanagement_remove:nn {Page}{Trans}
2324     }
2325     {
2326         \group_begin:
2327         \keys_set:nn { hyp / trans }{style=R,#1}
2328         \pdf_object_unnamed_write:ne { dict }
2329         {
2330             \pdfdict_use:n {l__hyp_page/Trans}
2331         }
2332         \pdfmanagement_add:nne {Page}{Trans}{\pdf_object_ref_last:}
2333         \group_end:
2334     }
2335 }
2336 }
2337 \keys_define:nn { hyp / trans }
2338 {
2339     ,style .choices:nn =
2340     {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2341     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2342     ,style / unknown .code:n =
2343     {
2344         \msg_warning:nneee { hyp } { unknown-choice }
2345         { trans / style }
2346         { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2347         { \exp_not:n {#1} }
2348     }
2349     ,duration .code:n =
2350     {
2351         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}
2352     }
2353     ,direction .choices:nn =
2354     {H,V}
2355     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2356     ,direction .choices:nn =
2357     {0,90,180,270,315}
2358     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2359     ,direction / None .code:n =
2360     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2361     ,direction / unknown .code:n =
2362     {
2363         \msg_warning:nneee { hyp } { unknown-choice }
2364         { trans / direction }
2365         {
2366             H~(horizontal,~only~Split,~Blinds),
2367             V~(vertical,~only~Split,~Blinds),
2368             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2369             90~(bottom~to~top,~only~Wipe),
2370             180~(right~to~left,~only~Wipe),
2371             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2372             315~(top~left~to~bottom,~only~Glitter),
2373             None~(only~Fly)
2374         }
2375         { \exp_not:n {#1} }

```

```

2376     }
2377     ,motion .choices:nn =
2378     {I,0}
2379     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2380     ,motion / unknown .code:n =
2381     {
2382         \msg_warning:nneee { hyp } { unknown-choice }
2383         { trans / motion }
2384         { I~(inwards) , 0~(outwards) }
2385         { \exp_not:n {#1} }
2386     }
2387     ,scale .code:n =
2388     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2389     ,opaque .choices:nn = {true,false}
2390     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2391     ,opaque / unknown .code:n =
2392     {
2393         \msg_warning:nneee { hyp } { unknown-choice }
2394         { trans / B }
2395         { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2396         { \exp_not:n {#1} }
2397     }
2398     % try to set unknown keys as style
2399     ,unknown .code:n =
2400     {
2401         % warning ...
2402         \exp_args:Nne\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2403     }
2404 }

```

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

```

2405 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}

```

Unfinished Form field code

```

2406 \NewDocumentCommand \MakeFieldObject { m m }
2407 {
2408     \pdfxform_new:nnn { #2 }{ } { #1 }
2409 }
2410
2411
2412 \prop_new:N    \g__hyp_AcroForm_CoFields_prop
2413 \prop_new:N    \g__hyp_AcroForm_Fields_prop
2414
2415 \let\HyField@afields\@empty
2416 \let\HyField@cofields\@empty
2417 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2418
2419 % the value doesn't matter, but with a prop we avoid duplicates and it is
2420 % clearly faster than removing them from a sequence
2421 \def\HyField@AuxAddToFields#1
2422 {
2423     \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2424 }%
2425
2426 %fields with empty key get a value too -- lets hope that

```

```

2427 %this give the expected behaviour
2428 \def\HyField@AuxAddToCoFields #1 #2
2429 {
2430   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2431 }
2432
2433 \Hy@AtBeginDocument
2434 {
2435   \if@filesw
2436     \immediate\write\@mainaux{%
2437       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2438     }%
2439     \immediate\write\@mainaux{%
2440       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2441     }%
2442   \fi
2443   \let\HyField@AfterAuxOpen\@firstofone
2444 }%
2445
2446 \def\HyField@AddToFields
2447 {
2448   \exp_args:Ne\HyField__hypAddToFields
2449   {
2450     \pdfannot_box_ref_last:
2451   }
2452   \ifx\Fld@calculate@code\@empty
2453   \else
2454     \begingroup
2455       \Hy@safe@activetrue
2456       \edef\Hy@temp{%
2457         \endgroup
2458         \if@filesw
2459           \write\@mainaux
2460           {
2461             \string\HyField@AuxAddToCoFields
2462             {
2463               \Fld@calculate@sortkey
2464             }
2465             {
2466               \pdfannot_box_ref_last:
2467             }
2468           }
2469         \fi
2470       }%
2471     \Hy@temp
2472   \fi
2473 }%
2474
2475 \def\HyField__hypAddToFields#1{
2476   \HyField@AfterAuxOpen{%
2477     \if@filesw
2478       \write\@mainaux{%
2479         \string\HyField@AuxAddToFields{#1}%
2480       }%

```

```

2481     \fi
2482   }%
2483 }%
2484
2485 \ExplSyntaxOff
2486 \ExplSyntaxOn
2487
2488 \def\@Form[#1]
2489 {
2490   \kvsetkeys{Form}{#1}
2491   \pdf@ifdraftmode{}
2492   {
2493     \Hy@FormObjects
2494     \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2495     {
2496       \pdfmanagement_add:nne { Catalog / AcroForm } { Fields }{##1}
2497       %\pdfmanagement_show:n { Catalog / AcroForm }
2498     }
2499     \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2500     {
2501       \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2502       {
2503         \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2504       }
2505       \seq_sort:Nn \l__hyp_tmpa_seq
2506       {
2507         \str_compare:nNnTF {##1} > {##2}
2508         { \sort_return_swapped: }
2509         { \sort_return_same: }
2510       }
2511       \seq_map_inline:Nn \l__hyp_tmpa_seq
2512       {
2513         \pdfmanagement_add:nne { Catalog / AcroForm }
2514         { CO }
2515         {
2516           \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2517         }
2518       }
2519     }
2520     \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2521     {ZaDb} {\pdf_object_ref:n {__hyp/Font/ZaDb} }
2522     \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2523     {Helv} {\pdf_object_ref:n {__hyp/Font/Helv} }
2524     \pdfmanagement_add:nne {Catalog /AcroForm}
2525     {DA}{(/Helv~10~Tf~0~g)}
2526     \pdfmeta_standard_verify:nTF {form_no_NeedAppearance}
2527     {
2528       \legacy_if:nT { HyField@NeedAppearances }
2529       {
2530         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2531       }
2532     }
2533     {
2534       \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}

```

```

2535     }
2536   }
2537 }
2538 \ExplSyntaxOff
2539 \let\@endForm\@empty
2540 \let\HyAnn@AbsPageLabel\@empty
2541 \let\Fld@pageobjref\@empty
2542
2543 \ExplSyntaxOn
2544 \newcount\HyAnn@Count
2545 \HyAnn@Count=\z@
2546 \def\HyAnn@AbsPageLabel
2547 {
2548   \global\advance\HyAnn@Count by\@ne
2549   %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2550   %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2551   %\zref@refused{HyAnn@\the\HyAnn@Count}%
2552   \__hyp_property_record:ee {HyAnn@\the\HyAnn@Count}{abspage}
2553   \property_ref_undefined_warn:ee {HyAnn@\the\HyAnn@Count}{abspage}
2554 }%
2555 \prg_generate_conditional_variant:Nnn \property_if_recorded:nn {ee} {T}
2556 \def\Fld@pageobjref
2557 {
2558   \property_if_recorded:eeT {HyAnn@\the\HyAnn@Count}{abspage}
2559   {
2560     /P~\pdf_pageobject_ref:e
2561     {
2562       \property_ref:ee{HyAnn@\the\HyAnn@Count}{abspage}
2563     }
2564   }
2565 }
2566 \ExplSyntaxOff
2567 \ExplSyntaxOn
2568 %% check if the attr should be set through
2569 %% hooks.
2570 %% check if options are missing.
2571 \def\@TextField[#1]#2{% parameters, label
2572   \def\Fld@name{#2}%
2573   \let\Fld@default\@empty
2574   \let\Fld@value\@empty
2575   \def\Fld@width{\DefaultWidthofText}%
2576   \def\Fld@height{%
2577     \ifFld@multiline
2578       \DefaultHeightofTextMultiline
2579     \else
2580       \DefaultHeightofText
2581     \fi
2582   }%
2583   \begingroup
2584     \expandafter\HyField@SetKeys\expandafter{%
2585       \DefaultOptionsofText,#1%
2586     }%
2587     \PDFForm@Name
2588     \HyField@FlagsText

```



```

2589 \ifFld@hidden\def\Fld@width{1sp}\fi
2590 \ifx\Fld@value\empty\def\Fld@value{\Fld@default}\fi
2591 \LayoutTextField{#2}{%
2592 \leavevmode
2593 \HyAnn@AbsPageLabel
2594 \Hy@escapeform\PDFForm@Text
2595 \pdfannot_box:nnnn
2596 {\Fld@width}
2597 {\Fld@height}
2598 {Opt} %is this correct?
2599 {\PDFForm@Text}
2600 \MakeTextField{\Fld@width}{\Fld@height}
2601 \HyField@AddToFields
2602 }%
2603 \endgroup
2604 }
2605 \providecommand\@curropt{}
2606 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2607 \def\Fld@name{#2}
2608 \let\Fld@default\relax
2609 \let\Fld@value\relax
2610 \def\Fld@width{\DefaultWidthofChoiceMenu}
2611 \def\Fld@height{\DefaultHeightofChoiceMenu}
2612 \begingroup
2613 \Fld@menulength=0 %
2614 \@tempdima\z@
2615 \clist_map_variable:nNn { #3 } \@curropt
2616 %\@for\@curropt:=#3\do
2617 {%
2618 \expandafter\Fld@checkequals\@curropt==\%
2619 \Hy@StepCount\Fld@menulength
2620 \settowidth{\@tempdimb}{\@currDisplay}%
2621 \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2622 }%
2623 \advance\@tempdima by~15\p@
2624 \begingroup
2625 \HyField@SetKeys{#1}
2626 \edef\x{\endgroup
2627 \noexpand\expandafter
2628 \noexpand\HyField@SetKeys
2629 \noexpand\expandafter{%
2630 \expandafter\noexpand\csname DefaultOptionsof%
2631 \ifFld@radio
2632 Radio%
2633 \else
2634 \ifFld@combo
2635 \ifFld@popdown
2636 PopdownBox%
2637 \else
2638 ComboBox%
2639 \fi
2640 \else
2641 ListBox%
2642 \fi

```

```

2643         \fi
2644     \endcsname
2645 }%
2646 }\x
2647 \HyField@SetKeys{#1}%
2648 \PDFForm@Name
2649 \ifFld@hidden\def\Fld@width{1sp}\fi
2650 \ifx\Fld@value\relax
2651     \let\Fld@value\Fld@default
2652 \fi
2653 \LayoutChoiceField{#2}{%
2654     \ifFld@radio
2655         \HyField@FlagsRadioButton
2656         \__hypRadio{#3}%
2657     \else
2658         \begingroup
2659             \HyField@FlagsChoice
2660             \ifdim\Fld@width<\@tempdima
2661                 \ifdim\@tempdima<1cm\@tempdima1cm\fi
2662                 \edef\Fld@width{\the\@tempdima}%
2663             \fi
2664             \ifFld@combo
2665             \else
2666                 \@tempdima=\the\Fld@menulength\Fld@charsize
2667                 \advance\@tempdima by~\Fld@borderwidth bp %
2668                 \advance\@tempdima by~\Fld@borderwidth bp %
2669                 \edef\Fld@height{\the\@tempdima}%
2670             \fi
2671             \__hypListbox{#3}%
2672         \endgroup
2673     \fi
2674 }%
2675 \endgroup
2676 }
2677
2678 \def\__hypRadio#1{%
2679     \Fld@listcount=0~%
2680     %\show\Fld@default
2681     \EdefEscapeName\Fld@default{\Fld@default}%
2682     \clist_map_variable:nNn { #1 } \@curropt
2683     %\@for\@curropt:=#1\do
2684     {%
2685         \expandafter\Fld@checkequals\@curropt==\%
2686         \EdefEscapeName\@currValue{\@currValue}%
2687         \Hy@StepCount\Fld@listcount
2688         \@currDisplay\space
2689         \leavevmode
2690         \HyAnn@AbsPageLabel
2691         \Hy@escapeform\PDFForm@Radio
2692         \pdfxform_if_exist:nF { __hyp_xform_Ding }
2693         {
2694             \pdfxform_new:nnn { __hyp_xform_Ding } {}
2695             {
2696                 \group_begin:

```

```

2697         \fontfamily{pzd}
2698         \fontencoding{U}
2699         \fontseries{m}
2700         \fontshape{n}
2701         \selectfont
2702         \char123
2703         \group_end:
2704     }
2705 }
2706 \pdfannot_box:nnne
2707     {\Fld@width}
2708     {\Fld@height}
2709     {Opt} %is this correct?
2710     {
2711         \PDFForm@Radio
2712         /AP
2713         <<
2714         /N
2715         <<
2716         /@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2717         %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2718         >>
2719         >>
2720     }
2721     {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2722     \int_compare:nNnT { \Fld@listcount} = { 1 }
2723     { \HyField@AddToFields }
2724     \c_space_tl % deliberate space between radio buttons
2725     % to do: --> should be configurable
2726 }%
2727 }
2728
2729 \newcount\Fld@listcount
2730 \def\__hypListbox#1
2731 {
2732     \HyField@PDFChoices{#1}
2733     \mode_leave_vertical:
2734     \HyAnn@AbsPageLabel
2735     \Hy@escapeform\PDFForm@List
2736     \pdf_link_user:nnn
2737         {widget} %perhaps we need more types??
2738         {\PDFForm@List}
2739         {\MakeChoiceField{\Fld@width}{\Fld@height}}
2740     \HyField@AddToFields
2741 }
2742
2743
2744 \def\@PushButton[#1]#2{% parameters, label
2745     \def\Fld@name{#2}%
2746     \group_begin:
2747         \exp_args:No\HyField@SetKeys
2748         {
2749             \DefaultOptionsofPushButton,#1
2750         }

```

```

2751 \PDFForm@Name
2752 \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2753 {
2754   \HyField@FlagsPushButton
2755   \legacy_if:nT {Fld@hidden}
2756   {
2757     \def\Fld@width{1sp}
2758   }
2759   \LayoutPushButtonField
2760   {
2761     \mode_leave_vertical:
2762     \HyAnn@AbsPageLabel
2763     \Hy@escapeform\PDFForm@Push
2764     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2765     \pdfannot_box:nnnn
2766     {\box_wd:N\l_tmpa_box}
2767     {\box_ht:N\l_tmpa_box}
2768     {\box_dp:N\l_tmpa_box} %is this correct?
2769     {\PDFForm@Push}
2770     {\box_use:N\l_tmpa_box}
2771     \HyField@AddToFields
2772   }
2773 }
2774 {
2775   \msg_error:nn { hyp }{ pdfa-no-push-button }
2776   \LayoutPushButtonField
2777   {
2778     \mode_leave_vertical:
2779     \MakeButtonField{#2}
2780   }
2781 }
2782 \group_end:
2783 }
2784
2785 \def\@Submit[#1]#2
2786 {
2787   \def\Fld@width {\DefaultWidthofSubmit}
2788   \def\Fld@height{\DefaultHeightofSubmit}
2789   \group_begin:
2790     \exp_args:No\HyField@SetKeys
2791     {
2792       \DefaultOptionsofSubmit,#1
2793     }
2794     \HyField@FlagsPushButton
2795     \HyField@FlagsSubmit
2796     \legacy_if:nT { Fld@hidden }
2797     {
2798       \def\Fld@width{1sp}
2799     }
2800     \mode_leave_vertical:
2801     \HyAnn@AbsPageLabel
2802     \Hy@escapeform\PDFForm@Submit
2803     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2804     \pdfxform_if_exist:nF

```

```

2805 { __hyp_xform_Submit }
2806 {
2807   \pdfxform_new:nnn { __hyp_xform_Submit }{ }
2808   {
2809     \fbox{\color_select:n{yellow}\textsf{Submit}}
2810   }
2811   \pdfxform_new:nnn { __hyp_xform_SubmitP }{ }
2812   {
2813     \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2814   }
2815 }
2816 \pdfannot_box:nnnn
2817 {\box_wd:N\l_tmpa_box}
2818 {\box_ht:N\l_tmpa_box}
2819 {\box_dp:N\l_tmpa_box} %is this correct?
2820 {
2821   \PDFForm@Submit
2822   /AP<<
2823     /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2824     /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2825   >>
2826 }
2827 \HyField@AddToFields
2828 \box_use:N\l_tmpa_box
2829
2830 \group_end:
2831 }
2832
2833 \def\@Reset[#1]#2
2834 {
2835   \def\Fld@width {\DefaultWidthofReset}
2836   \def\Fld@height{\DefaultHeightofReset}
2837   \group_begin:
2838     \exp_args:No\HyField@SetKeys
2839     {
2840       \DefaultOptionsofReset,#1
2841     }
2842   \mode_leave_vertical:
2843   \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2844   {
2845     \HyField@FlagsPushButton
2846     \legacy_if:nT { Fld@hidden }
2847     { \def\Fld@width{1sp} }
2848     \HyAnn@AbsPageLabel
2849     \Hy@escapeform\PDFForm@Reset
2850     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2851     \pdfannot_box:nnnn
2852     {\box_wd:N\l_tmpa_box}
2853     {\box_ht:N\l_tmpa_box}
2854     {\box_dp:N\l_tmpa_box} %is this correct?
2855     { \PDFForm@Reset }
2856     \HyField@AddToFields
2857     \box_use:N \l_tmpa_box
2858   }

```

```

2859     {
2860         \msg_error:nn { hyp }{ pdfa-no-reset-button }
2861         \MakeButtonField{#2}
2862     }
2863     \group_end:
2864 }
2865
2866 \def\@CheckBox[#1]#2
2867 {% parameters, label
2868     \def\Fld@name{#2}
2869     \def\Fld@default{0}
2870     \group_begin:
2871         \def\Fld@width {\DefaultWidthofCheckBox}
2872         \def\Fld@height{\DefaultHeightofCheckBox}
2873         \exp_args:No\HyField@SetKeys
2874         {
2875             \DefaultOptionsofCheckBox,#1
2876         }
2877         \PDFForm@Name
2878         \HyField@FlagsCheckBox
2879         \legacy_if:nT { Fld@hidden }
2880         {
2881             \def\Fld@width{1sp}
2882         }
2883         \LayoutCheckField{#2}
2884         {
2885             \mode_leave_vertical:
2886             \HyAnn@AbsPageLabel
2887             \Hy@escapeform\PDFForm@Check
2888             \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
2889             {
2890                 \pdfxform_new:nnn
2891                 {__hyp_xform_CheckMarkYes}{-}
2892                 {
2893                     \group_begin:
2894                     \fontfamily{pzd}
2895                     \fontencoding{U}
2896                     \fontseries{m}
2897                     \fontshape{n}
2898                     \selectfont
2899                     \char51
2900                     \group_end:
2901                 }
2902                 \pdfxform_new:nnn
2903                 {__hyp_xform_CheckMarkOff}{-}
2904                 {
2905                     \group_begin:
2906                     \fontfamily{pzd}
2907                     \fontencoding{U}
2908                     \fontseries{m}
2909                     \fontshape{n}
2910                     \selectfont
2911                     \phantom{\char51} %perhaps xetex needs some small glyph ..
2912                     \group_end:

```

```

2913     }
2914   }
2915   \pdfannot_box:nnnn
2916     {\Fld@width}
2917     {\Fld@height}
2918     {Opt} %is this correct?
2919     {\PDFForm@Check}
2920     \HyField@AddToFields %check if this works with xelatex ...
2921   }
2922 \group_end:
2923 }
2924 \ExplSyntaxOff
2925
2926 %hm. Should a luatex driver use type1 fonts in fields????
2927 \ExplSyntaxOn
2928 \def\Hy@FormObjects
2929 {
2930   \pdf_object_new:n   {__hyp/Encoding/pdfdoc }
2931   \pdf_object_new:n   {__hyp/Font/ZaDb }
2932   \pdf_object_new:n   {__hyp/Font/Helv }
2933   \pdf_object_write:nne {__hyp/Encoding/pdfdoc } { dict }
2934   {
2935     /Type/Encoding
2936     /Differences[
2937       24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2938       /ring/tilde
2939       \c_space_tl
2940       39/quotesingle
2941       \c_space_tl
2942       96/grave %
2943       \iow_newline:
2944       128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2945       /fraction/guilsinglleft/guilsinglright/minus/perthousand
2946       /quotedblbase/quotedblleft/quotedblright/quoteleft
2947       /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2948       /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2949       \iow_newline:
2950       164/currency
2951       \c_space_tl
2952       166/brokenbar
2953       \c_space_tl
2954       168/dieresis/copyright/ordfeminine
2955       \c_space_tl
2956       172/logicalnot/.notdef/registered/macron/degree/plusminus
2957       /twosuperior/threesuperior/acute/mu
2958       \c_space_tl
2959       183/periodcentered/cedilla/onesuperior/ordmasculine
2960       \c_space_tl
2961       188/onequarter/onehalf/threequarters
2962       \iow_newline:
2963       192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
2964       /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
2965       /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
2966       /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave

```

```

2967         /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
2968         /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
2969         /cedilla/egrave/eacute/ecircumflex/edieresis/igrave
2970         /iacute/icircumflex/idieresis/eth/ntilde/ograve/oacute
2971         /ocircumflex/otilde/odieresis/divide/oshlash/ugrave
2972         /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
2973     ]
2974 }
2975 \pdf_object_write:nnn {__hyp/Font/ZaDb } { dict }
2976 {
2977     /Type/Font
2978     /Subtype/Type1
2979     /Name/ZaDb
2980     /BaseFont/ZapfDingbats
2981 }
2982 \pdf_object_write:nne {__hyp/Font/Helv } { dict }
2983 {
2984     /Type/Font
2985     /Subtype/Type1
2986     /Name/Helv
2987     /BaseFont/Helvetica
2988     /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
2989 }
2990 \global\let\Hy@FormObjects\relax
2991 }
2992 \ExplSyntaxOff
2993 \providecommand*{\Fld@pageobjref}{\}
2994 \ifcsname pdf@escapestring\endcsname
2995   \def\Hy@escapeform#1{%
2996     \ifHy@pdfescapeform
2997       \let\Hy@escapestring\pdfescapestring
2998     \else
2999       \let\Hy@escapestring\@firstofone
3000     \fi
3001   }%
3002   \Hy@escapeform{}}%
3003 \else
3004   \let\Hy@escapestring\@firstofone
3005   \def\Hy@escapeform#1{%
3006     \ifHy@pdfescapeform
3007       \def\Hy@escapestring##1{%
3008         \noexpand\Hy@escapestring{\noexpand##1}%
3009       }%
3010       \edef\Hy@temp{#1}%
3011       \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{\}\@nil
3012       \def\Hy@escapestring##1{%
3013         \@ifundefined{Hy@esc@\string##1}{%
3014           ##1%
3015           \ThisShouldNotHappen
3016         }{%
3017           \csname Hy@esc@\string##1\endcsname
3018         }%
3019       }%
3020   \else

```



```

3021     \let\Hy@escapestring\@firstofone
3022     \fi
3023 }%
3024 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3025     \ifx\#3\%
3026     \else
3027     \expandafter
3028     \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3029     \Hy@ReturnAfterFi{%
3030     \Hy__hypescapeform#3\@nil
3031     }%
3032     \fi
3033 }%
3034 \fi
3035 \def\PDFForm@Name{%
3036     \PDFForm__hypName\Fld@name
3037     \ifx\Fld@altname\relax
3038     \else
3039     \PDFForm__hypName\Fld@altname
3040     \fi
3041     \ifx\Fld@mappingname\relax
3042     \else
3043     \PDFForm__hypName\Fld@mappingname
3044     \fi
3045 }
3046 \def\PDFForm__hypName#1{%
3047     \beginngroup
3048     \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3049     \ifHy@unicode
3050     \Hy@unicodedefalse
3051     \fi
3052     \fi
3053     \pdfstringdef\Hy@gtemp#1%
3054     \endgroup
3055     \let#1\Hy@gtemp
3056 }
3057 \def\Fld@X@additionalactions{%
3058     \ifx\Fld@keystroke@code\@empty
3059     \else
3060     /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3061     \fi
3062     \ifx\Fld@format@code\@empty
3063     \else
3064     /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3065     \fi
3066     \ifx\Fld@validate@code\@empty
3067     \else
3068     /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3069     \fi
3070     \ifx\Fld@calculate@code\@empty
3071     \else
3072     /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3073     \fi
3074     \ifx\Fld@onfocus@code\@empty

```

```

3075 \else
3076 /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3077 \fi
3078 \ifx\Fld@onblur@code\@empty
3079 \else
3080 /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3081 \fi
3082 \ifx\Fld@onmousedown@code\@empty
3083 \else
3084 /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3085 \fi
3086 \ifx\Fld@onmouseup@code\@empty
3087 \else
3088 /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3089 \fi
3090 \ifx\Fld@onenter@code\@empty
3091 \else
3092 /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3093 \fi
3094 \ifx\Fld@onexit@code\@empty
3095 \else
3096 /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3097 \fi
3098 }
3099 \ExplSyntaxOn
3100 \def\Fld@additionalactions
3101 {%
3102 \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{\}
3103 {
3104 \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3105 {/AA<<\Fld@X@additionalactions>>}}
3106 }
3107 }
3108 \ExplSyntaxOff
3109 \def\Fld@annotnames{%
3110 /T(\Fld@name)%
3111 \ifx\Fld@altname\relax
3112 \else
3113 /TU(\Fld@altname)%
3114 \fi
3115 \ifx\Fld@mappingname\relax
3116 \else
3117 /TM(\Fld@mappingname)%
3118 \fi
3119 }
3120 \ExplSyntaxOn
3121 \def\PDFForm@Check
3122 {
3123 /Subtype/Widget
3124 ~\Fld@annotflags
3125 ~\Fld@pageobjref
3126 ~\Fld@annotnames
3127 /FT/Btn
3128 \Fld@flags

```

```

3129 /Q~\Fld@align
3130 /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3131 /AP
3132 <<
3133 /N
3134 <<
3135 /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3136 /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3137 >>
3138 >>
3139 /MK<<
3140 \int_compare:nNnF {\Fld@rotation}={0}
3141 {
3142 /R~\Fld@rotation
3143 }
3144 \tl_if_empty:NF\Fld@bordercolor
3145 {
3146 /BC[\Fld@bordercolor]
3147 }
3148 \tl_if_empty:NF\Fld@bcolor
3149 {
3150 /BG[\Fld@bcolor]
3151 }
3152 /CA(\Hy@escapestring{\Fld@cbsymbol})%
3153 >>
3154 /DA
3155 (
3156 /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3157 \tl_if_empty:NF \Fld@color
3158 {
3159 \c_space_tl \Fld@color
3160 }
3161 )
3162 /H/P
3163 \legacy_if:nTF {\Fld@checked}
3164 {
3165 /V/Yes /AS/Yes
3166 }
3167 {
3168 /V/Off /AS/Off
3169 }
3170 \Fld@additionalactions
3171 }
3172 \ExplSyntaxOff
3173 \ExplSyntaxOn
3174 \def\PDFForm@Push
3175 {
3176 /Subtype/Widget
3177 ~\Fld@annotflags
3178 ~\Fld@pageobjref
3179 ~\Fld@annotnames
3180 /FT/Btn
3181 ~\Fld@flags
3182 /H/P

```

```

3183 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3184 \bool_if:nT
3185 {
3186   !\int_compare_p:nNn {\Fld@rotation} = {0}
3187   ||
3188   \tl_if_exist_p:N \Fld@bordercolor
3189 }
3190 {
3191   /MK
3192   <<
3193     \int_compare:nNf {\Fld@rotation} = {0}
3194     {
3195       /R~\Fld@rotation
3196     }
3197     \tl_if_exist:NT \Fld@bordercolor
3198     {
3199       /BC[\Fld@bordercolor]
3200     }
3201   >>
3202 }
3203 /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3204 \Fld@additionalactions
3205 }
3206
3207 \ExplSyntaxOff
3208 \def\PDFForm@List{%
3209   /Subtype/Widget%
3210   \Fld@annotflags
3211   \Fld@pageobjref
3212   \Fld@annotnames
3213   /FT/Ch%
3214   \Fld@flags
3215   /Q \Fld@align
3216   /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3217   \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3218     \ifx\Fld@bordercolor\relax\else 1\fi
3219     \ifx\Fld@bcolor\relax \else 1\fi
3220     \space
3221   \else
3222     /MK<<%
3223     \ifnum\Fld@rotation=\z@
3224     \else
3225       /R \Fld@rotation
3226     \fi
3227     \ifx\Fld@bordercolor\relax
3228     \else
3229       /BC[\Fld@bordercolor]%
3230     \fi
3231     \ifx\Fld@bcolor\relax
3232     \else
3233       /BG[\Fld@bcolor]%
3234     \fi
3235   >>%
3236   \fi

```

```

3237 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3238 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3239 \Fld@choices
3240 \Fld@additionalactions
3241 }
3242 \ExplSyntaxOn
3243 \def\PDFForm@Radio
3244 {
3245 /Subtype/Widget
3246 ~\Fld@annotflags
3247 ~\Fld@pageobjref
3248 ~\Fld@annotnames
3249 /FT/Btn
3250 \Fld@flags
3251 /H/P
3252 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3253 /MK<<
3254 \ifnum\Fld@rotation=\z@
3255 \else
3256 /R~\Fld@rotation
3257 \fi
3258 \ifx\Fld@bordercolor\relax
3259 \else
3260 /BC[\Fld@bordercolor]%
3261 \fi
3262 \ifx\Fld@bcolor\relax
3263 \else
3264 /BG[\Fld@bcolor]%
3265 \fi
3266 /CA(\Hy@escapestring{\Fld@radiosymbol})%
3267 >>
3268 /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3269 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3270 \ifx\Fld@default\@empty
3271 /V/Off%
3272 /DV/Off%
3273 \else
3274 /V/\Fld@default
3275 /DV/\Fld@default
3276 \fi
3277 \Fld@additionalactions
3278 }
3279 \ExplSyntaxOff
3280 \ExplSyntaxOn
3281 % Does an appeareance dict make sense here?
3282 \def\PDFForm@Text
3283 {
3284 /Subtype/Widget
3285 ~\Fld@annotflags
3286 ~\Fld@pageobjref
3287 ~\Fld@annotnames
3288 /FT/Tx
3289 ~\Fld@flags
3290 /Q~\Fld@align

```

```

3291     /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3292     \bool_if:nT
3293     {
3294         !\int_compare_p:nNn {\Fld@rotation} = {0}
3295         ||
3296         \tl_if_exist_p:N \Fld@bordercolor
3297         ||
3298         \tl_if_exist_p:N \Fld@bcolor
3299     }
3300     {
3301         /MK
3302         <<
3303             \int_compare:nNnF {\Fld@rotation} = {0}
3304             {
3305                 /R~\Fld@rotation
3306             }
3307             \tl_if_exist:NT \Fld@bordercolor
3308             {
3309                 /BC[\Fld@bordercolor]
3310             }
3311             \tl_if_exist:NT \Fld@bcolor
3312             {
3313                 /BG[\Fld@bcolor]
3314             }
3315         >>
3316     }
3317     /DA
3318     (
3319         /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3320         \tl_if_empty:NF {\c_space_tl\Fld@color}
3321     )
3322     /DV(\Hy@escapestring{\Fld@default})
3323     /V(\Hy@escapestring{\Fld@value})
3324     ~\Fld@additionalactions
3325     \int_compare:nNnT { \Fld@maxlen}>{0}
3326     {
3327         /MaxLen~\Fld@maxlen
3328     }
3329 }
3330 \ExplSyntaxOff
3331
3332 \def\PDFForm@Submit{%
3333     /Subtype/Widget%
3334     \Fld@annotflags
3335     \Fld@pageobjref
3336     \Fld@annotnames
3337     /FT/Btn%
3338     \Fld@flags
3339     /H/P%
3340     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3341     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3342         \ifx\Fld@bordercolor\relax\else 1\fi
3343         \space
3344     \else

```

```

3345 /MK<<%
3346 \ifnum\Fld@rotation=\z@
3347 \else
3348 /R \Fld@rotation
3349 \fi
3350 \ifx\Fld@bordercolor\relax
3351 \else
3352 /BC[\Fld@bordercolor]%
3353 \fi
3354 >>%
3355 \fi
3356 /A<<%
3357 /S/SubmitForm%
3358 /F<<%
3359 /FS/URL%
3360 /F(\Hy@escapestring{\Form@action})%
3361 >>%
3362 \Fld@submitflags
3363 >>%
3364 \Fld@additionalactions
3365 }
3366 \ExplSyntaxOn
3367 \def\PDFForm@Reset{%
3368 /Subtype/Widget%
3369 \Fld@annotflags
3370 \Fld@pageobjref
3371 \Fld@annotnames
3372 /FT/Btn%
3373 \Fld@flags
3374 /H/P%
3375 /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3376 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3377 \ifx\Fld@bordercolor\relax\else 1\fi
3378 \space
3379 \else
3380 /MK<<%
3381 \ifnum\Fld@rotation=\z@
3382 \else
3383 /R~\Fld@rotation
3384 \fi
3385 \ifx\Fld@bordercolor\relax
3386 \else
3387 /BC[\Fld@bordercolor]%
3388 \fi
3389 >>%
3390 \fi
3391 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3392 /A<</S/ResetForm>>%
3393 \Fld@additionalactions
3394 }%
3395
3396
3397 %these patterns are used in hyperref checks.
3398 %it is unclear if they are really useful and if a backend support is

```

```

3399 %needed.
3400 \str_case:VnF \c_sys_backend_str
3401 {
3402   { pdfmode }
3403   {
3404     \def\HyPat@ObjRef
3405     {
3406       [0-9]*[1-9][0-9]*~0~R
3407     }
3408   }
3409   { dvipdfmx }
3410   {
3411     \def\HyPat@ObjRef
3412     {
3413       @[\~]+
3414     }
3415   }
3416   { xdvipdfmx }
3417   {
3418     \def\HyPat@ObjRef
3419     {
3420       @[\~]+
3421     }
3422   }
3423 }
3424 { %also set in hyperref sty, so probably not needed.
3425   \def\HyPat@ObjRef/{.+}
3426 }
3427
3428
3429 \ExplSyntaxOff
3430 % UF: removed Hy@writebookmark
3431 %   \Hy@currentbookmarklevel{0}
3432 %   \Hy@numberline
3433 %   \__hypwritetorep
3434 %   counter{bookmark@seq@number}
3435 % removed \HyPsd@SanitizeForOutFile, not needed
3436 % removed \currentpdfbookmark, defined by bookmark,
3437 % should use \newcommand there
3438 % removed \subpdfbookmark, defined by bookmark,
3439 % should use \newcommand there
3440 % removed \belowpdfbookmark, defined by bookmark,
3441 % should use \newcommand there
3442 % removed \pdfbookmark, defined by bookmark,
3443 % \BOOKMARK
3444 % \@BOOKMARK
3445 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3446 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3447 %% removed \ReadBookmarks / unneeded with bookmark.
3448 %% removed \Hy@OutlineName
3449 %% removed \check@bm@number
3450 %% removed \calc@bm@number
3451
3452 \ifHy@implicit

```



```

3453 \else
3454   \expandafter\endinput
3455 \fi
3456 \newlength\Hy@SectionHShift
3457 \def\Hy@SectionAnchorHref#1{%
3458   \ifx\protect\@typeset@protect
3459     \Hy__hypSectionAnchor{#1}%
3460   \fi
3461 }
3462 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%
3463   \leavevmode
3464   \hbox to Opt{%
3465     \kern-\Hy@SectionHShift
3466     \Hy@raisedlink{%
3467       \hyper@anchorstart{#1}\hyper@anchorend
3468     }%
3469     \hss
3470   }%
3471 }
3472 \@ifundefined{hyper@nopatch@sectioning}
3473 {
3474   \let\H@old@ssect\@ssect
3475   \def\@ssect#1#2#3#4#5{%
3476     \Hy@MakeCurrentHrefAuto{section*}%
3477     \setlength{\Hy@SectionHShift}{#1}%
3478     \begingroup
3479       \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3480       \toks\tw@\expandafter{%
3481         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3482         #5%
3483       }%
3484       \edef\x{\endgroup
3485         \the\toks@{\the\toks\tw@}%
3486       }\x
3487   }
3488   \let\H@old@schapter\@schapter
3489   \def\@schapter#1{%
3490     \begingroup
3491       \let\@mkboth\@gobbletwo
3492       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3493       \Hy@raisedlink{%
3494         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3495       }%
3496     \endgroup
3497     \H@old@schapter{#1}%
3498   }
3499   \@ifundefined{chapter}{-}{%
3500     \let\Hy@org@chapter\@chapter
3501     \def\@chapter{%
3502       \def\Hy@next{%
3503         \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3504         \Hy@raisedlink{%
3505           \hyper@anchorstart{\@currentHref}\hyper@anchorend
3506         }%

```

```

3507 }%
3508 \ifnum\c@secnumdepth>\m@ne
3509   \ifundefined{if@mainmatter}%
3510     \iftrue{\csname if@mainmatter\endcsname}%
3511     \let\Hy@next\relax
3512   \fi
3513   \fi
3514   \Hy@next
3515   \Hy@org@chapter
3516 }%
3517 }
3518 \let\H@old@part\@part
3519 \begingroup\expandafter\expandafter\expandafter\endgroup
3520 \expandafter\ifx\csname chapter\endcsname\relax
3521   \let\Hy@secnum@part\z@
3522 \else
3523   \let\Hy@secnum@part\m@ne
3524 \fi
3525 \def\@part{%
3526   \ifnum\Hy@secnum@part>\c@secnumdepth
3527     \phantomsection
3528   \fi
3529   \H@old@part
3530 }
3531 \let\H@old@spart\@spart
3532 \def\@spart#1{%
3533   \Hy@MakeCurrentHrefAuto{part*}%
3534   \Hy@raisedlink{%
3535     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3536   }%
3537   \H@old@spart{#1}%
3538 }
3539 \let\H@old@sect\@sect
3540 \def\@sect#1#2#3#4#5#6[#7]#8{%
3541   \ifnum #2>\c@secnumdepth
3542     \expandafter\@firstoftwo
3543   \else
3544     \expandafter\@secondoftwo
3545   \fi
3546   {%
3547     \Hy@MakeCurrentHrefAuto{section*}%
3548     \setlength{\Hy@SectionHShift}{#3}%
3549     \begingroup
3550       \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3551       \toks\tw@\expandafter{%
3552         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3553         #8%
3554       }%
3555     \edef\x{\endgroup
3556       \the\toks@\the\toks\tw@}%
3557   }\x
3558 }{%
3559   \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]{#8}%
3560 }%

```

```

3561 }
3562 }{}
3563 \expandafter\def\csname Parent-4\endcsname{}
3564 \expandafter\def\csname Parent-3\endcsname{}
3565 \expandafter\def\csname Parent-2\endcsname{}
3566 \expandafter\def\csname Parent-1\endcsname{}
3567 \expandafter\def\csname Parent0\endcsname{}
3568 \expandafter\def\csname Parent1\endcsname{}
3569 \expandafter\def\csname Parent2\endcsname{}
3570 \expandafter\def\csname Parent3\endcsname{}
3571 \expandafter\def\csname Parent4\endcsname{}
3572 %%
3573 %% End of file 'hgeneric-testphase.def'.
3574 \</package>

3575 <*colorscheme>
3576 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3577 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3578 % linkcolor=
3579 %,filecolor=
3580 %,urlcolor=
3581 %,menucolor=
3582 %,runcolor=
3583 %,linkbordercolor=
3584 %,filebordercolor=
3585 %,urlbordercolor=
3586 %,menubordercolor=
3587 %,runbordercolor=
3588
3589 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }
3590 {
3591     linkcolor      = [rgb]{1,0,0}, %red
3592     filecolor      = [rgb]{0,1,1}, %cyan
3593     urlcolor       = [rgb]{1,0,1}, %magenta
3594     menucolor      = [rgb]{1, 0, 0}, %red
3595     runcolor       = [rgb]{0,1,1}, %cyan
3596 %-----
3597     linkbordercolor = [rgb]{1, 0 ,0 },
3598     filebordercolor = [rgb]{0, .5, .5},
3599     urlbordercolor  = [rgb]{0, 1, 1},
3600     menubordercolor = [rgb]{1, 0, 0},
3601     runbordercolor  = [rgb]{0, .7, .7}
3602 }
3603
3604 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3605 {
3606     linkcolor      = [rgb]{0,0.2,0.6},
3607     filecolor      = [rgb]{0.8,0,0.8},
3608     urlcolor       = [rgb]{0.8,0,0.8},
3609     menucolor      = [rgb]{0,0.2,0.6},
3610     runcolor       = [rgb]{0.8,0,0.8},
3611 %----- %-----
3612     linkbordercolor = [rgb]{0,0.2,0.6},
3613     filebordercolor = [rgb]{0.8,0,0.8},
3614     urlbordercolor  = [rgb]{0.8,0,0.8},

```

```

3615     menubordercolor = [rgb]{0,0.2,0.6},
3616     runbordercolor  = [rgb]{0.8,0,0.8}
3617 }
3618
3619 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3620 { %two colors: intern/extern
3621     linkcolor      = [rgb]{0.79216, 0, 0.12549},
3622     filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3623     urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3624     menucolor      = [rgb]{0.79216, 0, 0.12549 },
3625     runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3626     %----- %-----
3627     linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3628     filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3629     urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3630     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3631     runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3632 }
3633
3634 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3635 { %all darkgray
3636     linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3637     filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3638     urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3639     menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3640     runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3641     %----- %-----
3642     linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3643     filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3644     urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3645     menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3646     runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3647 }
3648
3649 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3650 { %dvipsnam.def
3651     linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3652     filecolor      = [rgb]{1, 0, 0}, %Red
3653     urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3654     menucolor      = [rgb]{1, 0, 0}, %Red
3655     runcolor       = [rgb]{1, 0, 0}, %Red
3656     %----- %-----
3657     linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3658     filebordercolor = [rgb]{1, 0, 0}, %Red
3659     urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3660     menubordercolor = [rgb]{1, 0, 0}, %Red
3661     runbordercolor  = [rgb]{1, 0, 0} %Red
3662 }
3663
3664 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3665 { %dvipsnam.def
3666     linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3667     filecolor      = [rgb]{0, 1, 0}, %Green
3668     urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry

```

```

3669     menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3670     runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3671 %----- %-----
3672     linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3673     filebordercolor = [rgb]{0, 1, 0}, %Green
3674     urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3675     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3676     runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3677 }
3678
3679
3680 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3681 {
3682     linkcolor      = [rgb]{0.50196, 0, 0.02353},
3683     filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3684     urlcolor       = [rgb]{0.54118, 0, 0.52941},
3685     menucolor      = [rgb]{0.44706, 0.45882, 0},
3686     runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3687 %----- %-----
3688     linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3689     filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3690     urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3691     menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3692     runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3693 }
3694
3695 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3696 {
3697     linkcolor      = [rgb]{0,0,0},
3698     filecolor      = [rgb]{0,0,0},
3699     urlcolor       = [rgb]{0,0,0},
3700     menucolor      = [rgb]{0,0,0},
3701     runcolor       = [rgb]{0,0,0},
3702 %----- %-----
3703     linkbordercolor = [rgb]{0,0,0},
3704     filebordercolor = [rgb]{0,0,0},
3705     urlbordercolor  = [rgb]{0,0,0},
3706     menubordercolor = [rgb]{0,0,0},
3707     runbordercolor  = [rgb]{0,0,0}
3708 }
3709 </colorscheme>

```

Index

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

Symbols	
\#	280, 714
\\$	279
\%	715
\-	2097
\.	524, 528, 530
@curropt commands:	
\@curropt:	2616, 2683
\[2097
\]	20, 21, 31, 32, 33, 41, 45, 55, 75, 82, 89, 96, 103, 118, 123, 124, 132, 133, 142, 143, 144, 145, 146, 153, 161, 281, 282, 845, 2618, 2685, 3025
_	522, 524, 528, 530
\]	2097
A	
\A	522, 2097
\Acrobatmenu	18, 174
\addcontentsline	13
\AddToDocumentProperties	389, 2092, 2193
\AddToHook	427, 440
\AddToHookNext	199
\advance	2548, 2623, 2667, 2668
allcolors (hypersetup key)	958
\author	2
B	
\b	526
\begingroup	210, 278, 376, 2454, 2583, 2612, 2624, 2658, 3047, 3478, 3490, 3519, 3549
\belowpdfbookmark	3440
\bgroup	275, 376
\BOOKMARK	3443
bookmarkstype (hypersetup key)	13
bool commands:	
\bool_if:NTF	268, 297, 356, 661, 677, 689, 702, 710, 740, 797, 856, 928, 938, 1182, 1193
\bool_if:nTF	809, 1867, 1892, 3184, 3292
\bool_lazy_and:nnTF	424
\bool_lazy_or:nnTF	1212, 1261
\bool_new:N	204, 205, 508, 512, 516
\bool_set_true:N	517
bordercolormodel (hypersetup key)	13, 979
box commands:	
\box_dp:N	598, 2768, 2819, 2854
\box_ht:N	597, 2767, 2818, 2853
\box_new:N	457, 519
\box_set_dp:Nn	2064
\box_set_ht:Nn	2071
\box_use:N	1199, 2770, 2828, 2857
\box_use_drop:N	1204
\box_wd:N	596, 2766, 2817, 2852
\l_tmpa_box	2764, 2766, 2767, 2768, 2770, 2803, 2817, 2818, 2819, 2828, 2850, 2852, 2853, 2854, 2857
C	
\catcode	279, 280
\char	2702, 2899, 2911
\chardef	159
\cite	32
clist commands:	
\clist_item:nn	2124
\clist_map_function:nN	125, 134
\clist_map_inline:nn	2257
\clist_map_variable:nNn	2615, 2682
color commands:	
\color_export:nnN	38, 432, 887, 1000
\color_select:n	895, 931, 1203, 2809, 2813
\color_select:nn	901
\color_set:nn	2, 394, 912
\color_set:nnn	2, 393, 918
color names:	
hyp/annot/file	505
hyp/annot/link	505
hyp/annot/menu	505
hyp/annot/run	505
hyp/annot/url	505
colorfile (hypersetup key)	958
colorlink (hypersetup key)	958
colorlinks (hypersetup key)	944
colormenu (hypersetup key)	958
colorrun (hypersetup key)	958
colorscheme (hypersetup key)	1, 1370
colorurl (hypersetup key)	958
cs commands:	
\cs_generate_variant:Nn	156, 157, 158, 391, 456, 560, 737, 887, 904, 921
\cs_gset:Npn	1174
\cs_gset_eq:NN	410, 421
\cs_if_exist:NTF	13
\cs_if_exist_p:N	425

\ExplSyntaxOn	hidefile (hypersetup key)	1346
... 8, 11, 2486, 2543, 2567, 2927,	hidelink (hypersetup key)	1346
3099, 3120, 3173, 3242, 3280, 3366	hidelinks (hypersetup key)	1346
extension (hypersetup key)	hidemenu (hypersetup key)	1346
13, 1412	hiderun (hypersetup key)	1346
	hideurl (hypersetup key)	1346
F	hook commands:	
\fbox	\hook_gput_code:nnn	
2721, 2809, 2813 403, 414, 924, 934, 1178, 1189	
\fi	\hook_new:n	392, 535, 610
7, 2442,	\hook_new_pair:nn	
2469, 2472, 2481, 2581, 2589, 2590, 261, 289, 313, 330, 348	
2621, 2639, 2642, 2643, 2649, 2652,	\hook_use:n	
2661, 2663, 2670, 2673, 3000, 3022,	. 265, 285, 294, 310, 317, 327, 334,	
3032, 3034, 3040, 3044, 3051, 3052,	344, 353, 372, 552, 616, 625, 666, 682	
3061, 3065, 3069, 3073, 3077, 3081,	\href	4, 5, 19, 263
3085, 3089, 3093, 3097, 3114, 3118,	\hreflaunch	5
3217, 3218, 3219, 3226, 3230, 3234,	\hrefpdf	5, 19, 314, 384
3236, 3238, 3257, 3261, 3265, 3269,	\hrefrun	5, 331, 385
3276, 3341, 3342, 3349, 3353, 3355,	\hrefurl	5, 19, 291, 383
3376, 3377, 3384, 3388, 3390, 3455,	\hss	3469
3460, 3512, 3513, 3524, 3528, 3545	Hy internal commands:	
file (hypersetup key)	\Hy_hyescapeform ..	3011, 3024, 3030
10, 1443	\Hy_hypSectionAnchor ...	3459, 3462
file commands:	HyField internal commands:	
\file_input:n	\HyField_hypAddToFields .	2448, 2475
9	hyp commands:	
fileborderstyle (hypersetup key) 14, 1072	\l_hyp_annot_colorfile_bool ...	506
filecolor (hypersetup key)	\l_hyp_annot_colorlink_bool ...	506
958	\l_hyp_annot_colormenu_bool ...	506
final (hypersetup key)	\l_hyp_annot_colorrun_bool ...	506
1394	\l_hyp_annot_colorurl_bool ...	506
\fontencoding	\l_hyp_annot_ocgcolorfile_bool .	510
2698, 2895, 2907	\l_hyp_annot_ocgcolorlink_bool .	510
\fontfamily	\l_hyp_annot_ocgcolormenu_bool .	510
2697, 2894, 2906	\l_hyp_annot_ocgcolorrun_bool ..	510
\fontseries	\l_hyp_annot_ocgcolorurl_bool ..	510
2699, 2896, 2908	\l_hyp_annot_ocgcolorurl_bool ..	510
\fontshape	\l_hyp_current_dest_name_tl	
2700, 2897, 2909 609, 665, 681	
fp commands:	hyp internal commands:	
\fp_eval:n	\g__hyp_AcroForm_CoFields_prop ..	
2022, 2050, 2066 2412, 2430, 2499, 2501, 2516	
	\g__hyp_AcroForm_Fields_prop ...	
G 2413, 2423, 2494	
\gdef	\l__hyp_annot_GoTo_bool 661, 677, 689	
281, 282	\l__hyp_annot_GoTo_bool\l__-	
\GetDocumentProperties	_hyp_annot_URI_bool\l__-	
3	hyp_annot_GoToR_bool\l__-	
\global	hyp_annot_Named_bool\l__-	
2548, 2990	hyp_annot_Launch_bool	514
group commands:	\l_hyp_annot_GoToR_bool	740
\group_begin: .. 266, 295, 318, 335,	\l_hyp_annot_Launch_bool	797
354, 550, 664, 680, 704, 742, 799,	\l_hyp_annot_Named_bool	856
858, 930, 1185, 1202, 2326, 2696,		
2746, 2789, 2837, 2870, 2893, 2905		
\group_end:		
309, 326, 343,		
371, 557, 669, 692, 728, 787, 839,		
882, 940, 1205, 1208, 2333, 2703,		
2782, 2830, 2863, 2900, 2912, 2922		
H		
\hbox		
376, 3464		
hbox commands:		
\hbox_overlap_right:n		
1199		
\hbox_set:Nn		
2764, 2803, 2850		
\hbox_set:Nw		
1186		
\hbox_set_end:		
1195		
\hbox_set_to_wd:Nnn		
2048		

\c__hyp_annot_types_seq	\l__hyp_href_run_parameter_tl ...
..... 469, 506, 510, 958, 1259, 1357 209, 230, 342
\l__hyp_annot_URI_bool	__hyp_href_url_aux:n
702	363, 366
\g__hyp_bordercolormodel_str ...	__hyp_href_url_aux:nn
..... 434, 505, 983, 1002	304, 306
__hyp_citebordercolor_hook_-	\l__hyp_href_url_encode_bool ...
init: 204, 215, 253, 268, 297, 356
399, 412, 421	__hyp_href_url_format:
__hyp_citecolor_hook_init: 210, 216, 258, 369
..... 397, 401, 410	\l__hyp_href_url_ismap_bool
__hyp_clist_display:n . 118, 125, 134 205, 229, 710
__hyp_color_select:n 38, 888, 888, 904	\l__hyp_href_url_protocol_tl ...
__hyp_color_select_aux:wn 206, 217, 256, 308, 370
..... 888, 892, 899	__hyp_link_goto_begin:nw
__hyp_color_set:nn 632, 667, 683
38, 397, 399, 431, 905, 905, 921, 963, 999	__hyp_link_goto_end: . 654, 668, 691
__hyp_color_set_aux:nwn 905, 909, 916	\c__hyp_map_annot_hyp_prop 469
\c__hyp_colorscheme_daleif_prop 3604	\c__hyp_map_hyp_annot_prop
\c__hyp_colorscheme_henryford_- 469, 922,
prop	986, 1026, 1053, 1061, 1072, 1099,
3695	1107, 1176, 1284, 1320, 1330, 1443
\c__hyp_colorscheme_julian_prop 3619	__hyp_ocg_init: 1118, 1118, 1174, 1184
\c__hyp_colorscheme_phelype_prop	\l__hyp_optlang_regex 2096, 2097, 2126
..... 3680	l__hyp_page/Trans
\c__hyp_colorscheme_szabolcsA_-	533
prop	__hyp_PageLabels_gpush: 567, 567, 580
3649	\l__hyp_para_tmpa_tl 465, 808, 811, 822
\c__hyp_colorscheme_szabolcsB_-	\l__hyp_para_tmpa_tl_uuu\l__hyp_-
prop	text_tmpa_str_uuu\g__hyp_text_-
3664	tmpa_str
\c__hyp_colorscheme_tivv_prop . 3634	462
\l__hyp_dest_box	__hyp_property_record:nn
31, 519, 596, 597, 598, 2048, 2064, 2071 449, 450, 456, 2552
\l__hyp_dest_name_tmpa_tl .. 462,	__hyp_secondoftwoewithopt:wnn ...
635, 636, 642, 646, 648, 651, 760, 773 382, 383, 384, 385
\l__hyp_dest_pdfremotestartview_-	__hyp_setup_info_date_key:nn ...
tl 2179, 2212, 2213
488, 768, 1855, 1859	__hyp_setup_info_key:nn . 2098,
\g__hyp_dest_pdfstartpage_tl ...	2172, 2173, 2174, 2175, 2177, 2178
... 488, 1866, 1868, 1875, 1893, 1900	__hyp_store_metadata:nn
\g__hyp_dest_pdfstartview_tl 386, 391, 1462, 2140,
488, 1868, 1875, 1886, 1890, 1893, 1900	2158, 2192, 2207, 2216, 2232, 2300
\l__hyp_dest_pdfview_tl	__hyp_text_cleanup:N . 540, 540, 554
..... 504, 614, 623,	\l__hyp_text_enc_dest_print_tl ..
2017, 2020, 2027, 2030, 2031, 2032, 491, 759
2033, 2034, 2035, 2040, 2044, 2080	\l__hyp_text_enc_dest_tl
\c__hyp_dest_startview_regex 31, 491, 590, 647
..... 520, 1853, 1884	\l__hyp_text_enc_file_print_tl ..
\c__hyp_dest_undefined_tl 491, 802
..... 468, 641, 642	\l__hyp_text_enc_info_print_tl ...
__hyp_destination:nn 491, 565
..... 31, 583, 583, 614, 623	\l__hyp_text_enc_para_print_tl ...
\l__hyp_filename_tmpa_tl ... 462, 491, 807
743, 745, 750, 751, 756, 803, 804, 822	\l__hyp_text_enc_uri_print_tl ...
__hyp_href_pdf_aux:nn 270,
320, 323	273, 299, 302, 358, 361, 491, 707, 1454
\l__hyp_href_pdf_destination_tl .	
..... 207, 218, 257, 325	
\l__hyp_href_pdf_page_tl 208, 226, 767	
__hyp_href_run_aux:nn	
337, 340	

_hyp_text_pdfstring:nnN	final	1394
..... 548, 548, 560, 562,	hidefile	1346
565, 588, 645, 705, 757, 800, 805, 1454	hidelink	1346
_hyp_text_pdfstring_info:nN ...	hidelinks	1346
..... 563, 563, 2129, 2133, 2152, 2162	hidemenu	1346
_hyp_text_purify:nN . 536, 536, 553	hiderun	1346
_hyp_text_string_from_unicode:nN	hideurl	1346
..... 544, 544, 555	hypertextnames	1412
\g_hyp_text_tmpa_str . 467, 556, 558	link	10, 1443
\l_hyp_text_tmpa_str	linkborder	13
..... 466, 553, 554, 555, 556	linkborderstyle	14, 1072
\l_hyp_tmpa_box 457, 1186, 1199, 1204	linkcolor	958
\l_hyp_tmpa_int	linkfileprefix	1412
457	linktoc	1412
\l_hyp_tmpa_seq	linktocpage	1412
..... 457, 1853, 1855, 1884, 1886,	localanchorname	1412
2007, 2008, 2012, 2014, 2038, 2045,	menu	10, 1443
2046, 2047, 2056, 2058, 2068, 2073,	menuborder	13
2126, 2127, 2133, 2503, 2505, 2511	menuborderstyle	14, 1072
\l_hyp_tmpa_str	menucolor	958
..... 457, 2129, 2133, 2135, 2137,	naturalnames	1412
2152, 2153, 2155, 2162, 2163, 2167	nesting	13
\l_hyp_tmpa_tl	ocgcolorfile	1212
.. 457, 591, 595, 603, 1003, 1007,	ocgcolorlink	1212
1454, 1455, 1460, 1851, 1853, 1882,	ocgcolorlinks	1212
1884, 2014, 2015, 2022, 2124, 2126	ocgcolormenu	1212
\l_hyp_uri_tmpa_tl ... 462, 708, 709	ocgcolorrun	1212
hyp/anchor	ocgcolorurl	1212
hyp/annot/file (color name)	pageanchor	1412
hyp/annot/link (color name)	pdfauthor	2096
hyp/annot/menu (color name)	pdfborder	13
hyp/annot/run (color name)	pdfborderstyle	14, 1072
hyp/annot/url (color name)	pdfcreationdate	14, 2179
hyp/text/pdfstring	pdfcreator	2096
535	pdfencoding	1384
\hypercalcbp	pdfinfo	2246
11, 18, 162	pdfkeywords	2096
\HyperDestNameFilter	pdflang	14, 2085
13, 589, 646	pdflinkmargin	14
\hypersetup 1, 2, 5, 9, 10, 13, 18, 66, 98, 181	pdfmetadate	14, 2179
\hypersetup keys:	pdfmoddate	14, 2179
allcolors	pdfproducer	2096
958	pdfremotestartview	11
bookmarkstyle	pdfstartview	11
13	pdfsubject	2096
bordercolormodel	pdftitle	2096
13, 979	pdftrapped	2218
colorfile	pdfversion	1384
958	pdfview	11, 2003
colorlink	plainpages	1412
958	run	10, 1443
colorlinks	runborder	13
944	runborderstyle	14, 1072
colormenu	runcolor	958
958		
colorrun		
958		
colorscheme		
1, 1370		
colorurl		
958		
debug		
1394		
destlabel		
13		
draft		
1394		
extension		
13, 1412		
file		
10, 1443		
fileborderstyle		
14, 1072		
filecolor		
958		

\msg_new:nnnn	15, 26, 37	\pdf_pageobject_ref:n	158, 1875, 1900, 2560
\msg_warning:nn	171, 1390	\pdf_string_from_unicode:nnN	546
\msg_warning:nnn	180, 638, 879	\pdf_version:	1536, 1655, 1684, 1718, 1733, 1759, 1784, 1819, 1835, 1940, 1967
\msg_warning:nnnn	1234, 1275, 1488, 1532, 1566, 1589, 1651, 1680, 1714, 1729, 1742, 1755, 1780, 1815, 1831, 1858, 1889, 1922, 1936, 1963, 1997, 2079	\pdf_version_compare:NnTF	1526, 1617, 1639, 1675, 1708, 1723, 1749, 1774, 1802, 1825, 1930, 1957
\msg_warning:nnnnn	191, 1306, 1340, 1432, 1508, 1545, 1632, 1665, 1690, 1766, 1793, 1844, 1949, 1976, 2240, 2344, 2363, 2382, 2393	\pdf_version_compare_p:Nn	813, 1213, 1262
N		\pdf_version_major:	168, 1214, 1237, 1263, 1279
naturalnames (hypersetup key)	1412	\pdf_version_minor:	167, 1237, 1279
nesting (hypersetup key)	13	pdfannot commands:	
\newcommand	165, 3437, 3439, 3441	\pdfannot_box:nnnn	2595, 2706, 2765, 2816, 2851, 2915
\newcount	2544, 2729	\pdfannot_box_ref_last:	2450, 2466
\NewDocumentCommand	2406	\pdfannot_dict_put:nnn	718, 777, 825, 864, 1004, 1039, 1063, 1085, 1109, 1291, 1322
\NewExpandableDocumentCommand	382	\pdfannot_dict_remove:nn	994, 1034, 1055, 1080, 1101, 1299, 1332
\newlength	3456	\pdfannot_link:nnn	719, 778, 826, 865
\noexpand	2627, 2628, 2629, 2630, 3008	\pdfannot_link_goto_begin:nw	651
\nolinkurl	4	\pdfannot_link_goto_end:	656
O		\pdfannot_link_margin:n	8, 1570
ocgcolorfile (hypersetup key)	1212	\c_pdfannot_link_types_seq	514
ocgcolorlink (hypersetup key)	1212	pdfauthor (hypersetup key)	2096
ocgcolorlinks (hypersetup key)	1212	\pdfbookmark	3442
ocgcolormenu (hypersetup key)	1212	pdfborder (hypersetup key)	13
ocgcolorrun (hypersetup key)	1212	pdfborderstyle (hypersetup key)	14, 1072
ocgcolorurl (hypersetup key)	1212	pdfcreationdate (hypersetup key)	14, 2179
P		pdfcreator (hypersetup key)	2096
pageanchor (hypersetup key)	1412	\pdfdest	11
\paperwidth	3	pdfdict commands:	
\PassOptionsToPackage	447, 1404, 1409	\pdfdict_new:n	533, 696, 733, 791, 850
pdf commands:		\pdfdict_put:nnn	238, 245, 534, 697, 698, 709, 712, 734, 735, 747, 753, 763, 773, 792, 793, 804, 819, 851, 852, 862, 1596, 1597, 1601, 1602, 2341, 2351, 2355, 2358, 2360, 2379, 2388, 2390
\pdf_bdcobject:nn	1198, 1201	\pdfdict_remove:nn	235, 816, 1606, 1607
\pdf_destination:nn	26, 156, 602	\pdfdict_use:n	718, 776, 825, 831, 864, 869, 2330
\pdf_destination:nnnn	595	pdfencoding (hypersetup key)	1384
\pdf_emc:	1200, 1206	\pdfescapestring	2997
\pdf_link_user:nnn	2736	pdffile commands:	
\pdf_name_from_unicode_e:n	246, 747, 863	\pdffile_embed_file:nnn	737, 748
\pdf_object_if_exist:nTF	745	PDFForm internal commands:	
\pdf_object_new:n	1120, 1121, 1122, 1123, 2930, 2931, 2932	\PDFForm_hypName	3036, 3039, 3043, 3046
\pdf_object_ref:n	157, 756, 1126, 1128, 1150, 1151, 1154, 1158, 1163, 1168, 1173, 2521, 2523, 2988	pdfinfo (hypersetup key)	2246
\pdf_object_ref_last:	777, 2332		
\pdf_object_unnamed_write:nn	776, 2328		
\pdf_object_write:nnn	1124, 1130, 1140, 1152, 2933, 2975, 2982		

pdfkeywords (hypersetup key)	<u>2096</u>	\prg_generate_conditional_- variant:Nnn	2555
pdflang (hypersetup key)	<u>14</u> , <u>2085</u>	prop commands:	
pdflinkmargin (hypersetup key)	<u>14</u>	\prop_const_from_keyval:Nn	471, 479, 3589, 3604, 3619, 3634, 3649, 3664, 3680, 3695
pdfmanagement commands:		\prop_gput:Nnn	12, 2423, 2430
\pdfmanagement_add:nn	1579	\prop_if_empty:Ntf	2499
\pdfmanagement_add:nnn		\prop_item:Nn	2516
.	569, 1150, 1151, 1173, 1460, 1480, 1496, 1500, 1520, 1528, 1557, 1612, 1622, 1646, 1660, 1672, 1677, 1702, 1710, 1725, 1751, 1776, 1810, 1827, 1873, 1898, 1912, 1932, 1959, 1988, 2091, 2111, 2116, 2137, 2155, 2166, 2190, 2205, 2227, 2313, 2332, 2496, 2513, 2520, 2522, 2524, 2530	\prop_map_inline:Nn	922, 986, 1026, 1053, 1061, 1072, 1099, 1107, 1176, 1284, 1320, 1330, 1374, 1443, 2494, 2501
\pdfmanagement_if_active_p: . . .	425	\prop_new:N	2412, 2413
\pdfmanagement_remove:nn		property commands:	
.	1457, 1476, 1484, 1504, 1516, 1541, 1553, 1561, 1575, 1584, 1628, 1643, 1662, 1699, 1738, 1763, 1789, 1806, 1840, 1870, 1895, 1908, 1917, 1945, 1972, 1984, 1993, 2120, 2149, 2187, 2202, 2310, 2323, 2534	\property_if_recorded:nn	2555
\pdfmanagement_show:n	2497	\property_if_recorded:nnTF . . .	2558
pdfmeta commands:		\property_record:nn	453
\pdfmeta_standard_verify:nnTF . . .		\property_ref:nn	2562
.	859, 2752, 2843	\property_ref_undefined_warn:nn	2553
\pdfmeta_standard_verify:nTF . . .		\protect	672, 723, 730, 783, 789, 835, 841, 884, 3458
.	2526, 3104	\providecommand	163, 164, 2437, 2440, 2605, 2993
pdfmetadate (hypersetup key)	<u>14</u> , <u>2179</u>	\ProvidesFile	3
pdfmoddate (hypersetup key)	<u>14</u> , <u>2179</u>		
pdfproducer (hypersetup key)	<u>2096</u>		
pdfremotestartview (hypersetup key) . .	<u>11</u>		
pdfstartview (hypersetup key)	<u>11</u>		
\pdfstringdef	<u>8</u> , <u>9</u> , <u>22</u> , <u>28</u> , <u>3053</u>		
\pdfstringdefDisableCommands			
.	383, 384, 385		
pdfsubject (hypersetup key)	<u>2096</u>		
pdftitle (hypersetup key)	<u>2096</u>		
pdftrapped (hypersetup key)	<u>2218</u>		
pdfversion (hypersetup key)	<u>1384</u>		
pdfview (hypersetup key)	<u>11</u> , <u>2003</u>		
pdfxform commands:			
\pdfxform_if_exist:nTF			
.	2692, 2804, 2888		
\pdfxform_new:nnn			
.	2408, 2694, 2807, 2811, 2890, 2902		
\pdfxform_ref:n			
.	2716, 2717, 2823, 2824, 3135, 3136		
\phantom	2911		
\phantomsection	<u>12</u> , <u>19</u> , <u>3527</u>		
plainpages (hypersetup key)	<u>1412</u>		
prg commands:			
\prg_do_nothing:	410, 421		

<code>\seq_if_empty:NTF</code>	2127	<code>\@Form</code>	2488
<code>\seq_item:Nn</code>	1855, 1886, 2008, 2056, 2058, 2068, 2073, 2133	<code>\@PushButton</code>	2744
<code>\seq_map_inline:Nn</code>		<code>\@Reset</code>	2833
. 506, 510, 514, 958, 1259, 1357, 2511		<code>\@Submit</code>	2785
<code>\seq_new:N</code>	459	<code>\@TextField</code>	2571
<code>\seq_put_right:Nn</code>		<code>\@bookmarksopenlevel</code>	1471
. 2045, 2046, 2047, 2503		<code>\@bsphack</code>	452
<code>\seq_set_split:Nnn</code>	2007	<code>\@chapter</code>	3500, 3501
<code>\seq_sort:Nn</code>	2505	<code>\@currDisplay</code>	2620, 2688
<code>\setbox</code>	376	<code>\@currValue</code>	2686, 2716
<code>\setlength</code>	3477, 3548	<code>\@currentHref</code>	
<code>\setpdflinkmargin</code>	8	. 3481, 3494, 3505, 3535, 3552	
<code>\settowidth</code>	2620	<code>\@curropt</code> ..	2605, 2615, 2618, 2682, 2685
<code>\show</code>	2680	<code>\@empty</code> ...	2415, 2416, 2452, 2539, 2540, 2541, 2573, 2574, 2590, 3058, 3062, 3066, 3070, 3074, 3078, 3082, 3086, 3090, 3094, 3238, 3269, 3270
sort commands:		<code>\@endForm</code>	2539
<code>\sort_return_same:</code>	2509	<code>\@esphack</code>	454
<code>\sort_return_swapped:</code>	2508	<code>\@firstofone</code> ..	2443, 2999, 3004, 3021
<code>\space</code>	2688, 3220, 3237, 3238, 3268, 3269, 3343, 3375, 3378	<code>\@firstoftwo</code>	3542
<code>\spacefactor</code>	585, 607	<code>\@for</code>	2616, 2683
<code>\stockwidth</code>	3	<code>\@gobbletwo</code>	3491
str commands:		<code>\@hyper@launch</code>	37, 845
<code>\c_colon_str</code>	845	<code>\@ifnextchar</code>	275
<code>\c_hash_str</code>	714	<code>\@ifundefined</code> ..	3013, 3472, 3499, 3509
<code>\c_percent_str</code>	715	<code>\@mainaux</code>	2436, 2439, 2459, 2478
<code>\str_case:nnTF</code>	2106, 3400	<code>\@mkboth</code>	3491
<code>\str_case_e:nnTF</code>	2008	<code>\@ne</code>	2548
<code>\str_compare:nNnTF</code>	2507	<code>\@nil</code>	3011, 3024, 3030
<code>\str_gset:Nn</code>	983	<code>\@part</code>	3518, 3525
<code>\str_gset_eq:NN</code>	556	<code>\@pdfauthor</code>	23
<code>\str_head:n</code>	2229, 2234	<code>\@pdfborder</code>	163
<code>\str_if_eq:nnTF</code>		<code>\@pdfborderstyle</code>	164
. 592, 2135, 2153, 2163, 3102		<code>\@savsf</code>	585, 607
<code>\str_if_eq_p:nn</code>	811, 1214, 1263	<code>\@schapter</code>	3488, 3489
<code>\str_lowercase:n</code>		<code>\@secondoftwo</code>	3544
. 2008, 2158, 2207, 2230, 2235		<code>\@sect</code>	3539, 3540
<code>\str_new:N</code>	461, 466, 467, 505	<code>\@spart</code>	3531, 3532
<code>\str_set:Nn</code>	538	<code>\@ssect</code>	3474, 3475
<code>\str_set_convert:Nnnn</code>	8	<code>\@tempdima</code> ..	2614, 2621, 2623, 2660, 2661, 2662, 2666, 2667, 2668, 2669
<code>\str_set_eq:NN</code>	558	<code>\@tempdimb</code>	2620, 2621
<code>\str_tail:n</code>	2230, 2235	<code>\@typeset@protect</code>	3458
<code>\str_uppercase:n</code>	2223, 2229, 2234	<code>\BKM@color</code>	435
<code>\string</code>	2437, 2440, 2461, 2479, 3013, 3017, 3028	<code>\c@secnumdepth</code>	3508, 3526, 3541
<code>\subpdfbookmark</code>	3438	<code>\calc@bm@number</code>	3450
sys commands:		<code>\check@bm@number</code>	3449
<code>\c_sys_backend_str</code>	3400	<code>\define@key</code>	429
T			
T _E X and L ^A T _E X 2 _ε commands:			
<code>\@BOOKMARK</code>	3444	<code>\Fld@additionalactions</code> ..	3100, 3170, 3204, 3240, 3277, 3324, 3364, 3393
<code>\@CheckBox</code>	2866	<code>\Fld@align</code>	3129, 3215, 3290
<code>\@ChoiceMenu</code>	2606	<code>\Fld@altname</code> ..	3037, 3039, 3111, 3113

\Fld@annotflags	3124, 3177, 3210, 3246, 3285, 3334, 3369	\Fld@submitflags	3362
\Fld@annotnames	3109, 3126, 3179, 3212, 3248, 3287, 3336, 3371	\Fld@validate@code	3066, 3068
\Fld@bcolor	3148, 3150, 3231, 3233, 3262, 3264, 3298, 3311, 3313	\Fld@value	2574, 2590, 2609, 2650, 2651, 3323
\fld@bcolor	3219	\Fld@width	2575, 2589, 2596, 2600, 2610, 2649, 2660, 2662, 2707, 2721, 2739, 2757, 2787, 2798, 2835, 2847, 2871, 2881, 2916
\Fld@bordercolor	3144, 3146, 3188, 3197, 3199, 3218, 3227, 3229, 3258, 3260, 3296, 3307, 3309, 3342, 3350, 3352, 3377, 3385, 3387	\Fld@X@additionalactions	3057, 3102, 3105
\Fld@borderstyle	3130, 3183, 3216, 3252, 3291, 3340, 3391	\Form@action	3360
\Fld@borderwidth	2667, 2668, 3130, 3183, 3216, 3252, 3291, 3340, 3391	\H@old@part	3518, 3529
\Fld@calculate@code	2452, 3070, 3072	\H@old@schapter	3488, 3497
\Fld@calculate@sortkey	2463	\H@old@ssect	3539, 3550, 3559
\Fld@cbsymbol	3152	\H@old@spart	3531, 3537
\Fld@charsize	2666, 3156, 3237, 3268, 3319, 3375	\H@old@sssect	3474, 3479
\Fld@checkequals	2618, 2685	\href@	275, 281
\Fld@choices	3239	\href@split	281, 282
\Fld@color	3157, 3159, 3238, 3269, 3320	\Hy@abspage	579
\Fld@default	2573, 2590, 2608, 2651, 2680, 2681, 2869, 3270, 3274, 3275, 3322	\Hy@activeanchorfalse	630
\Fld@flags	3128, 3181, 3214, 3250, 3289, 3338, 3373	\Hy@activeanchortrue	621
\Fld@format@code	3062, 3064	\Hy@AtBeginDocument	2417, 2433
\Fld@height	2576, 2597, 2600, 2611, 2669, 2708, 2721, 2739, 2788, 2836, 2872, 2917	\Hy@bookmarkstype	1472
\Fld@keystroke@code	3058, 3060	\Hy@chapapp	3492, 3503
\Fld@listcount	2679, 2687, 2722, 2729	\Hy@colorlink	33
\Fld@mappingname	3041, 3043, 3115, 3117	\Hy@currentbookmarklevel	3431
\Fld@maxlen	3325, 3327	\Hy@DisableOption	173
\Fld@menulength	2613, 2619, 2666	\Hy@drafttrue	1403
\Fld@name	2572, 2607, 2745, 2868, 3036, 3110	\Hy@escapeform	2594, 2691, 2735, 2763, 2802, 2849, 2887, 2995, 3002, 3005
\Fld@onblur@code	3078, 3080	\Hy@escapestring	2997, 2999, 3004, 3007, 3008, 3011, 3012, 3021, 3024, 3060, 3064, 3068, 3072, 3076, 3080, 3084, 3088, 3092, 3096, 3152, 3203, 3266, 3322, 3323, 3360
\Fld@onclick@code	3203	\Hy@finaltrue	1408
\Fld@onenter@code	3090, 3092	\Hy@FormObjects	2493, 2928, 2990
\Fld@onexit@code	3094, 3096	\Hy@gtemp	3053, 3055
\Fld@onfocus@code	3074, 3076	\Hy@href	275
\Fld@onmousedown@code	3082, 3084	\Hy@href@nextactionraw	239
\Fld@onmouseup@code	3086, 3088	\Hy@href@page	227
\Fld@pageobjref	2541, 2556, 2993, 3125, 3178, 3211, 3247, 3286, 3335, 3370	\Hy@linkfileprefix	1417
\Fld@radiosymbol	3266	\Hy@linktoc	1428
\Fld@rotation	3140, 3142, 3186, 3193, 3195, 3217, 3223, 3225, 3254, 3256, 3294, 3303, 3305, 3341, 3346, 3348, 3376, 3381, 3383	\Hy@MakeCurrentHref	200
		\Hy@MakeCurrentHrefAuto	3476, 3492, 3503, 3533, 3547
		\Hy@next	3502, 3511, 3514
		\Hy@numberline	166, 3432
		\Hy@org@chapter	3500, 3515
		\Hy@OutlineName	3448
		\Hy@OutlineRerunCheck	3446
		\Hy@pdfmajorversion	168
		\Hy@pdfminorversion	167

\Hy@pdfstringtrue	29, 551	\hyper@link	33, 659
\Hy@pdfversion	3048	\hyper@linkend	33, 687
\Hy@pstringdef	561, 3028	\hyper@linkfile	325, 738
\Hy@PutCatalog	567	\hyper@linklaunch ..	36, 342, 795, 848
\Hy@raisedlink	3466, 3493, 3504, 3534	\hyper@linknamed	37, 176, 854
\Hy@RestoreLastskip	606	\hyper@linkstart	33, 675
\Hy@ReturnAfterFi	7, 3029	\hyper@linkurl	308, 369, 700
\Hy@safe@activetrue	716, 2455	\hyper@normalise	
\Hy@SaveLastskip	586	275, 304, 320, 337, 363, 376
\Hy@secnum@part	3521, 3523, 3526	\HyPL@Labels	569, 579
\Hy@SectionAnchorHref	3457, 3481, 3552	\HyPL@storePageLabel	567
\Hy@SectionHShift		\HyPsd@SanitizeForOutFile	3435
.....	3456, 3465, 3477, 3548	\if@filesw	2435, 2458, 2477
\Hy@StepCount	2619, 2687	\ifFld@combo	2634, 2664
\Hy@temp	2456, 2471, 3010, 3011	\ifFld@hidden	2589, 2649
\Hy@unicodefalse	3050	\ifFld@multiline	2577
\Hy@VerboseAnchor	587	\ifFld@popdown	2635
\Hy@VerboseLinkStart	663, 679	\ifFld@radio	2631, 2654
\Hy@VerboseLinkStop		\ifHy@implicit	3452
.....	670, 693, 726, 785, 837, 875	\ifHy@pdfescapeform	2996, 3006
\Hy@VersionChecked	159	\ifHy@unicode	3049
\Hy@WrapperDef	583	\kv@set@family@handler	179
\Hy@xspace@end	667, 725, 784, 836, 874	\m@ne	3508, 3523
\HyAnn@AbsPageLabel		\OBJ@OCG@view	165
.....	2540, 2546, 2593,	\p@	2623
	2690, 2734, 2762, 2801, 2848, 2886	\pdf@ifdraftmode	2491
\HyAnn@Count	2544, 2545, 2548, 2549,	\PDF@SetupDoc	160
	2550, 2551, 2552, 2553, 2558, 2562	\PDFForm@Check	2887, 2919, 3121
\HyField@AddToFields ..	2446, 2601,	\PDFForm@List	2735, 2738, 3208
	2723, 2740, 2771, 2827, 2856, 2920	\PDFForm@Name	
\HyField@afields	2415	2587, 2648, 2751, 2877, 3035
\HyField@AfterAuxOpen	2417, 2443, 2476	\PDFForm@Push	2763, 2769, 3174
\HyField@AuxAddToCoFields		\PDFForm@Radio	2691, 2711, 3243
.....	2428, 2440, 2461	\PDFForm@Reset	2849, 2855, 3367
\HyField@AuxAddToFields		\PDFForm@Submit	2802, 2821, 3332
.....	2421, 2437, 2479	\PDFForm@Text	2594, 2599, 3282
\HyField@cofields	2416	\protected@edef	635
\HyField@FlagsCheckBox	2878	\strip@pt ..	3156, 3237, 3268, 3319, 3375
\HyField@FlagsChoice	2659	\toks@	3479, 3485, 3550, 3556
\HyField@FlagsPushButton		\tw@	3480, 3485, 3551, 3556
.....	2754, 2794, 2845	\url@	376, 379
\HyField@FlagsRadioButton	2655	\Url@def	377
\HyField@FlagsSubmit	2795	\Url@HyperHook	377
\HyField@FlagsText	2588	\XR@ext	1414
\HyField@PDFChoices	2732	\z@	376, 2545, 2614, 3217, 3223,
\HyField@SetKeys	2584, 2625,		3254, 3341, 3346, 3376, 3381, 3521
	2628, 2647, 2747, 2790, 2838, 2873	\zref@labelbylist	2550
\HyPat@ObjRef ..	3404, 3411, 3418, 3425	\zref@labelbyprops	2549
\hyper@@link	283	\zref@refused	2551
\hyper@anchor	609	\texorpdfstring	29
\hyper@anchorend		text commands:	
.....	609, 3467, 3494, 3505, 3535	\text_expand:n	743
\hyper@anchorstart		\text_purify:n	8, 538
.....	609, 3467, 3494, 3505, 3535	\textsf	2809, 2813

<code>\the</code>	579,	501, 502, 503, 615, 624, 665, 681,
	2549, 2550, 2551, 2552, 2553, 2558,	743, 1851, 1855, 1859, 1882, 2017,
	2562, 2662, 2666, 2669, 3485, 3556	2020, 2027, 2030, 2031, 2032, 2033,
<code>\theH...</code>	12	2034, 2035, 2040, 2044, 2080, 2124
<code>\ThisShouldNotHappen</code>	3015	<code>\tl_set_eq:Nn</code>
<code>\title</code>	2	<code>\tl_to_str:N</code>
tl commands:		<code>\tl_to_str:n</code>
<code>\c_space_tl</code>	166, 1127,	token commands:
	1158, 1163, 1168, 2716, 2717, 2724,	<code>\token_to_str:N</code>
	2939, 2941, 2951, 2953, 2955, 2958,	<code>\toks</code>
	2960, 3156, 3159, 3291, 3319, 3320	
<code>\tl_const:Nn</code>	468	U
<code>\tl_gput_right:Nn</code>	579	unicode (hypersetup key)
<code>\tl_gset:Nn</code>	1866, 1886, 1890	<code>\Url</code>
<code>\tl_if_blank:nTF</code>		url (hypersetup key)
	761, 2104, 2147, 2185, 2200, 2308, 2321	<code>\url</code>
<code>\tl_if_empty:nTF</code>		urlborder (hypersetup key)
	... 636, 1455, 3144, 3148, 3157, 3320	urlborderstyle (hypersetup key) .
<code>\tl_if_empty:nTF</code>		urlcolor (hypersetup key)
	... 233, 992, 1032, 1051,	<code>\urldef</code>
	1078, 1097, 1641, 1697, 1804, 2089	<code>\Urlfont</code>
<code>\tl_if_empty_p:N</code>	1868, 1893	<code>\UrlSpecial</code>
<code>\tl_if_eq:NnTF</code>	2015	use commands:
<code>\tl_if_exist:nTF</code>	3197, 3307, 3311	<code>\use:n</code>
<code>\tl_if_exist_p:N</code>	3188, 3296, 3298	
<code>\tl_if_head_eq_charcode:nNTF</code>	890, 907	V
<code>\tl_new:N</code> ..	206, 207, 208, 209, 458,	verbose (hypersetup key)
	462, 463, 464, 465, 488, 489, 490,	
	491, 492, 493, 494, 495, 496, 504, 609	W
<code>\tl_set:Nn</code>		<code>\write</code>
	... 226, 227, 239, 270, 273, 299,	
	302, 358, 361, 388, 498, 499, 500,	X
		<code>\x</code>