

The `hyperref-generic` module

A generic driver for `hyperref`

The L^AT_EX Project*

Version 0.95w, released 2023-02-20

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

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

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

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

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

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

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

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

1 Avoiding transition problems

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

- Package options are processed at the end of the driver, Class options are ignored. But not every option already works as package options, in some cases `hyperref` interferes. So it is recommended for most options—with the exception of a few mentioned below in section 9—to set them in `\hypersetup`, not as package option.
- This driver uses the `l3color` module for the colors. All colors defined with `\color_set:nn` or `\color_set:nnn` will work. Colors defined with `xcolor` will work if they don't use one of the special color models not supported by `l3color` as `pdfmanagement-firstaid` contains a patch for `xcolor`. If the package `color` is used it is currently recommended to define colors after `hyperref`.
- Load a color package or `graphicx` to get the right page sizes.
- Report problems! Only known problem can be resolved.

2 Bookmarks / outlines

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

3 “Metadata”

“Metadata”, informations about the document, are stored in a PDF in two places: The `/Info` dictionary and the XMP-metadata. `hyperref` only handles the `/Info` dictionary. The XMP-metadata 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 comm
```

- 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/graphicx`, `geometry`, the KOMA-classes, memoir, ... all try to set this.

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

6 Commands to create “external” references

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

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

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

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

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

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

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

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

6.1 Special problem: links to files

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

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

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

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

6.2 Splits

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

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

For this reason three new commands have been create:

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

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

6.3 Options

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

key	applicable commands	note
urlencode	<code>\hrefurl</code>	if set the code will convert the argument to percent encoding. This allows non-ascii input.
protocol	<code>\hrefurl</code> , <code>\url</code>	This sets a prefix/protocol that is added to the url, see below.
format	<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.
destination	<code>\href</code> , <code>\hrefpdf</code>	A destination name in the PDF
page	<code>\href</code> , <code>\hrefpdf</code>	destination page, default: 1
pdfremotestartview	<code>\href</code> , <code>\hrefpdf</code>	start view, default: Fit
ismap	<code>\href</code> , <code>\hrefurl</code>	see PDF reference
afrelationship	<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.
run-parameter	<code>\hreflaunch</code>	run parameter (see the PDF reference)
nextactionraw	various	puts a <code>/Next</code> entry in the action dictionary (see the PDF reference)

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

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

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

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

Some main changes are

- The default colors have been changed.

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

7.1 Background information

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

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

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

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

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

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

7.2 New Keys

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

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

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

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

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

The default is `phelype`.

7.3 Public interfaces

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

```

\l_hyp_annot_colorlink_bool,
\l_hyp_annot_colorurl_bool,
\l_hyp_annot_colorfile_bool,
\l_hyp_annot_colorryn_bool,
\l_hyp_annot_colormenu_bool,
\l_hyp_annot_ocgcolorlink_bool,
\l_hyp_annot_ocgcolorurl_bool,
\l_hyp_annot_ocgcolorfile_bool,
\l_hyp_annot_ocgcolorryn_bool,
\l_hyp_annot_ocgcolormenu_bool,

```

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

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

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

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

7.4 Changed behaviour

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

Color keys accept the following input syntax:

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

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

frenchlinks The option `frenchlinks` does nothing at all.

cite colors As mentioned above the support for `citecolor` and `citebordercolor` has been reduced. A package like `hyperref` can't keep track of such semantic contexts like cite, acronym, glossaries and special references and maintain keys for them. The keys are not 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 is 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 `cmk` (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>)	<u>Key</u>	<u>Values</u>	<u>comment / example</u>
<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 <code>dvips</code> it is possible to change links locally, <code>pdflatex</code> and <code>lualatex</code> work by page, with <code>dvipdfmx</code> 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{hgeneric-testphase.def}[2023-02-20 v0.95w %
4   generic Hyperref driver for the LaTeX PDF management testphase bundle]
5
6  \RequirePackage{etoolbox} %why?
7  \ExplSyntaxOn
8  \file_input:n {hyperref-colorschemes.def}
9  \ExplSyntaxOff

```

1 messages

Redirect the message name:

```

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

```

At first a message for the testing of the resource management

```

12 \cs_if_exist:NTF \DocumentMetadata

```

```

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

```

The pdfversion should be set in \DocumentMetadata

```

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

```

A generic message for ignored keys.

```

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

```

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

```

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

```

```

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

```

pdf/A message for not allowed Named actions

```

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

```

A message if the destination name is empty.

```

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

```

A message if the destination check fails

```

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

```

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

```

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

```

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

```

105 \msg_new:nnn
106 { hyp }

```



```

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

```

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

```

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

```

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

```

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

```

2 Variants

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

3 Overwriting/providing commands from hyperref

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

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

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

`\hypercalcbp`

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

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

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

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

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

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

`\Acrobatmenu` should use the new internal link command

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

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

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

```

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

```

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

```

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

```

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

```

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

```

```

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

```

```

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

```

```

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

```

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

```

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

```

make the new commands compatible with `\pdfstringdef`:

```

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

```

4 Compability commands

4.1 Metadata

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

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

4.2 citecolor

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

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

5 Checks

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

TODO

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

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

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

6 Reference and label commands

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

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

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


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

```
455 \prg_new_eq_conditional:Nn \_hyp_ref_if_exist:nn \ref_if_exist:nn { p , T , F, TF }
456 \prg_generate_conditional_variant:Nnn \_hyp_ref_if_exist:nn {en} { p , T , F, TF }
```

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

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

```
469 \cs_generate_variant:Nn \_hyp_ref_check:nn {en}
```

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

```
470 \cs_new:Npn \_hyp_ref_value:nn #1 #2
471 {
472   \ref_value:nn{#1}{#2}
473 }
474 \cs_generate_variant:Nn \_hyp_ref_value:nn {en}
```

(End definition for `_hyp_ref_label:nn` and others.)

7 Variables

7.1 Private temporary variables

At first a few generic tmp variables

```
\l__hyp_tmpa_tl
\l__hyp_tmpa_seq 475 \box_new:N \l__hyp_tmpa_box
\l__hyp_tmpa_int 476 \tl_new:N \l__hyp_tmpa_tl
\l__hyp_tmpa_box 477 \seq_new:N \l__hyp_tmpa_seq
\l__hyp_tmpa_str 478 \int_new:N \l__hyp_tmpa_int
479 \str_new:N \l__hyp_tmpa_str
```

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

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

```
\l__hyp_dest_name_tmpa_tl TODO: document and check use!
\l__hyp_uri_tmpa_tl 480 \tl_new:N \l__hyp_dest_name_tmpa_tl
\l__hyp_filename_tmpa_tl 481 \tl_new:N \l__hyp_uri_tmpa_tl
\_hyp_text_tmpa_str_uu\g__hyp_text_tmpa_str 482 \tl_new:N \l__hyp_filename_tmpa_tl
483 \tl_new:N \l__hyp_para_tmpa_tl
484 \str_new:N \l__hyp_text_tmpa_str
485 \str_new:N \g__hyp_text_tmpa_str
```

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

7.2 Constants

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

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

(End definition for `\c__hyp_dest_undefined_tl`.)

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

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

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

7.3 Variables

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

```
\g__hyp_dest_pdfstartview_tl
\l__hyp_dest_pdfremotestartview_tl
506 \tl_new:N \g__hyp_dest_pdfstartpage_tl
507 \tl_new:N \g__hyp_dest_pdfstartview_tl
508 \tl_new:N \l__hyp_dest_pdfremotestartview_tl
```

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

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

```
\l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_info_print_tl
509 \tl_new:N \l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_dest_tl
510 \tl_new:N \l__hyp_text_enc_info_print_tl
\l__hyp_text_enc_dest_print_tl
511 \tl_new:N \l__hyp_text_enc_dest_tl
\l__hyp_text_enc_file_print_tl
512 \tl_new:N \l__hyp_text_enc_dest_print_tl
\l__hyp_text_enc_para_print_tl
513 \tl_new:N \l__hyp_text_enc_file_print_tl
514 \tl_new:N \l__hyp_text_enc_para_print_tl
```

```

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

```

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

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

```

522 \tl_new:N \l__hyp_dest_pdfview_tl

```

(End definition for `\l__hyp_dest_pdfview_tl`.)

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

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

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

`\g__hyp_bordercolormodel_str` (*color name*)

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

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

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

```

523 \str_new:N \g__hyp_bordercolormodel_str

```

(End definition for `\g__hyp_bordercolormodel_str`.)

7.4 Booleans

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

`\l_hyp_annot_colorurl_bool`

`\l_hyp_annot_colorfile_bool`

`\l_hyp_annot_colorrurl_bool`

`\l_hyp_annot_colormenu_bool`

```

524 \seq_map_inline:Nn \c__hyp_annot_types_seq
525 {
526   \bool_new:c {l_hyp_annot_color#1_bool}
527 }

```

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

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

`\l_hyp_annot_ocgcolorurl_bool`

`\l_hyp_annot_ocgcolorfile_bool`

`\l_hyp_annot_ocgcolorrurl_bool`

`\l_hyp_annot_ocgcolormenu_bool`

```

528 \seq_map_inline:Nn \c__hyp_annot_types_seq
529 {
530   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
531 }

```

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

`\not_Named_bool_1111``\l_hyp_annot_Launch_bool` This booleans are used to disable some link types while keeping others.

```

532 \seq_map_inline:Nn \c_pdfannot_link_types_seq
533 {
534   \bool_new:c {l_hyp_annot_#1_bool}
535   \bool_set_true:c {l_hyp_annot_#1_bool}
536 }

```

(End definition for `\l_hyp_annot_GoTo_bool` `\l_hyp_annot_URI_bool` `\l_hyp_annot_GoToR_bool` `\l_hyp_annot_Named_bool` `\l_hyp_annot_Launch_bool`.)

7.5 Boxes

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

```
537 \box_new:N \l__hyp_dest_box
```

(End definition for `\l__hyp_dest_box`.)

7.6 Regex

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

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

(End definition for `\c__hyp_dest_startview_regex`.)

7.7 PDF dictionaries

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

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

(End definition for `l__hyp_page/Trans`.)

8 PDF string conversion

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

`hyp/text/pdfstring`

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

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

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

`_hyp_text_purify:nN`

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

(End definition for _hyp_text_purify:nN.)

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

`_hyp_text_cleanup:N`

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

(End definition for _hyp_text_cleanup:N.)

The last step converts the string to a PDF encoding. As we have at least two targets (hex and literal) there is an argument. The conversion assumes utf8 input, it is based on `cspdf_string_from_unicode:nnN` in `l3pdftools`.

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

`_hyp_text_string_from_unicode:nN`

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

(End definition for _hyp_text_string_from_unicode:nN.)

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

`_hyp_text_pdfstring:nnN`

```
566 \cs_new_protected:Npn \_hyp_text_pdfstring:nnN #1 #2 #3
567 {
568   \group_begin:
569   \Hy@pdfstringtrue
570   \hook_use:n {hyp/text/pdfstring}
571   \_hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
572   \_hyp_text_cleanup:N      \l__hyp_text_tmpa_str
573   \_hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
574   \str_gset_eq:NN \g__hyp_text_tmpa_str\l__hyp_text_tmpa_str
575   \group_end:
576   \str_set_eq:NN #3 \g__hyp_text_tmpa_str
577 }
578 \cs_generate_variant:Nn \_hyp_text_pdfstring:nnN {xnN,onN,xoN,ooN,noN}
```

```
(End definition for \_hyp\_text\_pdfstring:nnN.)
!!! temporary until all instances are gone
579 \cs\_new\_protected:Npn\Hy@pstringdef #1 #2
580 { \_hyp\_text\_pdfstring:xnN {#2} {utf8/string-raw}#1 }
This is a special version for info keys:
```

_hyp_text_pdfstring_info:nN

```
581 \cs\_new\_protected:Npn \_hyp\_text\_pdfstring\_info:nN #1 #2
582 {
583 \_hyp\_text\_pdfstring:nonN { #1 }{\_hyp\_text\_enc\_info\_print\_tl } #2
584 }
```

(End definition for _hyp_text_pdfstring_info:nN.)

9 Pagelabels

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

The current implementation in hyperref/hyperref drivers:

xetex: hxdetex.def, line 80-110

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

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

dvipdfm: identical to xetex

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

luatex identical to pdftex

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

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

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

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

```
\_hyp\_PageLabels\_gpush:
  \Hy@PutCatalog
  \HyPL@StorePageLabel
585 \cs\_new\_protected:Npn\_hyp\_PageLabels\_gpush:
586 {
587 \pdfmanagement\_add:nnx {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
```

```

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

```

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

10 Core Hyperref Commands

Every hyperref has to define eight core command:

```

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

```

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

10.1 Anchors / destinations

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

```

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

```

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

```

\__hyp_destination:nn

```

```

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

```

```

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

```

(End definition for `__hyp_destination:nn`.)

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

```

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

```

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

10.2 GoTo Links

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

At first the internal link commands:

```

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

```

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

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

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

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

```

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

```

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

```

\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

```

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

```

10.3 URI links

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

```

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

```

```

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

```

10.4 GoToR Links files

```

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

```

```

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

```

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

```

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

```

10.5 Launch links

We define `\hyper@linklaunch` for naming consistency

```

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

```

```

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

```

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

```

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

```

```

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

```

10.6 Named links (menu)

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

```

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

```

11 Link decorations

11.1 Functions to export and select colors

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

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

_hyp_color_select:n _hyp_color_select:n {<color>}

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

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

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

(End definition for _hyp_color_select:n and _hyp_color_select_aux:wn.)

_hyp_color_set:nn _hyp_color_set:nn {< name >} {<color>}

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

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

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

(End definition for _hyp_color_set:nn and _hyp_color_set_aux:nwn.)

11.2 Textcolor of links

colors are added in the hooks. This means that they can also be removed if needed. They add a group—this isn't needed with `hyperref` code, but could be relevant with low-level annotations.

```

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

```

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

```

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

```

`colorurl` (*setup key*)

`colorlink` (*setup key*)

`colorrurl` (*setup key*)

`colormenu` (*setup key*)

`colorfile` (*setup key*)

`urlcolor` (*setup key*)

`linkcolor` (*setup key*)

`runcolor` (*setup key*)

`menucolor` (*setup key*)

`filecolor` (*setup key*)

`allcolors` (*setup key*)

```

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

```



```

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

```

11.3 Style and color of borders

11.3.1 Border color

The border color is set by link type. The color can be set as rgb (default) or cmyk (unusual). This can be set with the `bordercolormodel` key:

`bordercolormodel` (*setup key*)

```

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

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

```

```

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

```

11.3.2 Borderwidth and -arc

```

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

```

```

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

```

11.3.3 Borderstyle

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

```

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

```

```

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

```

11.4 ocolorlinks

OCG colorlinks need objects and an entry in the catalog. Perhaps the objects need public names to avoid that ocolor2 has to create duplicates? TODO

`_hyp_ocg_init:` This commands write the objects as needed if ocolor links are used. The initialization should happens only once.

```

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

```

```

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

```

(End definition for __hyp_ocg_init:.)

We use like with colors a hook, this allows ocgx to replace it. The implementation is rather simple and uses a box.

```

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

```

```

1225         \group_end:
1226     }
1227 }
1228 }

```

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

```

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

```

```

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

```

11.5 Highlighting

This keys set what happens if you click on a link

```

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

```

```

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

```

11.6 Hiding links

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

```

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

```



```

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

```

11.7 color schemes and settings

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

`colorscheme` (*setup key*)

```

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

```

12 Keys

12.1 Ignored keys

The following are ignored (with or without warnings)

`unicode` (*setup key*)

`pdfencoding` (*setup key*)

`pdfversion` (*setup key*)

```

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

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
debug (setup key) 1411 \keys_define:nn { hyp / setup }
draft (setup key) 1412 {
final (setup key) 1413   ,verbose .legacy_if_set:n = {Hy@verbose}
1414   ,debug .legacy_if_set:n = {Hy@verbose}
1415 }
1416 \keys_define:nn { hyp / setup }
1417 {
1418   ,draft .code:n =
1419   {
1420     \Hy@drafttrue
1421     \PassOptionsToPackage{draft}{bookmark}
1422   }
1423   ,final .code:n =
1424   {
1425     \Hy@finaltrue
1426     \PassOptionsToPackage{final}{bookmark}
1427   }
1428 }

extension (setup key)
hypertextnames (setup key) 1429 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1430 {
pageanchor (setup key) 1431   ,extension .tl_set:N = \XR@ext
linktoc (setup key) 1432   ,extension .initial:n= pdf
linktocpage (setup key) 1433   ,hypertextnames .legacy_if_set:n = {Hy@hypertextnames}
plainpages (setup key) 1434   ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
localanchorname (setup key) 1435   ,localanchorname .legacy_if_set:n = {Hy@localanchorname}
linkfileprefix (setup key) 1436   ,naturalnames .legacy_if_set:n = {Hy@naturalnames}
1437   ,pageanchor .legacy_if_set:n = {Hy@pageanchor}
1438   ,plainpages .legacy_if_set:n = {Hy@plainpages}
1439 }
1440
1441 \keys_define:nn { hyp / setup }
1442 {
1443   ,linktoc .choices:nn = { none, section, all, page }
1444   {
1445     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1446   }
1447   ,linktoc / unknown .code:n =
1448   {
1449     \msg_warning:nxxx { hyp } { unknown-choice }
1450     { linktoc }
1451     { none, section, all, page }
1452     { \exp_not:n {#1} }
1453   }
1454   ,linktocpage .choice:
1455   ,linktocpage / true .meta:n = {linktoc=page}
1456   ,linktocpage / false .meta:n = {linktoc=section}
1457   ,linktocpage .default:n = true

```

```

1458 }
1459
link (setup key) This booleans allow to disable the link types.
url (setup key) 1460 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1461 {
menu (setup key) 1462 \keys_define:nn { hyp / setup }
run (setup key) 1463 {
1464 ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1465 }
1466 }

1467 \keys_define:nn { hyp / setup }
1468 {
1469 ,baseurl .code:n =
1470 {
1471 \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1472 \tl_if_empty:NTF \l__hyp_tmpa_tl
1473 {
1474 \pdfmanagement_remove:nn {Catalog} { URI }
1475 }
1476 {
1477 \pdfmanagement_add:nxx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1478 }
1479 \__hyp_store_metadata:nn {baseurl}{#1}
1480 }
1481 %only false does something ...
1482 ,bookmarks .choice:
1483 ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1484 ,bookmarks / true .code:n = {}
1485 ,bookmarks .default:n = {true}
1486 ,bookmarksnumbered .legacy_if_set:n = {Hy@bookmarksnumbered}
1487 ,bookmarksopen .legacy_if_set:n = {Hy@bookmarksopen}
1488 ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1489 ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1490 ,pdfcenterwindow .choice:
1491 ,pdfcenterwindow / false .code:n =
1492 {
1493 \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1494 }
1495 ,pdfcenterwindow / true .code:n =
1496 {
1497 \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1498 }
1499 ,pdfcenterwindow / .code:n =
1500 {
1501 \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1502 }
1503 ,pdfcenterwindow / unknown .code:n =
1504 {
1505 \msg_warning:nxx { hyp } { no-bool }
1506 { pdfcenterwindow }
1507 { \exp_not:n {#1} }
1508 }

```

```

1509 ,pdfcenterwindow .default:n = true
1510 ,pdfdirection .choice:
1511 ,pdfdirection / L2R .code:n =
1512 {
1513   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{/L2R }
1514 }
1515 ,pdfdirection / R2L .code:n =
1516 {
1517   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{/R2L }
1518 }
1519 ,pdfdirection / .code:n =
1520 {
1521   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1522 }
1523 ,pdfdirection / unknown .code:n =
1524 {
1525   \msg_warning:nxxx { hyp } { unknown-choice+empty }
1526   { pdfdirection }
1527   { L2R , R2L }
1528   { \exp_not:n {#1} }
1529 }
1530 ,pdfdisplaydoctitle .choice:
1531 ,pdfdisplaydoctitle / false .code:n =
1532 {
1533   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1534 }
1535 ,pdfdisplaydoctitle / true .code:n =
1536 {
1537   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1538 }
1539 ,pdfdisplaydoctitle .default:n = true
1540 ,pdfduplex .choices:nn =
1541 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1542 {
1543   \pdf_version_compare:NnTF > {1.6}
1544   {
1545     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1546     { PrintDuplex } { /#1 }
1547   }
1548   {
1549     \msg_warning:nxxx
1550     {hyp}
1551     {ignore-deprecated-or-unknown-option-in-pdf-version}
1552     {pdfduplex}
1553     {\pdf_version:}
1554   }
1555 }%
1556 ,pdfduplex / .code:n =
1557 {
1558   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1559 }
1560 ,pdfduplex / unknown .code:n =
1561 {
1562   \msg_warning:nxxx { hyp } { unknown-choice+empty }

```

```

1563     { pdfduplex }
1564     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1565     { \exp_not:n {#1} }
1566   }
1567 ,pdffitwindow .choice:
1568 ,pdffitwindow / false .code:n =
1569   {
1570     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { FitWindow }
1571   }
1572 ,pdffitwindow / true .code:n =
1573   {
1574     \pdfmanagement_add:nnn {Catalog / ViewerPreferences} { FitWindow } { true }
1575   }
1576 ,pdffitwindow / .code:n =
1577   {
1578     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { FitWindow }
1579   }
1580 ,pdffitwindow .default:n = true
1581 ,pdffitwindow / unknown .code:n =
1582   {
1583     \msg_warning:nxx { hyp } { no-bool }
1584     { pdffitwindow }
1585     { \exp_not:n {#1} }
1586   }
1587 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1588 ,pdflinkmargin .initial:n = {1pt}
1589 ,pdfmenubar .choice:
1590 ,pdfmenubar / true .code:n =
1591   {
1592     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideMenubar }
1593   }
1594 ,pdfmenubar / false .code:n =
1595   {
1596     \pdfmanagement_add:nn {Catalog / ViewerPreferences}
1597     { HideMenubar } { true }
1598   }
1599 ,pdfmenubar / .code:n =
1600   {
1601     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideMenubar }
1602   }
1603 ,pdfmenubar .default:n = true
1604 ,pdfmenubar / unknown .code:n =
1605   {
1606     \msg_warning:nxx { hyp } { no-bool }
1607     { pdfmenubar }
1608     { \exp_not:n {#1} }
1609   }
1610 ,pdfnewwindow .choice:
1611 ,pdfnewwindow / true .code:n =
1612   {
1613     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1614     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1615   }
1616 ,pdfnewwindow / false .code:n =

```

```

1617     {
1618         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1619         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1620     }
1621 ,pdfnewwindow / .code:n =
1622     {
1623         \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1624         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1625     }
1626 ,pdfnonfullscreenpagemode .choices:nn =
1627     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1628     {
1629         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1630         { NonFullScreenPageMode } {/#1}
1631     }
1632 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1633     {
1634         \pdf_version_compare:NnTF < {1.6}
1635         {
1636             %message
1637         }
1638         {
1639             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1640             {NonFullScreenPageMode}{/UseAttachments}
1641         }
1642     }
1643 ,pdfnonfullscreenpagemode / .code:n =
1644     {
1645         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1646     }
1647 ,pdfnonfullscreenpagemode / unknown .code:n =
1648     {
1649         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1650         { pdfnonfullscreenpagemode }
1651         { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1652         { \exp_not:n {#1} }
1653     }
1654 ,pdfnumcopies .code:n =
1655     {
1656         \pdf_version_compare:NnTF > {1.6}
1657         {
1658             \tl_if_empty:nTF {#1}
1659             {
1660                 \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1661             }
1662             {
1663                 \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1664                 {NumCopies}{#1}
1665             }
1666         }
1667     {
1668         \msg_warning:nnxx
1669         {hyp}
1670         {ignore-deprecated-or-unknown-option-in-pdf-version}

```

```

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

```

```

1725     \pdf_version_compare:NnTF > {1.6}
1726     {
1727         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1728         { PickTrayByPDFSize } { true }
1729     }
1730     {
1731         \msg_warning:nxxx
1732         {hyp}
1733         {ignore-deprecated-or-unknown-option-in-pdf-version}
1734         {pdfpicktraybypdfsize}
1735         {\pdf_version:}
1736     }
1737 }
1738 ,pdfpicktraybypdfsize / false .code:n =
1739 {
1740     \pdf_version_compare:NnTF > {1.6}
1741     {
1742         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1743         { PickTrayByPDFSize } { false }
1744     }
1745     {
1746         \msg_warning:nxxx
1747         {hyp}
1748         {ignore-deprecated-or-unknown-option-in-pdf-version}
1749         {pdfpicktraybypdfsize}
1750         {\pdf_version:}
1751     }
1752 }
1753 ,pdfpicktraybypdfsize / .code:n =
1754 {
1755     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1756 }
1757 ,pdfpicktraybypdfsize / unknown .code:n =
1758 {
1759     \msg_warning:nxxx { hyp } { no-bool }
1760     { picktraybypdfsize }
1761     { \exp_not:n {#1} }
1762 }
1763 ,pdfprintarea .choices:nn =
1764 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1765 {
1766     \pdf_version_compare:NnTF < {2.0}
1767     {
1768         \pdfmanagement_add:nxx {Catalog / ViewerPreferences }
1769         { PrintArea } { /#1 }
1770     }
1771     {
1772         \msg_warning:nxxx
1773         {hyp}
1774         {ignore-deprecated-or-unknown-option-in-pdf-version}
1775         {pdfprintarea}
1776         {\pdf_version:}
1777     }
1778 }%

```



```

1779 ,pdfprintarea / .code:n =
1780 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1781 ,pdfprintarea / unknown .code:n =
1782 {
1783   \msg_warning:nxxx { hyp } { unknown-choice+empty }
1784   { pdfprintarea }
1785   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1786   { \exp_not:n {#1} }
1787 }
1788 ,pdfprintclip .choices:nn =
1789 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1790 {
1791   \pdf_version_compare:NnTF < {2.0}
1792   {
1793     \pdfmanagement_add:nxx {Catalog / ViewerPreferences }
1794     { PrintClip } { /#1 }
1795   }
1796   {
1797     \msg_warning:nxxx
1798     {hyp}
1799     {ignore-deprecated-or-unknown-option-in-pdf-version}
1800     {pdfprintclip}
1801     {\pdf_version:}
1802   }
1803 }%
1804 ,pdfprintclip / .code:n =
1805 {
1806   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1807 }
1808 ,pdfprintclip / unknown .code:n =
1809 {
1810   \msg_warning:nxxx { hyp } { unknown-choice+empty }
1811   { pdfprintclip }
1812   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1813   { \exp_not:n {#1} }
1814 }
1815 ,pdfprintpagerange .code:n =
1816 {
1817   \pdf_version_compare:NnTF > {1.6}
1818   {
1819     \tl_if_empty:nTF { #1}
1820     {
1821       \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1822       { PrintPageRange }
1823     }
1824     {
1825       \pdfmanagement_add:nxx {Catalog / ViewerPreferences }
1826       {PrintPageRange}{[#1]}
1827     }
1828   }
1829   {
1830     \msg_warning:nxxx
1831     {hyp}
1832     {ignore-deprecated-or-unknown-option-in-pdf-version}

```

```

1833         {pdfprintpagerange}
1834         {\pdf_version:}
1835     }
1836 }
1837 ,pdfprintscaling .choices:nn =
1838 { None, AppDefault }
1839 {
1840     \pdf_version_compare:NnTF > {1.5}
1841     {
1842         \pdfmanagement_add:nmx {Catalog / ViewerPreferences }
1843         { PrintScaling } { /#1 }
1844     }
1845     {
1846         \msg_warning:nxxx
1847         {hyp}
1848         {ignore-deprecated-or-unknown-option-in-pdf-version}
1849         {pdfprintscaling}
1850         {\pdf_version:}
1851     }
1852 }%
1853 ,pdfprintscaling / .code:n =
1854 {
1855     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1856 }
1857 ,pdfprintscaling / unknown .code:n =
1858 {
1859     \msg_warning:nxxx { hyp } { unknown-choice+empty }
1860     { pdfprintarea }
1861     { None, AppDefault }
1862     { \exp_not:n {#1} }
1863 }
1864 ,pdfremotestartview .code:n =
1865 {
1866     \tl_set:Nx \l__hyp_tmpa_tl {#1-null-null-null~}
1867     \exp_args:NNV
1868     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1869     {
1870         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1871     }
1872     {
1873         \msg_warning:nxxx {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1874         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1875     }
1876 }
1877 ,pdfremotestartview .initial:n = {Fit}
1878 % pdfstartpage is special as it shares code with pdfstartview
1879 ,pdfstartpage .code:n =
1880 {
1881     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1882     \bool_if:nTF
1883     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1884     {
1885         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1886     }

```

```

1887     {
1888         \pdfmanagement_add:nx {Catalog} { OpenAction }
1889         {
1890             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1891         }
1892     }
1893 }
1894 ,pdfstartpage .initial:n =1
1895 ,pdfstartview .code:n =
1896 {
1897     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~}
1898     \exp_args:NNV
1899     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1900     {
1901         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1902     }
1903     {
1904         \msg_warning:nxxx {hyp}{invalid-destination-value}{#1}{pdfstartview}
1905         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1906     }
1907     \bool_if:nTF
1908     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1909     {
1910         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1911     }
1912     {
1913         \pdfmanagement_add:nx {Catalog} { OpenAction }
1914         {
1915             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1916         }
1917     }
1918 }
1919 ,pdfstartview .initial:n = Fit
1920 ,pdftoolbar .choice:
1921 ,pdftoolbar / true .code:n =
1922 {
1923     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1924 }
1925 ,pdftoolbar / false .code:n =
1926 {
1927     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
1928     { HideToolbar } { true }
1929 }
1930 ,pdftoolbar / true .code:n =
1931 {
1932     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1933 }
1934 ,pdftoolbar .default:n = true
1935 ,pdftoolbar / unknown .code:n =
1936 {
1937     \msg_warning:nxxx { hyp } { no-bool }
1938     { pdftoolbar }
1939     { \exp_not:n {#1} }
1940 }

```

```

1941 % pdfview see below.
1942 ,pdfviewarea .choices:nn =
1943 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1944 {
1945   \pdf_version_compare:NnTF < {2.0}
1946   {
1947     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1948     { ViewArea } { /#1 }
1949   }
1950   {
1951     \msg_warning:nxxx
1952     {hyp}
1953     {ignore-deprecated-or-unknown-option-in-pdf-version}
1954     {pdfviewarea}
1955     {\pdf_version:}
1956   }
1957 }%
1958 ,pdfviewarea / .code:n =
1959 {
1960   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1961 }
1962 ,pdfviewarea / unknown .code:n =
1963 {
1964   \msg_warning:nxxx { hyp } { unknown-choice+empty }
1965   { pdfviewarea }
1966   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1967   { \exp_not:n {#1} }
1968 }
1969 ,pdfviewclip .choices:nn =
1970 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1971 {
1972   \pdf_version_compare:NnTF < {2.0}
1973   {
1974     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1975     { ViewClip } { /#1 }
1976   }
1977   {
1978     \msg_warning:nxxx
1979     {hyp}
1980     {ignore-deprecated-or-unknown-option-in-pdf-version}
1981     {pdfviewclip}
1982     {\pdf_version:}
1983   }
1984 }%
1985 ,pdfviewclip / .code:n =
1986 {
1987   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
1988 }
1989 ,pdfviewclip / unknown .code:n =
1990 {
1991   \msg_warning:nxxx { hyp } { unknown-choice+empty }
1992   { pdfviewclip }
1993   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1994   { \exp_not:n {#1} }

```

```

1995     }
1996     ,pdfwindowui .choice:
1997     ,pdfwindowui / true .code:n =
1998     {
1999         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2000     }
2001     ,pdfwindowui / false .code:n =
2002     {
2003         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2004         { HideWindowUI } { true }
2005     }
2006     ,pdfwindowui / .code:n =
2007     {
2008         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2009     }
2010     ,pdfwindowui / unknown .code:n =
2011     {
2012         \msg_warning:nxxx { hyp } { no-bool }
2013         { pdfwindowui }
2014         { \exp_not:n {#1} }
2015     }
2016     ,pdfwindowui .default:n = true
2017 }

```

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

```

2018 \keys_define:nn { hyp / setup }
2019 {
2020     ,pdfview .code:n =
2021     {
2022         \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
2023         \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } } }
2024         {
2025             { xyz }
2026             {
2027                 \int_compare:nNnTF { \seq_count:N \l__hyp_tmpa_seq } > { 1 }
2028                 {
2029                     \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2030                     \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2031                     {
2032                         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2033                     }
2034                     {
2035                         \tl_set:Nx \l__hyp_dest_pdfview_tl
2036                         {
2037                             \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2038                         }
2039                     }
2040                 }
2041             }
2042             \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2043         }
2044     }
2045     { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2046     { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }

```

```

2047 { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh}}
2048 { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv}}
2049 { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2050 { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2051 { fitr }
2052 {
2053   \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2054   {
2055     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2056   }
2057   {
2058     %ensure 4 values ...
2059     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2060     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2061     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2062     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2063     \hbox_set_to_wd:Nnn \l__hyp_dest_box
2064     {
2065       \fp_eval:n
2066       {
2067         round
2068         (
2069           abs
2070           (
2071             \seq_item:Nn\l__hyp_tmpa_seq{4}
2072             -
2073             (\seq_item:Nn\l__hyp_tmpa_seq{2})
2074           ),
2075           3
2076         )
2077       }bp
2078     }{}
2079     \box_set_dp:Nn \l__hyp_dest_box
2080     {
2081       \fp_eval:n
2082       {
2083         round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2084       }bp
2085     }
2086     \box_set_ht:Nn \l__hyp_dest_box
2087     {
2088       \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2089     }
2090   }
2091 }
2092 }
2093 {
2094   \msg_warning:nmmn {hyp}{invalid-destination-value}{#1}{pdfview}
2095   \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2096 }
2097 }
2098 ,pdfview .initial:n = {xyz}
2099 }

```

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.

```
2100 \keys_define:nn { hyp / setup }
2101 {
2102   ,pdflang      .code:n =
2103     {
2104       \tl_if_empty:nF { #1 }
2105       {
2106         \pdfmanagement_add:nnx {Catalog} { Lang } { (#1) }
2107         \AddToDocumentProperties[document]{lang}{#1}
2108       }
2109     }
2110 }
```

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 empty value. If the value begin with an optional argument, we assume a multilanguage
`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*)

```
2111 \regex_new:N\l__hyp_optlang_regex
2112 \regex_set:Nn\l__hyp_optlang_regex {\A\[[A-Za-z\-\]]+\}\{.*\}
2113 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2114 {
2115   \keys_define:nn { hyp / setup }
2116   {
2117     pdf#1 .code:n =
2118     {
2119       \tl_if_blank:nTF {##1}
2120       {
2121         \str_case:nnF { #1 }
2122         {
2123           {creator}
2124           {
2125             \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2126             \pdfmanagement_add:nnx {Info}{Creator}{()}
2127           }
2128           {producer}
2129           {
2130             \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2131             \pdfmanagement_add:nnx {Info}{Producer}{()}
2132           }
2133         }
2134       }
2135       \pdfmanagement_remove:nn {Info}{#2}
2136     }
2137   }
2138   {
2139     \tl_set:Nx\l__hyp_tmpa_tl {\clist_item:nn{##1}{1}}
2140     \exp_args:NNV
```

```

2141         \regex_extract_once:NnN \l__hyp_optlang_regex \l__hyp_tmpa_tl\l__hyp_tmpa_s
2142 \seq_if_empty:NTF\l__hyp_tmpa_seq
2143 {
2144     \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2145 }
2146 {
2147     \exp_args:Nx
2148     \__hyp_text_pdfstring_info:nN {\seq_item:Nn \l__hyp_tmpa_seq{3}}\l__hyp_t
2149 }
2150 \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2151 {
2152     \pdfmanagement_add:nxn {Info}{#2}{\l__hyp_tmpa_str}
2153 }
2154 }
2155 \__hyp_store_metadata:nn {pdf#1}{##1}
2156 }
2157 }
2158 \keys_define:nn { hyp / info }
2159 {
2160     #2 .code:n =
2161     {
2162         \tl_if_blank:nTF {##1}
2163         {
2164             \pdfmanagement_remove:nn {Info}{#2}
2165         }
2166         {
2167             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2168             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2169             {
2170                 \pdfmanagement_add:nxn {Info}{#2}{\l__hyp_tmpa_str}
2171             }
2172         }
2173         \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2174     }
2175     ,unknown .code:n =
2176     {
2177         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2178         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2179         {
2180             \exp_args:Nno
2181             \pdfmanagement_add:nxn {Info}
2182                 { \l_keys_key_str } {\l__hyp_tmpa_str}
2183         }
2184     }
2185 }
2186 }
2187 \__hyp_setup_info_key:nn {author} {Author}
2188 \__hyp_setup_info_key:nn {title} {Title}
2189 \__hyp_setup_info_key:nn {producer} {Producer}
2190 \__hyp_setup_info_key:nn {creator} {Creator}
2191 % ignored key: addtopdfcreator
2192 \__hyp_setup_info_key:nn {subject} {Subject}
2193 \__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) 2194 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2195 {
2196   \keys_define:nn { hyp / setup }
2197   {
2198     pdf#1 .code:n =
2199     {
2200       \tl_if_blank:nTF {##1}
2201       {
2202         \pdfmanagement_remove:nn {Info}{#2}
2203       }
2204       {
2205         \pdfmanagement_add:nnx {Info}{#2}{(##1)}
2206       }
2207       \__hyp_store_metadata:nn {pdf#1}{##1}
2208       \AddToDocumentProperties[document]{#1}{##1}
2209     }
2210   }
2211   \keys_define:nn { hyp / info }
2212   {
2213     #2 .code:n =
2214     {
2215       \tl_if_blank:nTF {##1}
2216       {
2217         \pdfmanagement_remove:nn {Info}{#2}
2218       }
2219       {
2220         \pdfmanagement_add:nnx {Info}{#2}{(##1)}
2221       }
2222       \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2223     }
2224   }
2225 }
2226
2227 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2228 \__hyp_setup_info_date_key:nn {moddate} {ModDate}
2229 \keys_define:nn { hyp / setup }
2230 {
2231   pdfmetadate .code:n = { \__hyp_store_metadata:nn {pdfmetadate}{#1} }
2232 }

```

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

```

2233 \keys_define:nn { hyp / setup }
2234 {
2235   ,pdftrapped .code:n =
2236   {
2237     \exp_args:Nne
2238     \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2239   }
2240   ,pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2241   {
2242     \pdfmanagement_add:nnx {Info}{Trapped}
2243   }

```

```

2244         \str_uppercase:f { \str_head:n { #1 } }
2245         \str_lowercase:f { \str_tail:n { #1 } }
2246     }
2247     \__hyp_store_metadata:nx {pdftrapped}
2248     {
2249         \str_uppercase:f { \str_head:n { #1 } }
2250         \str_lowercase:f { \str_tail:n { #1 } }
2251     }
2252 }
2253 ,_pdftrapped / unknown .code:n =
2254 {
2255     \msg_warning:nxxxx { hyp } { unknown-choice }
2256     { pdftrapped }
2257     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2258     { \exp_not:n {#1} }
2259 }
2260 }

```

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

```

2261 \keys_define:nn { hyp / setup }
2262 {
2263     pdfinfo .code:n =
2264     {
2265         \keys_set:nn { hyp / info } { #1 }
2266     }
2267 }

```

Now we set some default values

```

2268 \keys_set:nn { hyp / setup} {pdfcreator = LaTeX-with-hyperref}
2269 \keys_set:nn { hyp / setup} {pdfauthor = }
2270 \keys_set:nn { hyp / setup} {pdftitle = }
2271 \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:

```

2272 \clist_map_inline:nn
2273 {
2274     ,pdfcopyright
2275     ,pdftype
2276     ,pdflicenseurl
2277     ,pdfauthortitle
2278     ,pdfcaptionwriter
2279     ,pdfmetalang
2280     ,pdfsource
2281     ,pdfdocumentid
2282     ,pdfinstanceid
2283     ,pdfversionid
2284     ,pdfrendition
2285     ,pdfpublication
2286     ,pdfpubtype
2287     ,pdfbytes

```

```

2288 ,pdfnumpages
2289 ,pdfissn
2290 ,pdfeissn
2291 ,pdfisbn
2292 ,pdfbookedition
2293 ,pdfpublisher
2294 ,pdfvolumenum
2295 ,pdfissuenum
2296 ,pdfpagerange
2297 ,pdfdoi
2298 ,pdfurl
2299 ,pdfidentifier
2300 ,pdfsubtitle
2301 ,pdfpubstatus
2302 ,pdfcontactaddress
2303 ,pdfcontactcity
2304 ,pdfcontactregion
2305 ,pdfcontactpostcode
2306 ,pdfcontactcountry
2307 ,pdfcontactphone
2308 ,pdfcontactemail
2309 ,pdfcontacturl
2310 ,pdfdate
2311 }
2312 {
2313   \keys_define:nn { hyp / setup }
2314   {
2315     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2316   }
2317 }
2318

```

12.5 Transitions

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

```

2319 \keys_define:nn { hyp / setup }
2320 {
2321   pdfpageduration .code:n =
2322   {
2323     \tl_if_blank:nTF { #1 }
2324     {
2325       \pdfmanagement_remove:nn {Page}{Dur}
2326     }
2327     {
2328       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2329     }
2330   }
2331 }

```

Transition settings are used by (some) pdf viewers when presenting a pdf in full screen mode. They are added to the page settings and describe the transition from the previous page to current page. Transition setting can be set in the preamble for all pages or in the document for the current and the following pages. Due to the asynchronous page breaking one has to be careful to set it on the right page, e.g. only after a `\newpage`. The

generic driver uses a different syntax than the other hyperref drivers: various transition options can be set by a keyval syntax in the value of `pdfpagetransition`. A typical setting looks e.g. like this

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

The keys allowed in the argument of `pdfpagetransition` are

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

```
2332 \keys_define:nn { hyp / setup }
2333 {
2334   pdfpagetransition .code:n =
2335   {
2336     \tl_if_blank:nTF {#1}
2337     {
2338       \pdfmanagement_remove:nn {Page}{Trans}
2339     }
2340     {
2341       \group_begin:
2342       \keys_set:nn { hyp / trans }{style=R,#1}
2343       \pdf_object_unnamed_write:nx { dict }
2344       {
2345         \pdfdict_use:n {l__hyp_page/Trans}
2346       }
2347       \pdfmanagement_add:nxx {Page}{Trans}{\pdf_object_ref_last:}
2348       \group_end:
2349     }
2350   }
2351 }
2352 \keys_define:nn { hyp / trans }
2353 {
2354   ,style .choices:nn =
2355   {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2356   { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2357   ,style / unknown .code:n =
2358   {
2359     \msg_warning:nxxx { hyp } { unknown-choice }
2360     { trans / style }
2361     { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }

```

```

2362         { \exp_not:n {#1} }
2363     }
2364 ,duration .code:n =
2365     {
2366         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{#1}
2367     }
2368 ,direction .choices:nn =
2369     {H,V}
2370     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2371 ,direction .choices:nn =
2372     {0,90,180,270,315}
2373     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2374 ,direction / None .code:n =
2375     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2376 ,direction / unknown .code:n =
2377     {
2378         \msg_warning:nnxxx { hyp } { unknown-choice }
2379         { trans / direction }
2380         {
2381             H~(horizontal,~only~Split,~Blinds),
2382             V~(vertical,~only~Split,~Blinds),
2383             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2384             90~(bottom~to~top,~only~Wipe),
2385             180~(right~to~left,~only~Wipe),
2386             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2387             315~(top~left~to~bottom,~only~Glitter),
2388             None~(only~Fly)
2389         }
2390         { \exp_not:n {#1} }
2391     }
2392 ,motion .choices:nn =
2393     {I,0}
2394     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2395 ,motion / unknown .code:n =
2396     {
2397         \msg_warning:nnxxx { hyp } { unknown-choice }
2398         { trans / motion }
2399         { I~(inwards) , 0~(outwards) }
2400         { \exp_not:n {#1} }
2401     }
2402 ,scale .code:n =
2403     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2404 ,opaque .choices:nn = {true,false}
2405     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2406 ,opaque / unknown .code:n =
2407     {
2408         \msg_warning:nnxxx { hyp } { unknown-choice }
2409         { trans / B }
2410         { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2411         { \exp_not:n {#1} }
2412     }
2413 % try to set unknown keys as style
2414 ,unknown .code:n =
2415     {

```

```

2416         % warning ...
2417         \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2418     }
2419 }

Finally we process the package option list, to get most keys working
2420 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}
Unfinished Form field code
2421 \NewDocumentCommand \MakeFieldObject { m m }
2422 {
2423     \pdfxform_new:nnn { #2 }{ } { #1 }
2424 }
2425
2426
2427 \prop_new:N    \g__hyp_AcroForm_CoFields_prop
2428 \prop_new:N    \g__hyp_AcroForm_Fields_prop
2429
2430 \let\HyField@afields\ltx@empty
2431 \let\HyField@cofields\ltx@empty
2432 %% UF test for old pdftex removed
2433 %\let\HyField@AuxAddToFields\ltx@gobble
2434 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2435 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2436
2437 % the value doesn't matter, but with a prop we avoid duplicates and it is
2438 % clearly faster than removing them from a sequence
2439 \def\HyField@AuxAddToFields#1
2440 {
2441     \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2442 }%
2443
2444 %fields with empty key get a value too -- lets hope that
2445 %this give the expected behaviour
2446 \def\HyField@AuxAddToCoFields #1 #2
2447 {
2448     \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2449 }
2450
2451 \Hy@AtBeginDocument
2452 {
2453     \if@filesw
2454         \immediate\write\@mainaux{%
2455             \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2456         }%
2457         \immediate\write\@mainaux{%
2458             \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2459         }%
2460     \fi
2461     \let\HyField@AfterAuxOpen\@firstofone
2462 }%
2463
2464 \def\HyField@AddToFields
2465 {
2466     \exp_args:Nx\HyField__hypAddToFields

```

```

2467     {
2468     \pdfannot_box_ref_last:
2469     }
2470 \ifx\Fld@calculate@code\ltx@empty
2471 \else
2472 \begingroup
2473 \Hy@safe@activestruer
2474 \edef\Hy@temp{%
2475 \endgroup
2476 \if@filesw
2477 \write\@mainaux
2478 {
2479 \string\HyField@AuxAddToCoFields
2480 {
2481 \Fld@calculate@sortkey
2482 }
2483 {
2484 \pdfannot_box_ref_last:
2485 }
2486 }
2487 \fi
2488 }%
2489 \Hy@temp
2490 \fi
2491 }%
2492
2493 \def\HyField__hypAddToFields#1{
2494 \HyField@AfterAuxOpen{%
2495 \if@filesw
2496 \write\@mainaux{%
2497 \string\HyField@AuxAddToFields{#1}%
2498 }%
2499 \fi
2500 }%
2501 }%
2502
2503 \ExplSyntaxOff
2504 \ExplSyntaxOn
2505
2506 \def\@Form[#1]
2507 {
2508 \kvsetkeys{Form}{#1}
2509 \pdf@ifdraftmode{}
2510 {
2511 \Hy@FormObjects
2512 \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2513 {
2514 \pdfmanagement_add:nxn { Catalog / AcroForm } { Fields }{##1}
2515 %\pdfmanagement_show:n { Catalog / AcroForm }
2516 }
2517 \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2518 {
2519 \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2520 {

```

```

2521         \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2522     }
2523     \seq_sort:Nn \l__hyp_tmpa_seq
2524     {
2525         \str_compare:nNnTF {##1} > {##2}
2526         { \sort_return_swapped: }
2527         { \sort_return_same: }
2528     }
2529     \seq_map_inline:Nn \l__hyp_tmpa_seq
2530     {
2531         \pdfmanagement_add:nnx { Catalog / AcroForm }
2532         { CO }
2533         {
2534             \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2535         }
2536     }
2537 }
2538 \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2539 {ZaDb} {\pdf_object_ref:n {__hyp/Font/ZaDb} }
2540 \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2541 {Helv} {\pdf_object_ref:n {__hyp/Font/Helv} }
2542 \pdfmanagement_add:nnx {Catalog /AcroForm}
2543 {DA}{(/Helv~10~Tf~0~g)}
2544 \pdfmeta_standard_verify:nTF {form_no_NeedAppearance}
2545 {
2546     \legacy_if:nT { HyField@NeedAppearances }
2547     {
2548         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2549     }
2550 }
2551 {
2552     \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2553 }
2554 }
2555 }
2556 \ExplSyntaxOff
2557 \let\@endForm\ltx@empty
2558 \let\HyAnn@AbsPageLabel\ltx@empty
2559 \let\Fld@pageobjref\ltx@empty
2560
2561 \ExplSyntaxOn
2562 \newcount\HyAnn@Count
2563 \HyAnn@Count=\ltx@zero
2564 \def\HyAnn@AbsPageLabel
2565 {
2566     \global\advance\HyAnn@Count by\ltx@one
2567     %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2568     %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2569     %\zref@refused{HyAnn@\the\HyAnn@Count}%
2570     \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
2571     \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
2572 }%
2573 \def\Fld@pageobjref
2574 {

```



```

2575     \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
2576     {
2577         /P~\pdf_pageobject_ref:n
2578         {
2579             \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
2580         }
2581     }
2582 }
2583 \ExplSyntaxOff
2584 \ExplSyntaxOn
2585 %% check if the attr should be set through
2586 %% hooks.
2587 %% check if options are missing.
2588 \def\@TextField[#1]#2{% parameters, label
2589     \def\Fld@name{#2}%
2590     \let\Fld@default\ltx@empty
2591     \let\Fld@value\@empty
2592     \def\Fld@width{\DefaultWidthofText}%
2593     \def\Fld@height{%
2594         \ifFld@multiline
2595             \DefaultHeightofTextMultiline
2596         \else
2597             \DefaultHeightofText
2598         \fi
2599     }%
2600     \begingroup
2601     \expandafter\HyField@SetKeys\expandafter{%
2602         \DefaultOptionsofText,#1%
2603     }%
2604     \PDFForm@Name
2605     \HyField@FlagsText
2606     \ifFld@hidden\def\Fld@width{1sp}\fi
2607     \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2608     \LayoutTextField{#2}{%
2609         \leavevmode
2610         \HyAnn@AbsPageLabel
2611         \Hy@escapeform\PDFForm@Text
2612         \pdfannot_box:nmmn
2613             {\Fld@width}
2614             {\Fld@height}
2615             {Opt} %is this correct?
2616             {\PDFForm@Text}
2617         \MakeTextField{\Fld@width}{\Fld@height}
2618         \HyField@AddToFields
2619     }%
2620     \endgroup
2621 }
2622 \providecommand\@curropt{}
2623 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2624     \def\Fld@name{#2}
2625     \let\Fld@default\relax
2626     \let\Fld@value\relax
2627     \def\Fld@width{\DefaultWidthofChoiceMenu}
2628     \def\Fld@height{\DefaultHeightofChoiceMenu}

```

```

2629 \begingroup
2630   \Fld@menulength=0 %
2631   \@tempdima\z@
2632   \clist_map_variable:nNn { #3 } \@curropt
2633   %\@for\@curropt:=#3\do
2634   {%
2635     \expandafter\Fld@checkequals\@curropt==\%
2636     \Hy@StepCount\Fld@menulength
2637     \settowidth{\@tempdimb}{\@currDisplay}%
2638     \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2639   }%
2640   \advance\@tempdima by~15\p@
2641   \begingroup
2642     \HyField@SetKeys{#1}
2643   \edef\x{\endgroup
2644     \noexpand\expandafter
2645     \noexpand\HyField@SetKeys
2646     \noexpand\expandafter{%
2647       \expandafter\noexpand\csgname DefaultOptionsof%
2648       \ifFld@radio
2649         Radio%
2650       \else
2651         \ifFld@combo
2652           \ifFld@popdown
2653             PopdownBox%
2654           \else
2655             ComboBox%
2656           \fi
2657         \else
2658           ListBox%
2659         \fi
2660       \fi
2661     \endcsname
2662   }%
2663   }\x
2664   \HyField@SetKeys{#1}%
2665   \PDFForm@Name
2666   \ifFld@hidden\def\Fld@width{1sp}\fi
2667   \ifx\Fld@value\relax
2668     \let\Fld@value\Fld@default
2669   \fi
2670   \LayoutChoiceField{#2}{%
2671     \ifFld@radio
2672       \HyField@FlagsRadioButton
2673       \__hypRadio{#3}%
2674     \else
2675       \begingroup
2676         \HyField@FlagsChoice
2677         \ifdim\Fld@width<\@tempdima
2678           \ifdim\@tempdima<1cm\@tempdima1cm\fi
2679           \edef\Fld@width{\the\@tempdima}%
2680         \fi
2681         \ifFld@combo
2682         \else

```

```

2683         \@tempdima=\the\Fld@menulength\Fld@charsize
2684         \advance\@tempdima by-\Fld@borderwidth bp %
2685         \advance\@tempdima by-\Fld@borderwidth bp %
2686         \edef\Fld@height{\the\@tempdima}%
2687     \fi
2688     \_\hypListBox{#3}%
2689 \endgroup
2690 \fi
2691 }%
2692 \endgroup
2693 }
2694
2695 \def\_\hypRadio#1{%
2696     \Fld@listcount=0-%
2697     %\show\Fld@default
2698     \EdefEscapeName\Fld@default{\Fld@default}%
2699     \clist_map_variable:nNn { #1 } \@curropt
2700     %\@for\@curropt:=#1\do
2701     {%
2702         \expandafter\Fld@checkequals\@curropt==\%
2703         \EdefEscapeName\@currValue{\@currValue}%
2704         \Hy@StepCount\Fld@listcount
2705         \@currDisplay\space
2706         \leavevmode
2707         \HyAnn@AbsPageLabel
2708         \Hy@escapeform\PDFForm@Radio
2709         \pdfxform_if_exist:nF { \_\hyp_xform_Ding }
2710         {
2711             \pdfxform_new:nnn { \_\hyp_xform_Ding } {}
2712             {
2713                 \group_begin:
2714                 \fontfamily{pzd}
2715                 \fontencoding{U}
2716                 \fontseries{m}
2717                 \fontshape{n}
2718                 \selectfont
2719                 \char123
2720                 \group_end:
2721             }
2722         }
2723         \pdfannot_box:nnnx
2724         {\Fld@width}
2725         {\Fld@height}
2726         {0pt} %is this correct?
2727         {
2728             \PDFForm@Radio
2729             /AP
2730             <<
2731             /N
2732             <<
2733             /\@currValue\c_space_tl \pdfxform_ref:n { \_\hyp_xform_Ding }
2734             %/Off \c_space_tl \pdfxform_ref:n { \_\hyp_xform_DingOff } %hm
2735             >>
2736             >>

```

```

2737     }
2738     {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2739     \int_compare:nNnT { \Fld@listcount} = { 1 }
2740     { \HyField@AddToFields }
2741     \c_space_tl % deliberate space between radio buttons
2742     % to do: --> should be configurable
2743 }%
2744 }
2745
2746 \newcount\Fld@listcount
2747 \def\__hypListBox#1
2748 {
2749   \HyField@PDFChoices{#1}
2750   \mode_leave_vertical:
2751   \HyAnn@AbsPageLabel
2752   \Hy@escapeform\PDFForm@List
2753   \pdf_link_user:nnn
2754     {widget} %perhaps we need more types??
2755     {\PDFForm@List}
2756     {\MakeChoiceField{\Fld@width}{\Fld@height}}
2757   \HyField@AddToFields
2758 }
2759
2760
2761 \def\@PushButton[#1]#2{% parameters, label
2762   \def\Fld@name{#2}%
2763   \group_begin:
2764     \exp_args:No\HyField@SetKeys
2765     {
2766       \DefaultOptionsofPushButton,#1
2767     }
2768     \PDFForm@Name
2769     \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2770     {
2771       \HyField@FlagsPushButton
2772       \legacy_if:nT {Fld@hidden}
2773       {
2774         \def\Fld@width{1sp}
2775       }
2776       \LayoutPushButtonField
2777       {
2778         \mode_leave_vertical:
2779         \HyAnn@AbsPageLabel
2780         \Hy@escapeform\PDFForm@Push
2781         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2782         \pdfannot_box:nnnn
2783           {\box_wd:N\l_tmpa_box}
2784           {\box_ht:N\l_tmpa_box}
2785           {\box_dp:N\l_tmpa_box} %is this correct?
2786           {\PDFForm@Push}
2787           {\box_use:N\l_tmpa_box}
2788         \HyField@AddToFields
2789       }
2790     }

```

```

2791     {
2792     \msg_error:nn { hyp }{ pdfa-no-push-button }
2793     \LayoutPushButtonField
2794     {
2795         \mode_leave_vertical:
2796         \MakeButtonField{#2}
2797     }
2798     }
2799     \group_end:
2800 }
2801
2802 \def\@Submit[#1]#2
2803 {
2804     \def\Fld@width {\DefaultWidthofSubmit}
2805     \def\Fld@height{\DefaultHeightofSubmit}
2806     \group_begin:
2807     \exp_args:No\HyField@SetKeys
2808     {
2809         \DefaultOptionsofSubmit,#1
2810     }
2811     \HyField@FlagsPushButton
2812     \HyField@FlagsSubmit
2813     \legacy_if:nT { Fld@hidden }
2814     {
2815         \def\Fld@width{1sp}
2816     }
2817     \mode_leave_vertical:
2818     \HyAnn@AbsPageLabel
2819     \Hy@escapeform\PDFForm@Submit
2820     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2821     \pdfxform_if_exist:nF
2822     { __hyp_xform_Submit }
2823     {
2824         \pdfxform_new:nnn { __hyp_xform_Submit }{}
2825         {
2826             \fbox{\color_select:n{yellow}\textsf{Submit}}
2827         }
2828         \pdfxform_new:nnn { __hyp_xform_SubmitP }{}
2829         {
2830             \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2831         }
2832     }
2833     \pdfannot_box:nnnn
2834     {\box_wd:N\l_tmpa_box}
2835     {\box_ht:N\l_tmpa_box}
2836     {\box_dp:N\l_tmpa_box} %is this correct?
2837     {
2838         \PDFForm@Submit
2839         /AP<<
2840         /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2841         /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2842         >>
2843     }
2844     \HyField@AddToFields

```

```

2845     \box_use:N\l_tmpa_box
2846
2847   \group_end:
2848 }
2849
2850 \def\@Reset[#1]#2
2851 {
2852   \def\Fld@width {\DefaultWidthofReset}
2853   \def\Fld@height{\DefaultHeightofReset}
2854   \group_begin:
2855     \exp_args:No\HyField@SetKeys
2856     {
2857       \DefaultOptionsofReset,#1
2858     }
2859     \mode_leave_vertical:
2860     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2861     {
2862       \HyField@FlagsPushButton
2863       \legacy_if:nT { Fld@hidden }
2864       { \def\Fld@width{1sp} }
2865       \HyAnn@AbsPageLabel
2866       \Hy@escapeform\PDFForm@Reset
2867       \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2868       \pdfannot_box:nnnn
2869       {\box_wd:N\l_tmpa_box}
2870       {\box_ht:N\l_tmpa_box}
2871       {\box_dp:N\l_tmpa_box} %is this correct?
2872       { \PDFForm@Reset }
2873       \HyField@AddToFields
2874       \box_use:N \l_tmpa_box
2875     }
2876     {
2877       \msg_error:nn { hyp }{ pdfa-no-reset-button }
2878       \MakeButtonField{#2}
2879     }
2880   \group_end:
2881 }
2882
2883 \def\@CheckBox[#1]#2
2884 {% parameters, label
2885   \def\Fld@name{#2}
2886   \def\Fld@default{0}
2887   \group_begin:
2888     \def\Fld@width {\DefaultWidthofCheckBox}
2889     \def\Fld@height{\DefaultHeightofCheckBox}
2890     \exp_args:No\HyField@SetKeys
2891     {
2892       \DefaultOptionsofCheckBox,#1
2893     }
2894     \PDFForm@Name
2895     \HyField@FlagsCheckBox
2896     \legacy_if:nT { Fld@hidden }
2897     {
2898       \def\Fld@width{1sp}

```

```

2899     }
2900 \LayoutCheckField{#2}
2901 {
2902   \mode_leave_vertical:
2903   \HyAnn@AbsPageLabel
2904   \Hy@escapeform\PDFForm@Check
2905   \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
2906   {
2907     \pdfxform_new:nnn
2908     {__hyp_xform_CheckMarkYes}{-}
2909     {
2910       \group_begin:
2911       \fontfamily{pzd}
2912       \fontencoding{U}
2913       \fontseries{m}
2914       \fontshape{n}
2915       \selectfont
2916       \char51
2917       \group_end:
2918     }
2919     \pdfxform_new:nnn
2920     {__hyp_xform_CheckMarkOff}{-}
2921     {
2922       \group_begin:
2923       \fontfamily{pzd}
2924       \fontencoding{U}
2925       \fontseries{m}
2926       \fontshape{n}
2927       \selectfont
2928       \phantom{\char51} %perhaps xetex needs some small glyph ..
2929       \group_end:
2930     }
2931   }
2932   \pdfannot_box:nmmm
2933   {\Fld@width}
2934   {\Fld@height}
2935   {Opt} %is this correct?
2936   {\PDFForm@Check}
2937   \HyField@AddToFields %check if this works with xelatex ...
2938 }
2939 \group_end:
2940 }
2941 \ExplSyntaxOff
2942
2943 %hm. Should a luatex driver use type1 fonts in fields????
2944 \ExplSyntaxOn
2945 \def\Hy@FormObjects
2946 {
2947   \pdf_object_new:n   {__hyp/Encoding/pdfdoc }
2948   \pdf_object_new:n   {__hyp/Font/ZaDb }
2949   \pdf_object_new:n   {__hyp/Font/Helv }
2950   \pdf_object_write:nmx {__hyp/Encoding/pdfdoc } { dict }
2951   {
2952     /Type/Encoding

```

```

2953 /Differences[
2954   24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2955   /ring/tilde
2956   \c_space_tl
2957   39/quotesingle
2958   \c_space_tl
2959   96/grave %
2960   \iow_newline:
2961   128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2962   /fraction/guilsinglleft/guilsinglright/minus/perthousand
2963   /quotedblbase/quotedblleft/quotedblright/quoteleft
2964   /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2965   /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2966   \iow_newline:
2967   164/currency
2968   \c_space_tl
2969   166/brokenbar
2970   \c_space_tl
2971   168/dieresis/copyright/ordfeminine
2972   \c_space_tl
2973   172/logicalnot/.notdef/registered/macron/degree/plusminus
2974   /twosuperior/threesuperior/acute/mu
2975   \c_space_tl
2976   183/periodcentered/cedilla/onesuperior/ordmasculine
2977   \c_space_tl
2978   188/onequarter/onehalf/threequarters
2979   \iow_newline:
2980   192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
2981   /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
2982   /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
2983   /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
2984   /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
2985   /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
2986   /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
2987   /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
2988   /ocircumflex/otilde/odieresis/divide/oshlash/ugrave
2989   /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
2990 ]
2991 }
2992 \pdf_object_write:nnn {__hyp/Font/ZaDb } { dict }
2993 {
2994   /Type/Font
2995   /Subtype/Type1
2996   /Name/ZaDb
2997   /BaseFont/ZapfDingbats
2998 }
2999 \pdf_object_write:nxx {__hyp/Font/Helv } { dict }
3000 {
3001   /Type/Font
3002   /Subtype/Type1
3003   /Name/Helv
3004   /BaseFont/Helvetica
3005   /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
3006 }

```



```

3007 \global\let\Hy@FormObjects\relax
3008 }
3009 \ExplSyntaxOff
3010 \providecommand*\Fld@pageobjref{}
3011 \ifcsname pdf@escapestring\endcsname
3012 \def\Hy@escapeform#1{%
3013 \ifHy@pdfescapeform
3014 \let\Hy@escapestring\pdfescapestring
3015 \else
3016 \let\Hy@escapestring\@firstofone
3017 \fi
3018 }%
3019 \Hy@escapeform{}%
3020 \else
3021 \let\Hy@escapestring\@firstofone
3022 \def\Hy@escapeform#1{%
3023 \ifHy@pdfescapeform
3024 \def\Hy@escapestring##1{%
3025 \noexpand\Hy@escapestring{\noexpand##1}%
3026 }%
3027 \edef\Hy@temp{#1}%
3028 \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring}\@nil
3029 \def\Hy@escapestring##1{%
3030 \@ifundefined{Hy@esc@\string##1}{%
3031 ##1%
3032 \ThisShouldNotHappen
3033 }{%
3034 \csname Hy@esc@\string##1\endcsname
3035 }%
3036 }%
3037 \else
3038 \let\Hy@escapestring\@firstofone
3039 \fi
3040 }%
3041 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3042 \ifx\#3\%
3043 \else
3044 \expandafter
3045 \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3046 \ltx@ReturnAfterFi{%
3047 \Hy__hypescapeform#3\@nil
3048 }%
3049 \fi
3050 }%
3051 \fi
3052 \def\PDFForm@Name{%
3053 \PDFForm__hypName\Fld@name
3054 \ifx\Fld@altname\relax
3055 \else
3056 \PDFForm__hypName\Fld@altname
3057 \fi
3058 \ifx\Fld@mappingname\relax
3059 \else
3060 \PDFForm__hypName\Fld@mappingname

```

```

3061 \fi
3062 }
3063 \def\PDFForm__hypName#1{%
3064 \beginingroup
3065 \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3066 \ifHy@unicode
3067 \Hy@unicodedefalse
3068 \fi
3069 \fi
3070 \pdfstringdef\Hy@gtemp#1%
3071 \endgroup
3072 \let#1\Hy@gtemp
3073 }
3074 \def\Fld@X@additionalactions{%
3075 \ifx\Fld@keystroke@code\@empty
3076 \else
3077 /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3078 \fi
3079 \ifx\Fld@format@code\@empty
3080 \else
3081 /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3082 \fi
3083 \ifx\Fld@validate@code\@empty
3084 \else
3085 /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3086 \fi
3087 \ifx\Fld@calculate@code\@empty
3088 \else
3089 /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3090 \fi
3091 \ifx\Fld@onfocus@code\@empty
3092 \else
3093 /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3094 \fi
3095 \ifx\Fld@onblur@code\@empty
3096 \else
3097 /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3098 \fi
3099 \ifx\Fld@onmousedown@code\@empty
3100 \else
3101 /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3102 \fi
3103 \ifx\Fld@onmouseup@code\@empty
3104 \else
3105 /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3106 \fi
3107 \ifx\Fld@onenter@code\@empty
3108 \else
3109 /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3110 \fi
3111 \ifx\Fld@onexit@code\@empty
3112 \else
3113 /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3114 \fi

```

```

3115 }
3116 \ExplSyntaxOn
3117 \def\Fld@additionalactions
3118 {%
3119   \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{}
3120   {
3121     \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3122     {/AA<<\Fld@X@additionalactions>>}
3123   }
3124 }
3125 \ExplSyntaxOff
3126 \def\Fld@annotnames{%
3127   /T(\Fld@name)%
3128   \ifx\Fld@altname\relax
3129   \else
3130     /TU(\Fld@altname)%
3131   \fi
3132   \ifx\Fld@mappingname\relax
3133   \else
3134     /TM(\Fld@mappingname)%
3135   \fi
3136 }
3137 \ExplSyntaxOn
3138 \def\PDFForm@Check
3139 {
3140   /Subtype/Widget
3141   ~\Fld@annotflags
3142   ~\Fld@pageobjref
3143   ~\Fld@annotnames
3144   /FT/Btn
3145   \Fld@flags
3146   /Q~\Fld@align
3147   /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3148   /AP
3149   <<
3150     /N
3151     <<
3152       /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3153       /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3154     >>
3155   >>
3156   /MK<<
3157     \int_compare:nNnF {\Fld@rotation}={0}
3158     {
3159       /R~\Fld@rotation
3160     }
3161   \tl_if_empty:NF\Fld@bordercolor
3162   {
3163     /BC[\Fld@bordercolor]
3164   }
3165   \tl_if_empty:NF\Fld@bcolor
3166   {
3167     /BG[\Fld@bcolor]
3168   }

```

```

3169 /CA(\Hy@escapestring{\Fld@cbsymbol})%
3170 >>
3171 /DA
3172 (
3173 /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3174 \tl_if_empty:NF \Fld@color
3175 {
3176 \c_space_tl \Fld@color
3177 }
3178 )
3179 /H/P
3180 \legacy_if:nTF {Fld@checked}
3181 {
3182 /V/Yes /AS/Yes
3183 }
3184 {
3185 /V/Off /AS/Off
3186 }
3187 \Fld@additionalactions
3188 }
3189 \ExplSyntaxOff
3190 \ExplSyntaxOn
3191 \def\PDFForm@Push
3192 {
3193 /Subtype/Widget
3194 ~\Fld@annotflags
3195 ~\Fld@pageobjref
3196 ~\Fld@annotnames
3197 /FT/Btn
3198 ~\Fld@flags
3199 /H/P
3200 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3201 \bool_if:nT
3202 {
3203 !\int_compare_p:nNn {\Fld@rotation} = {0}
3204 ||
3205 \tl_if_exist_p:N \Fld@bordercolor
3206 }
3207 {
3208 /MK
3209 <<
3210 \int_compare:nNnF {\Fld@rotation} = {0}
3211 {
3212 /R~\Fld@rotation
3213 }
3214 \tl_if_exist:NT \Fld@bordercolor
3215 {
3216 /BC[\Fld@bordercolor]
3217 }
3218 >>
3219 }
3220 /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3221 \Fld@additionalactions
3222 }

```

```

3223
3224 \ExplSyntaxOff
3225 \def\PDFForm@List{%
3226   /Subtype/Widget%
3227   \Fld@annotflags
3228   \Fld@pageobjref
3229   \Fld@annotnames
3230   /FT/Ch%
3231   \Fld@flags
3232   /Q \Fld@align
3233   /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3234   \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3235     \ifx\Fld@bordercolor\relax\else 1\fi
3236     \ifx\Fld@bcolor\relax \else 1\fi
3237     \space
3238   \else
3239     /MK<<%
3240     \ifnum\Fld@rotation=\z@
3241     \else
3242       /R \Fld@rotation
3243       \fi
3244       \ifx\Fld@bordercolor\relax
3245       \else
3246         /BC[\Fld@bordercolor]%
3247         \fi
3248         \ifx\Fld@bcolor\relax
3249         \else
3250           /BG[\Fld@bcolor]%
3251           \fi
3252       >>%
3253   \fi
3254   /DA(</Helv \strip@pt\Fld@charsize\space Tf%
3255     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3256   \Fld@choices
3257   \Fld@additionalactions
3258 }
3259 \ExplSyntaxOn
3260 \def\PDFForm@Radio
3261 {
3262   /Subtype/Widget
3263   ~\Fld@annotflags
3264   ~\Fld@pageobjref
3265   ~\Fld@annotnames
3266   /FT/Btn
3267   \Fld@flags
3268   /H/P
3269   /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3270   /MK<<
3271     \ifnum\Fld@rotation=\z@
3272     \else
3273       /R~\Fld@rotation
3274       \fi
3275     \ifx\Fld@bordercolor\relax
3276     \else

```

```

3277     /BC[\Fld@bordercolor]%
3278     \fi
3279     \ifx\Fld@bcolor\relax
3280     \else
3281     /BG[\Fld@bcolor]%
3282     \fi
3283     /CA(\Hy@escapestring{\Fld@radiosymbol})%
3284     >>
3285     /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3286     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3287     \ifx\Fld@default\@empty
3288     /V/Off%
3289     /DV/Off%
3290     \else
3291     /V/\Fld@default
3292     /DV/\Fld@default
3293     \fi
3294     \Fld@additionalactions
3295   }
3296   \ExplSyntaxOff
3297   \ExplSyntaxOn
3298   % Does an appearance dict make sense here?
3299   \def\PDFForm@Text
3300   {
3301     /Subtype/Widget
3302     ~\Fld@annotflags
3303     ~\Fld@pageobjref
3304     ~\Fld@annotnames
3305     /FT/Tx
3306     ~\Fld@flags
3307     /Q~\Fld@align
3308     /BS<</W~\Fld@borderwidth\c_space_t1 /S /\Fld@borderstyle>>
3309     \bool_if:nT
3310     {
3311       !\int_compare_p:nNn {\Fld@rotation} = {0}
3312       ||
3313       \tl_if_exist_p:N \Fld@bordercolor
3314       ||
3315       \tl_if_exist_p:N \Fld@bcolor
3316     }
3317     {
3318       /MK
3319       <<
3320         \int_compare:nNf {\Fld@rotation} = {0}
3321         {
3322           /R~\Fld@rotation
3323         }
3324         \tl_if_exist:NT \Fld@bordercolor
3325         {
3326           /BC[\Fld@bordercolor]
3327         }
3328         \tl_if_exist:NT \Fld@bcolor
3329         {
3330           /BG[\Fld@bcolor]

```

```

3331     }
3332     >>
3333   }
3334 /DA
3335   (
3336     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3337     \tl_if_empty:NF {\c_space_tl\Fld@color}
3338   )
3339 /DV(\Hy@escapestring{\Fld@default})
3340 /V(\Hy@escapestring{\Fld@value})
3341 ~\Fld@additionalactions
3342 \int_compare:nNnT { \Fld@maxlen}>{0}
3343   {
3344     /MaxLen~\Fld@maxlen
3345   }
3346 }
3347 \ExplSyntaxOff
3348
3349 \def\PDFForm@Submit{%
3350 /Subtype/Widget%
3351 \Fld@annotflags
3352 \Fld@pageobjref
3353 \Fld@annotnames
3354 /FT/Btn%
3355 \Fld@flags
3356 /H/P%
3357 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3358 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3359 \ifx\Fld@bordercolor\relax\else 1\fi
3360 \space
3361 \else
3362 /MK<<%
3363 \ifnum\Fld@rotation=\z@
3364 \else
3365 /R \Fld@rotation
3366 \fi
3367 \ifx\Fld@bordercolor\relax
3368 \else
3369 /BC[\Fld@bordercolor]%
3370 \fi
3371 >>%
3372 \fi
3373 /A<<%
3374 /S/SubmitForm%
3375 /F<<%
3376 /FS/URL%
3377 /F(\Hy@escapestring{\Form@action})%
3378 >>%
3379 \Fld@submitflags
3380 >>%
3381 \Fld@additionalactions
3382 }
3383 \ExplSyntaxOn
3384 \def\PDFForm@Reset{%

```

```

3385     /Subtype/Widget%
3386     \Fld@annotflags
3387     \Fld@pageobjref
3388     \Fld@annotnames
3389     /FT/Btn%
3390     \Fld@flags
3391     /H/P%
3392     /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3393     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3394         \ifx\Fld@bordercolor\relax\else 1\fi
3395         \space
3396     \else
3397         /MK<<%
3398             \ifnum\Fld@rotation=\z@
3399             \else
3400                 /R~\Fld@rotation
3401             \fi
3402             \ifx\Fld@bordercolor\relax
3403             \else
3404                 /BC[\Fld@bordercolor]%
3405             \fi
3406         >>%
3407     \fi
3408     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3409     /A<</S/ResetForm>>%
3410     \Fld@additionalactions
3411 }%
3412
3413
3414 %these patterns are used in hyperref checks.
3415 %it is unclear if they are really useful and if a backend support is
3416 %needed.
3417 \str_case:VnF \c_sys_backend_str
3418 {
3419     { pdfmode }
3420     {
3421         \def\HyPat@ObjRef
3422         {
3423             [0-9]*[1-9][0-9]*~0~R
3424         }
3425     }
3426     { dvipdfmx }
3427     {
3428         \def\HyPat@ObjRef
3429         {
3430             @[\^~]+
3431         }
3432     }
3433     { xdvipdfmx }
3434     {
3435         \def\HyPat@ObjRef
3436         {
3437             @[\^~]+
3438         }

```



```

3439 }
3440 }
3441 { %also set in hyperref sty, so probably not needed.
3442   \def\HyPat@ObjRef/{.+}
3443 }
3444
3445
3446 \ExplSyntaxOff
3447 % UF: removed Hy@writebookmark
3448 %   \Hy@currentbookmarklevel{0}
3449 %   \Hy@numberline
3450 %   \__hypwritetorep
3451 %   counter{bookmark@seq@number}
3452 % removed \HyPsd@SanitizeForOutFile, not needed
3453 % removed \currentpdfbookmark, defined by bookmark,
3454 % should use \newcommand there
3455 % removed \subpdfbookmark, defined by bookmark,
3456 % should use \newcommand there
3457 % removed \belowpdfbookmark, defined by bookmark,
3458 % should use \newcommand there
3459 % removed \pdfbookmark, defined by bookmark,
3460 % \BOOKMARK
3461 % \@BOOKMARK
3462 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3463 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3464 %% removed \ReadBookmarks / unneeded with bookmark.
3465 %% removed \Hy@OutlineName
3466 %% removed \check@bm@number
3467 %% removed \calc@bm@number
3468
3469 \ifHy@implicit
3470 \else
3471   \expandafter\endinput
3472 \fi
3473 \newlength\Hy@SectionHShift
3474 \def\Hy@SectionAnchorHref#1{%
3475   \ifx\protect\@typeset@protect
3476     \Hy__hypSectionAnchor{#1}%
3477   \fi
3478 }
3479 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%
3480   \leavevmode
3481   \hbox to 0pt{%
3482     \kern-\Hy@SectionHShift
3483     \Hy@raisedlink{%
3484       \hyper@anchorstart{#1}\hyper@anchorend
3485     }%
3486     \hss
3487   }%
3488 }
3489 \@ifundefined{hyper@nopatch@sectioning}
3490 {
3491   \let\H@old@ssect\@ssect
3492   \def\@ssect#1#2#3#4#5{%

```

```

3493 \Hy@MakeCurrentHrefAuto{section*}%
3494 \setlength{\Hy@SectionHShift}{#1}%
3495 \begingroup
3496   \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3497   \toks\tw@\expandafter{%
3498     \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3499     #5%
3500   }%
3501 \edef\x{\endgroup
3502   \the\toks@\the\toks\tw@}%
3503 }\x
3504 }
3505 \let\H@old@schapter\@schapter
3506 \def\@schapter#1{%
3507   \begingroup
3508     \let\@mkboth\@gobbletwo
3509     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3510     \Hy@raisedlink{%
3511       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3512     }%
3513   \endgroup
3514   \H@old@schapter{#1}%
3515 }
3516 \ltx@ifundefined{chapter}{-}{%
3517   \let\Hy@org@chapter\@chapter
3518   \def\@chapter{%
3519     \def\Hy@next{%
3520       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3521       \Hy@raisedlink{%
3522         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3523       }%
3524     }%
3525     \ifnum\c@secnumdepth>\m@ne
3526       \ltx@ifundefined{if@mainmatter}%
3527       \iftrue{\csname if@mainmatter\endcsname}%
3528       \let\Hy@next\relax
3529     \fi
3530     \fi
3531     \Hy@next
3532     \Hy@org@chapter
3533   }%
3534 }
3535 \let\H@old@part\@part
3536 \begingroup\expandafter\expandafter\expandafter\endgroup
3537 \expandafter\ifx\csname chapter\endcsname\relax
3538   \let\Hy@secnum@part\z@
3539 \else
3540   \let\Hy@secnum@part\m@ne
3541 \fi
3542 \def\@part{%
3543   \ifnum\Hy@secnum@part>\c@secnumdepth
3544     \phantomsection
3545   \fi
3546   \H@old@part

```

```

3547 }
3548 \let\H@old@spart\@spart
3549 \def\@spart#1{%
3550   \Hy@MakeCurrentHrefAuto{part*}%
3551   \Hy@raisedlink{%
3552     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3553   }%
3554   \H@old@spart{#1}%
3555 }
3556 \let\H@old@sect\@sect
3557 \def\@sect#1#2#3#4#5#6[#7]#8{%
3558   \ifnum #2>\c@secnumdepth
3559     \expandafter\@firstoftwo
3560   \else
3561     \expandafter\@secondoftwo
3562   \fi
3563   {%
3564     \Hy@MakeCurrentHrefAuto{section*}%
3565     \setlength{\Hy@SectionHShift}{#3}%
3566     \begingroup
3567     \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[#7]}%
3568     \toks\tw@\expandafter{%
3569       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3570       #8%
3571     }%
3572     \edef\x{\endgroup
3573       \the\toks@\the\toks\tw@}%
3574     }\x
3575   }{%
3576     \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[#7]{#8}%
3577   }%
3578 }
3579 }{}
3580 \expandafter\def\csname Parent-4\endcsname{}
3581 \expandafter\def\csname Parent-3\endcsname{}
3582 \expandafter\def\csname Parent-2\endcsname{}
3583 \expandafter\def\csname Parent-1\endcsname{}
3584 \expandafter\def\csname Parent0\endcsname{}
3585 \expandafter\def\csname Parent1\endcsname{}
3586 \expandafter\def\csname Parent2\endcsname{}
3587 \expandafter\def\csname Parent3\endcsname{}
3588 \expandafter\def\csname Parent4\endcsname{}
3589 %%
3590 %% End of file 'hgeneric-testphase.def'.
3591 \</package>
3592 <*colorscheme>
3593 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3594 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3595 % linkcolor=
3596 %,filecolor=
3597 %,urlcolor=
3598 %,menucolor=
3599 %,runcolor=
3600 %,linkbordercolor=

```

```

3601 %,filebordercolor=
3602 %,urlbordercolor=
3603 %,menubordercolor=
3604 %,runbordercolor=
3605
3606 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }
3607 {
3608     linkcolor      = [rgb]{1,0,0}, %red
3609     filecolor      = [rgb]{0,1,1}, %cyan
3610     urlcolor       = [rgb]{1,0,1}, %magenta
3611     menucolor      = [rgb]{1, 0, 0}, %red
3612     runcolor       = [rgb]{0,1,1}, %cyan
3613 %-----
3614     linkbordercolor = [rgb]{1, 0 ,0 },
3615     filebordercolor = [rgb]{0, .5, .5},
3616     urlbordercolor  = [rgb]{0, 1, 1},
3617     menubordercolor = [rgb]{1, 0, 0},
3618     runbordercolor  = [rgb]{0, .7, .7}
3619 }
3620
3621 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3622 {
3623     linkcolor      = [rgb]{0,0.2,0.6},
3624     filecolor      = [rgb]{0.8,0,0.8},
3625     urlcolor       = [rgb]{0.8,0,0.8},
3626     menucolor      = [rgb]{0,0.2,0.6},
3627     runcolor       = [rgb]{0.8,0,0.8},
3628 %----- %-----
3629     linkbordercolor = [rgb]{0,0.2,0.6},
3630     filebordercolor = [rgb]{0.8,0,0.8},
3631     urlbordercolor  = [rgb]{0.8,0,0.8},
3632     menubordercolor = [rgb]{0,0.2,0.6},
3633     runbordercolor  = [rgb]{0.8,0,0.8}
3634 }
3635
3636 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3637 { %two colors: intern/extern
3638     linkcolor      = [rgb]{0.79216, 0, 0.12549},
3639     filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3640     urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3641     menucolor      = [rgb]{0.79216, 0, 0.12549 },
3642     runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3643 %----- %-----
3644     linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3645     filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3646     urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3647     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3648     runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3649 }
3650
3651 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3652 { %all darkgray
3653     linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3654     filecolor      = [rgb]{0.4 ,0.4 ,0.4 },

```

```

3655     urlcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3656     menucolor     = [rgb]{0.4 ,0.4 ,0.4 },
3657     runcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3658     %-----      %-----
3659     linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3660     filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3661     urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3662     menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3663     runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3664 }
3665
3666 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3667 { %dvipsnam.def
3668     linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3669     filecolor      = [rgb]{1, 0, 0}, %Red
3670     urlcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3671     menucolor     = [rgb]{1, 0, 0}, %Red
3672     runcolor      = [rgb]{1, 0, 0}, %Red
3673     %-----      %-----
3674     linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3675     filebordercolor = [rgb]{1, 0, 0}, %Red
3676     urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3677     menubordercolor = [rgb]{1, 0, 0}, %Red
3678     runbordercolor  = [rgb]{1, 0, 0} %Red
3679 }
3680
3681 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3682 { %dvipsnam.def
3683     linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3684     filecolor      = [rgb]{0, 1, 0}, %Green
3685     urlcolor      = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3686     menucolor     = [rgb]{0.06, 0.46, 1}, %NavyBlue
3687     runcolor      = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3688     %-----      %-----
3689     linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3690     filebordercolor = [rgb]{0, 1, 0}, %Green
3691     urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3692     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3693     runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3694 }
3695
3696
3697 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3698 {
3699     linkcolor      = [rgb]{0.50196, 0, 0.02353},
3700     filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3701     urlcolor      = [rgb]{0.54118, 0, 0.52941},
3702     menucolor     = [rgb]{0.44706, 0.45882, 0},
3703     runcolor      = [rgb]{0.07451, 0.46667, 0.46275},
3704     %-----      %-----
3705     linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3706     filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3707     urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3708     menubordercolor = [rgb]{0.668236, 0.675292, 0.4},

```

```

3709     runbordercolor = [rgb]{0.444706, 0.680002, 0.67765}
3710   }
3711
3712   \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3713   {
3714     linkcolor      = [rgb]{0,0,0},
3715     filecolor      = [rgb]{0,0,0},
3716     urlcolor       = [rgb]{0,0,0},
3717     menucolor      = [rgb]{0,0,0},
3718     runcolor       = [rgb]{0,0,0},
3719     %----- %-----
3720     linkbordercolor = [rgb]{0,0,0},
3721     filebordercolor = [rgb]{0,0,0},
3722     urlbordercolor  = [rgb]{0,0,0},
3723     menubordercolor = [rgb]{0,0,0},
3724     runbordercolor  = [rgb]{0,0,0}
3725   }
3726 </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	
<code>\#</code>	278, 732
<code>\\$</code>	277
<code>\%</code>	733
<code>\-</code>	2112
<code>\.</code>	542, 546, 548
@curropt commands:	
<code>\@curropt:</code>	2633, 2700
<code>\[</code>	2112
<code>\]</code> ...	19, 20, 30, 31, 32, 40, 44, 54, 74, 81, 88, 95, 102, 117, 122, 123, 131, 132, 141, 142, 143, 144, 145, 152, 159, 279, 280, 862, 2635, 2702, 3042
<code>_</code>	540, 542, 546, 548
<code>\]</code>	2112
A	
<code>\A</code>	540, 2112
<code>\Acrobatmenu</code>	18, 172
<code>\addcontentsline</code>	13
<code>\AddToDocumentProperties</code> ..	387, 2107, 2208
<code>\AddToHook</code>	425, 438
<code>\AddToHookNext</code>	197
<code>\advance</code>	2566, 2640, 2684, 2685
<code>allcolors</code> (hypersetup key) ..	<u>975</u>
<code>\author</code>	2
B	
<code>\b</code>	544
<code>\begingroup</code>	208, 276, 374, 2472, 2600, 2629, 2641, 2675, 3064, 3495, 3507, 3536, 3566
<code>\belowpdfbookmark</code>	3457
<code>\bgroup</code>	273, 374
<code>\BOOKMARK</code>	3460
<code>bookmarkstype</code> (hypersetup key) ..	13
bool commands:	
<code>\bool_if:N</code> TF	266, 295, 354, 679, 695, 707, 720, 728, 758, 816, 873, 945, 955, 1199, 1210
<code>\bool_if:n</code> TF	828, 1882, 1907, 3201, 3309
<code>\bool_lazy_and:nn</code> TF	422
<code>\bool_lazy_or:nn</code> TF	1229, 1278
<code>\bool_new:N</code> ...	202, 203, 526, 530, 534
<code>\bool_set_true:N</code>	535
bordercolormodel (hypersetup key) ..	
<code>\box_dp:N</code>	616, 2785, 2836, 2871
<code>\box_ht:N</code>	615, 2784, 2835, 2870
<code>\box_new:N</code>	475, 537
<code>\box_set_dp:Nn</code>	2079
<code>\box_set_ht:Nn</code>	2086
<code>\box_use:N</code> ...	1216, 2787, 2845, 2874
<code>\box_use_drop:N</code>	1221
<code>\box_wd:N</code>	614, 2783, 2834, 2869
<code>\l_tmpa_box</code>	2781, 2783, 2784, 2785, 2787, 2820, 2834, 2835, 2836, 2845, 2867, 2869, 2870, 2871, 2874
C	
<code>\catcode</code>	277, 278
<code>\char</code>	2719, 2916, 2928
<code>\chardef</code>	157
<code>\cite</code>	33
clist commands:	
<code>\clist_item:nn</code>	2139
<code>\clist_map_function:nN</code>	124, 133
<code>\clist_map_inline:nn</code>	2272
<code>\clist_map_variable:nNn</code> ..	2632, 2699
color commands:	
<code>\color_export:nnN</code> ..	38, 430, 904, 1017
<code>\color_select:n</code>	912, 948, 1220, 2826, 2830
<code>\color_select:nn</code>	918
<code>\color_set:nn</code>	2, 392, 929
<code>\color_set:nnn</code>	2, 391, 935
color names:	
<code>hyp/annot/file</code>	<u>523</u>
<code>hyp/annot/link</code>	<u>523</u>
<code>hyp/annot/menu</code>	<u>523</u>
<code>hyp/annot/run</code>	<u>523</u>
<code>hyp/annot/url</code>	<u>523</u>
<code>colorfile</code> (hypersetup key)	<u>975</u>
<code>colorlink</code> (hypersetup key)	<u>975</u>
<code>colorlinks</code> (hypersetup key)	<u>961</u>
<code>colormenu</code> (hypersetup key)	<u>975</u>
<code>colorrun</code> (hypersetup key)	<u>975</u>
<code>colorscheme</code> (hypersetup key)	<u>1</u> , <u>1387</u>
<code>colorurl</code> (hypersetup key)	<u>975</u>
cs commands:	
<code>\cs_generate_variant:Nn</code> ..	155, 156, 389, 454, 469, 474, 578, 904, 921, 938
<code>\cs_gset:Npn</code>	1191
<code>\cs_gset_eq:NN</code>	408, 419
<code>\cs_if_exist:N</code> TF	12
<code>\cs_if_exist_p:N</code>	423

\l__hyp_annot_Launch_bool	816	\l__hyp_href_pdf_page_tl	206, 224, 786
\l__hyp_annot_Named_bool	873	__hyp_href_run_aux:nn	335, 338
\c__hyp_annot_types_seq		\l__hyp_href_run_parameter_tl	
.....	487, 524, 528, 975, 1276, 1374	207, 228, 340
\l__hyp_annot_URI_bool	720	__hyp_href_url_aux:n	361, 364
\g__hyp_bordercolormodel_str		__hyp_href_url_aux:nn	302, 304
.....	432, 523, 1000, 1019	\l__hyp_href_url_encode_bool	
__hyp_citebordercolor_hook_-		202, 213, 251, 266, 295, 354
init:	397, 410, 419	__hyp_href_url_format:	
__hyp_citecolor_hook_init:		208, 214, 256, 367
.....	395, 399, 408	\l__hyp_href_url_ismap_bool	
__hyp_clist_display:n	117, 124, 133	203, 227, 728
__hyp_color_select:n	39, 905, 905, 921	\l__hyp_href_url_protocol_tl	
__hyp_color_select_aux:wn		204, 215, 254, 306, 368
.....	905, 909, 916	__hyp_link_goto_begin:nw	
__hyp_color_set:nn	38, 39,	650, 685, 701
395, 397, 429, 922, 922, 938, 980, 1016		__hyp_link_goto_end:	672, 686, 709
__hyp_color_set_aux:nwn	922, 926, 933	\c__hyp_map_annot_hyp_prop	487
\c__hyp_colorscheme_daleif_prop	3621	\c__hyp_map_hyp_annot_prop	
\c__hyp_colorscheme_henryford_-		487, 939,
prop	3712	1003, 1043, 1070, 1078, 1089, 1116,	
\c__hyp_colorscheme_julian_prop	3636	1124, 1193, 1301, 1337, 1347, 1460	
\c__hyp_colorscheme_phelype_prop		__hyp_ocg_init:	1135, 1135, 1191, 1201
.....	3697	\l__hyp_optlang_regex	2111, 2112, 2141
\c__hyp_colorscheme_szabolcsA_-		l__hyp_page/Trans	551
prop	3666	__hyp_PageLabels_gpush:	585, 585, 598
\c__hyp_colorscheme_szabolcsB_-		\l__hyp_para_tmpa_tl	483, 827, 830, 839
prop	3681	\l__hyp_para_tmpa_tl _{uuu} \l__hyp_-	
\c__hyp_colorscheme_tivv_prop	3651	text_tmpa_str _{uuu} \g__hyp_text_-	
\l__hyp_dest_box	31,	tmpa_str	480
537, 614, 615, 616, 2063, 2079, 2086		__hyp_ref_check:nn	447, 457, 469, 2571
\l__hyp_dest_name_tmpa_tl	480,	__hyp_ref_if_exist:nn	447, 455, 456
653, 654, 660, 664, 666, 669, 779, 792		__hyp_ref_if_exist:nnTF	459, 2575
\l__hyp_dest_pdfremotestartview_-		__hyp_ref_label:nn	447, 448, 454, 2570
tl	506, 787, 1870, 1874	__hyp_ref_value:nn	447, 470, 474, 2579
\g__hyp_dest_pdfstartpage_tl		__hyp_secondoftwewithopt:wnn	
...	506, 1881, 1883, 1890, 1908, 1915	380, 381, 382, 383
\g__hyp_dest_pdfstartview_tl		__hyp_setup_info_date_key:nn	
...	506, 1883, 1890, 1901, 1905, 1908, 1915	2194, 2227, 2228
\l__hyp_dest_pdfview_tl		__hyp_setup_info_key:nn	
.....	522, 632, 641,	2187, 2188, 2189, 2190, 2192, 2193	
2032, 2035, 2042, 2045, 2046, 2047,		__hyp_store_metadata:nn	
2048, 2049, 2050, 2055, 2059, 2095		384, 389, 1479, 2155,
\c__hyp_dest_startview_regex		2173, 2207, 2222, 2231, 2247, 2315	
.....	538, 1868, 1899	__hyp_text_cleanup:N	558, 558, 572
\c__hyp_dest_undefined_tl		\l__hyp_text_enc_dest_print_tl	
.....	486, 659, 660	509, 778
__hyp_destination:nn		\l__hyp_text_enc_dest_tl	
.....	31, 601, 601, 632, 641	31, 509, 608, 665
\l__hyp_filename_tmpa_tl	480,	\l__hyp_text_enc_file_print_tl	
761, 763, 769, 770, 775, 822, 823, 839		509, 821
__hyp_href_pdf_aux:nn	318, 321	\l__hyp_text_enc_info_print_tl	
\l__hyp_href_pdf_destination_tl		509, 583
.....	205, 216, 255, 323		

\l__hyp_text_enc_para_print_tl	509, 826	draft	1411
\l__hyp_text_enc_uri_print_tl	268,	extension	13, 1429
271, 297, 300, 356, 359, 509, 725, 1471		file	10, 1460
__hyp_text_pdfstring:nnN	566, 566, 578, 580,	fileborderstyle	14, 1089
583, 606, 663, 723, 776, 819, 824, 1471		filecolor	975
__hyp_text_pdfstring_info:nn	581, 581, 2144, 2148, 2167, 2177	final	1411
__hyp_text_purify:nn	554, 554, 571	hidefile	1363
__hyp_text_string_from_unicode:nn	562, 562, 573	hidelink	1363
\g__hyp_text_tmpa_str	485, 574, 576	hidelinks	1363
\l__hyp_text_tmpa_str	484, 571, 572, 573, 574	hidemenu	1363
\l__hyp_tmpa_box 475, 1203, 1216, 1221		hiderun	1363
\l__hyp_tmpa_int	475	hideurl	1363
\l__hyp_tmpa_seq	475, 1868, 1870, 1899, 1901,	hypertextnames	1429
2022, 2023, 2027, 2029, 2053, 2060,		link	10, 1460
2061, 2062, 2071, 2073, 2083, 2088,		linkborder	13
2141, 2142, 2148, 2521, 2523, 2529		linkborderstyle	14, 1089
\l__hyp_tmpa_str	475, 2144, 2148, 2150, 2152,	linkcolor	975
2167, 2168, 2170, 2177, 2178, 2182		linkfileprefix	1429
\l__hyp_tmpa_tl	475, 609, 613, 621, 1020, 1024,	linktoc	1429
1471, 1472, 1477, 1866, 1868, 1897,		linktocpage	1429
1899, 2029, 2030, 2037, 2139, 2141		localanchorname	1429
\l__hyp_uri_tmpa_tl	480, 726, 727	menu	10, 1460
hyp/anchor	627	menuborder	13
hyp/annot/file (color name)	523	menuborderstyle	14, 1089
hyp/annot/link (color name)	523	menucolor	975
hyp/annot/menu (color name)	523	naturalnames	1429
hyp/annot/run (color name)	523	nesting	13
hyp/annot/url (color name)	523	ocgcolorfile	1229
hyp/text/pdfstring	553	ocgcolorlink	1229
\hypercalcbp	11, 18, 160	ocgcolorlinks	1229
\HyperDestNameFilter	13, 607, 664	ocgcolormenu	1229
\hypersetup 2, 3, 5, 9, 10, 13, 18, 66, 99, 179		ocgcolorrun	1229
\hypersetup keys:		ocgcolorurl	1229
allcolors	975	pageanchor	1429
bookmarkstype	13	pdfauthor	2111
bordercolormodel	13, 996	pdfborder	13
colorfile	975	pdfborderstyle	14, 1089
colorlink	975	pdfcreationdate	14, 2194
colorlinks	961	pdfcreator	2111
colormenu	975	pdfencoding	1401
colorrun	975	pdfinfo	2261
colorscheme	1, 1387	pdfkeywords	2111
colorurl	975	pdflang	14, 2100
debug	1411	pdflinkmargin	14
destlabel	13	pdfmetadate	14, 2194
		pdfmoddate	14, 2194
		pdfproducer	2111
		pdfremotestartview	11
		pdfstartview	11
		pdfsubject	2111
		pdftitle	2111
		pdftrapped	2233
		pdfversion	1401
		pdfview	11, 2018

\msg_line_context:	82	\pdf_object_ref_last:	796, 2347
\g_msg_module_name_prop	11	\pdf_object_unnamed_write:nn	795, 2343
\msg_new:nnn 50, 57, 62, 66, 70, 77, 84, 91, 98, 105, 111, 118, 127, 137, 148		\pdf_object_write:nnn	1141, 1147, 1157, 1169, 2950, 2992, 2999
\msg_new:nnnn	14, 25, 36	\pdf_pageobject_ref:n 1890, 1915, 2577	
\msg_warning:nn	169, 1407	\pdf_string_from_unicode:nnN	564
\msg_warning:nnn	178, 656, 896	\pdf_version:	1553, 1672, 1701, 1735, 1750, 1776, 1801, 1834, 1850, 1955, 1982
\msg_warning:nnnn	1251, 1292, 1505, 1549, 1583, 1606, 1668, 1697, 1731, 1746, 1759, 1772, 1797, 1830, 1846, 1873, 1904, 1937, 1951, 1978, 2012, 2094	\pdf_version_compare:NnTF	1543, 1634, 1656, 1692, 1725, 1740, 1766, 1791, 1817, 1840, 1945, 1972
\msg_warning:nnnnn	189, 1323, 1357, 1449, 1525, 1562, 1649, 1682, 1707, 1783, 1810, 1859, 1964, 1991, 2255, 2359, 2378, 2397, 2408	\pdf_version_compare_p:Nn	832, 1230, 1279
N		\pdf_version_major:	166, 1231, 1254, 1280, 1296
naturalnames (hypersetup key)	1429	\pdf_version_minor:	165, 1254, 1296
nesting (hypersetup key)	13	pdfannot commands:	
\newcommand	163, 3454, 3456, 3458	\pdfannot_box:nnnn	2612, 2723, 2782, 2833, 2868, 2932
\newcount	2562, 2746	\pdfannot_box_ref_last:	2468, 2484
\NewDocumentCommand	2421	\pdfannot_dict_put:nnn	736, 796, 842, 881, 1021, 1056, 1080, 1102, 1126, 1308, 1339
\NewExpandableDocumentCommand	380	\pdfannot_dict_remove:nn	1011, 1051, 1072, 1097, 1118, 1316, 1349
\newlength	3473	\pdfannot_link:nnn	737, 797, 843, 882
\noexpand	2644, 2645, 2646, 2647, 3025	\pdfannot_link_goto_begin:nw	669
\nolinkurl	4	\pdfannot_link_goto_end:	674
O		\pdfannot_link_margin:n	8, 1587
ocgcolorfile (hypersetup key)	1229	\c_pdfannot_link_types_seq	532
ocgcolorlink (hypersetup key)	1229	pdfauthor (hypersetup key)	2111
ocgcolorlinks (hypersetup key)	1229	\pdfbookmark	3459
ocgcolormenu (hypersetup key)	1229	pdfborder (hypersetup key)	13
ocgcolorrun (hypersetup key)	1229	pdfborderstyle (hypersetup key)	14 , 1089
ocgcolorurl (hypersetup key)	1229	pdfcreationdate (hypersetup key)	14 , 2194
P		pdfcreator (hypersetup key)	2111
pageanchor (hypersetup key)	1429	\pdfdest	11
\paperwidth	3	pdfdict commands:	
\PassOptionsToPackage	445, 1421, 1426	\pdfdict_new:n 551, 714, 751, 810, 867	
pdf commands:		\pdfdict_put:nnn	236, 243, 552, 715, 716, 727, 730, 752, 753, 765, 772, 782, 792, 811, 812, 823, 838, 868, 869, 879, 1613, 1614, 1618, 1619, 2356, 2366, 2370, 2373, 2375, 2394, 2403, 2405
\pdf_bdcobject:nn	1215, 1218	\pdfdict_remove:nn 233, 835, 1623, 1624	
\pdf_destination:nn	27, 155, 620	\pdfdict_use:n	736, 795, 842, 848, 881, 886, 2345
\pdf_destination:nnnn	613	pdfencoding (hypersetup key)	1401
\pdf_emc:	1217, 1223	\pdfescapestring	3014
\pdf_link_user:nnn	2753	pdffile commands:	
\pdf_name_from_unicode_e:n	244, 765, 880	\pdffile_embed_file:nnn	767
\pdf_object_if_exist:nTF	763		
\pdf_object_new:n	1137, 1138, 1139, 1140, 2947, 2948, 2949		
\pdf_object_ref:n	156, 775, 1143, 1145, 1167, 1168, 1171, 1175, 1180, 1185, 1190, 2539, 2541, 3005		

<code>\seq_count:N</code>	2027, 2053	<code>\@CheckBox</code>	2883		
<code>\seq_get_right:NN</code>	2029	<code>\@ChoiceMenu</code>	2623		
<code>\seq_if_empty:NTF</code>	2142	<code>\@Form</code>	2506		
<code>\seq_item:Nn</code>	1870, 1901, 2023, 2071, 2073, 2083, 2088, 2148	<code>\@PushButton</code>	2761		
<code>\seq_map_inline:Nn</code>	524, 528, 532, 975, 1276, 1374, 2529	<code>\@Reset</code>	2850		
<code>\seq_new:N</code>	477	<code>\@Submit</code>	2802		
<code>\seq_put_right:Nn</code>	2060, 2061, 2062, 2521	<code>\@TextField</code>	2588		
<code>\seq_set_split:Nnn</code>	2022	<code>\@bookmarksopenlevel</code>	1488		
<code>\seq_sort:Nn</code>	2523	<code>\@bsphack</code>	450		
<code>\setbox</code>	374	<code>\@chapter</code>	3517, 3518		
<code>\setlength</code>	3494, 3565	<code>\@currDisplay</code>	2637, 2705		
<code>\setpdflinkmargin</code>	8	<code>\@currValue</code>	2703, 2733		
<code>\settowidth</code>	2637	<code>\@currentHref</code>	3498, 3511, 3522, 3552, 3569		
<code>\show</code>	2697	<code>\@curropt</code>	2622, 2632, 2635, 2699, 2702		
sort commands:		<code>\@empty</code>	2591, 2607, 3075, 3079, 3083, 3087, 3091, 3095, 3099, 3103, 3107, 3111, 3255, 3286, 3287		
<code>\sort_return_same:</code>	2527	<code>\@endForm</code>	2557		
<code>\sort_return_swapped:</code>	2526	<code>\@esphack</code>	452		
<code>\space</code>	465, 2705, 3237, 3254, 3255, 3285, 3286, 3360, 3392, 3395	<code>\@firstofone</code> ..	2461, 3016, 3021, 3038		
<code>\spacefactor</code>	603, 625	<code>\@firstoftwo</code>	3559		
<code>\stockwidth</code>	3	<code>\@for</code>	2633, 2700		
str commands:		<code>\@gobbletwo</code>	3508		
<code>\c_colon_str</code>	862	<code>\@hyper@launch</code>	37, 862		
<code>\c_hash_str</code>	732	<code>\@ifnextchar</code>	273		
<code>\c_percent_str</code>	733	<code>\@ifundefined</code>	3030, 3489		
<code>\str_case:nnTF</code>	2121, 3417	<code>\@latex@warning</code>	462		
<code>\str_case_e:nnTF</code>	2023	<code>\@mainaux</code>	2454, 2457, 2477, 2496		
<code>\str_compare:nNnTF</code>	2525	<code>\@mkboth</code>	3508		
<code>\str_gset:Nn</code>	1000	<code>\@nil</code>	3028, 3041, 3047		
<code>\str_gset_eq:NN</code>	574	<code>\@part</code>	3535, 3542		
<code>\str_head:n</code>	2244, 2249	<code>\@pdfauthor</code>	23		
<code>\str_if_eq:nnTF</code>	610, 2150, 2168, 2178, 3119	<code>\@pdfborder</code>	161		
<code>\str_if_eq_p:nn</code>	830, 1231, 1280	<code>\@pdfborderstyle</code>	162		
<code>\str_lowercase:n</code>	2023, 2173, 2222, 2245, 2250	<code>\@savsf</code>	603, 625		
<code>\str_new:N</code>	479, 484, 485, 523	<code>\@schapter</code>	3505, 3506		
<code>\str_set:Nn</code>	556	<code>\@secondoftwo</code>	3561		
<code>\str_set_convert:Nnnn</code>	8	<code>\@sect</code>	3556, 3557		
<code>\str_set_eq:NN</code>	576	<code>\@spart</code>	3548, 3549		
<code>\str_tail:n</code>	2245, 2250	<code>\@ssect</code>	3491, 3492		
<code>\str_uppercase:n</code>	2238, 2244, 2249	<code>\@tempdima</code> ..	2631, 2638, 2640, 2677, 2678, 2679, 2683, 2684, 2685, 2686		
<code>\string</code>	2455, 2458, 2479, 2497, 3030, 3034, 3045	<code>\@tempdimb</code>	2637, 2638		
<code>\subpdfbookmark</code>	3455	<code>\@typeset@protect</code>	3475		
sys commands:		<code>\BKM@color</code>	433		
<code>\c_sys_backend_str</code>	3417	<code>\c@secnumdepth</code>	3525, 3543, 3558		
T				<code>\calc@bm@number</code>	3467
TeX and L ^A T _E X 2 _ε commands:		<code>\check@bm@number</code>	3466	<code>\define@key</code>	427
<code>\@BOOKMARK</code>	3461	<code>\Fld@additionalactions</code> ..	3117, 3187, 3221, 3257, 3294, 3341, 3381, 3410	<code>\Fld@align</code>	3146, 3232, 3307
		<code>\Fld@altname</code> ..	3054, 3056, 3128, 3130		

\Fld@annotflags	3141,	\Fld@submitflags	3379
3194, 3227, 3263, 3302, 3351, 3386		\Fld@validate@code	3083, 3085
\Fld@annotnames	3126, 3143,	\Fld@value	2591, 2607, 2626, 2667, 2668, 3340
3196, 3229, 3265, 3304, 3353, 3388		\Fld@width	2592,
\Fld@bcolor	3165, 3167, 3248,	2606, 2613, 2617, 2627, 2666, 2677,	
3250, 3279, 3281, 3315, 3328, 3330		2679, 2724, 2738, 2756, 2774, 2804,	
\fld@bcolor	3236	2815, 2852, 2864, 2888, 2898, 2933	
\Fld@bordercolor	3161,	\Fld@X@additionalactions	3074, 3119, 3122
3163, 3205, 3214, 3216, 3235, 3244,		Form@action	3377
3246, 3275, 3277, 3313, 3324, 3326,		\G@refundefinedtrue	461
3359, 3367, 3369, 3394, 3402, 3404		\H@old@part	3535, 3546
\Fld@borderstyle	3147,	\H@old@chapter	3505, 3514
3200, 3233, 3269, 3308, 3357, 3408		\H@old@sect	3556, 3567, 3576
\Fld@borderwidth	2684, 2685, 3147,	\H@old@spart	3548, 3554
3200, 3233, 3269, 3308, 3357, 3408		\H@old@ssect	3491, 3496
\Fld@calculate@code	2470, 3087, 3089	\href@	273, 279
\Fld@calculate@sortkey	2481	\href@split	279, 280
\Fld@cbsymbol	3169	\Hy@abspage	597
\Fld@charsize	2683, 3173, 3254, 3285, 3336, 3392	\Hy@activeanchorfalse	648
\Fld@checkequal	2635, 2702	\Hy@activeanchortrue	639
\Fld@choices	3256	\Hy@AtBeginDocument	2435, 2451
\Fld@color	3174, 3176, 3255, 3286, 3337	\Hy@bookmarkstype	1489
\Fld@default	2590, 2607, 2625, 2668, 2697,	\Hy@chapapp	3509, 3520
2698, 2886, 3287, 3291, 3292, 3339		\Hy@colorlink	34
\Fld@flags	3145,	\Hy@currentbookmarklevel	3448
3198, 3231, 3267, 3306, 3355, 3390		\Hy@DisableOption	171
\Fld@format@code	3079, 3081	\Hy@drafttrue	1420
\Fld@height	2593, 2614, 2617, 2628, 2686, 2725,	\Hy@escapeform	2611, 2708, 2752, 2780,
2738, 2756, 2805, 2853, 2889, 2934		2819, 2866, 2904, 3012, 3019, 3022	
\Fld@keystroke@code	3075, 3077	\Hy@escapestring	3014,
\Fld@listcount	2696, 2704, 2739, 2746	3016, 3021, 3024, 3025, 3028, 3029,	
\Fld@mappingname	3058, 3060, 3132, 3134	3038, 3041, 3077, 3081, 3085, 3089,	
\Fld@maxlen	3342, 3344	3093, 3097, 3101, 3105, 3109, 3113,	
\Fld@menulength	2630, 2636, 2683	3169, 3220, 3283, 3339, 3340, 3377	
\Fld@name	2589, 2624, 2762, 2885, 3053, 3127	\Hy@finaltrue	1425
\Fld@onblur@code	3095, 3097	\Hy@FormObjects	2511, 2945, 3007
\Fld@onclick@code	3220	\Hy@gttemp	3070, 3072
\Fld@onenter@code	3107, 3109	\Hy@href	273
\Fld@onexit@code	3111, 3113	\Hy@href@nextactionraw	237
\Fld@onfocus@code	3091, 3093	\Hy@href@page	225
\Fld@onmousedown@code	3099, 3101	\Hy@linkfileprefix	1434
\Fld@onmouseup@code	3103, 3105	\Hy@linktoc	1445
\Fld@pageobjref	2559, 2573, 3010, 3142,	\Hy@MakeCurrentHref	198
3195, 3228, 3264, 3303, 3352, 3387		\Hy@MakeCurrentHrefAuto	3493, 3509, 3520, 3550, 3564
\Fld@radiosymbol	3283	\Hy@next	3519, 3528, 3531
\Fld@rotation	3157,	\Hy@numberline	164, 3449
3159, 3203, 3210, 3212, 3234, 3240,		\Hy@org@chapter	3517, 3532
3242, 3271, 3273, 3311, 3320, 3322,		\Hy@OutlineName	3465
3358, 3363, 3365, 3393, 3398, 3400		\Hy@OutlineRerunCheck	3463
		\Hy@pdfmajorversion	166

<code>\Hy@pdfminorversion</code>	165	<code>\hyper@link</code>	33, 34, 677
<code>\Hy@pdfstringtrue</code>	29, 569	<code>\hyper@linkend</code>	33, 34, 705
<code>\Hy@pdfversion</code>	3065	<code>\hyper@linkfile</code>	323, 756
<code>\Hy@pstringdef</code>	579, 3045	<code>\hyper@linklaunch</code>	36, 340, 814, 865
<code>\Hy@PutCatalog</code>	585	<code>\hyper@linknamed</code>	38, 174, 871
<code>\Hy@raisedlink</code>	3483, 3510, 3521, 3551	<code>\hyper@linkstart</code>	33, 34, 693
<code>\Hy@RestoreLastskip</code>	624	<code>\hyper@linkurl</code>	306, 367, 718
<code>\Hy@safe@activestru</code>	734, 2473	<code>\hyper@normalise</code>	273, 302, 318, 335, 361, 374
<code>\Hy@SaveLastskip</code>	604	<code>\HyPL@Labels</code>	587, 597
<code>\Hy@secnum@part</code>	3538, 3540, 3543	<code>\HyPL@StorePageLabel</code>	585
<code>\Hy@SectionAnchorHref</code>	3474, 3498, 3569	<code>\HyPsd@SanitizeForOutFile</code>	3452
<code>\Hy@SectionHShift</code>	3473, 3482, 3494, 3565	<code>\if@filesw</code>	2453, 2476, 2495
<code>\Hy@StepCount</code>	2636, 2704	<code>\ifFld@combo</code>	2651, 2681
<code>\Hy@temp</code>	2474, 2489, 3027, 3028	<code>\ifFld@hidden</code>	2606, 2666
<code>\Hy@unicodefalse</code>	3067	<code>\ifFld@multiline</code>	2594
<code>\Hy@VerboseAnchor</code>	605	<code>\ifFld@popdown</code>	2652
<code>\Hy@VerboseLinkStart</code>	681, 697	<code>\ifFld@radio</code>	2648, 2671
<code>\Hy@VerboseLinkStop</code>	688, 711, 744, 804, 854, 892	<code>\ifHy@implicit</code>	3469
<code>\Hy@VersionChecked</code>	157	<code>\ifHy@pdfescapeform</code>	3013, 3023
<code>\Hy@WrapperDef</code>	601	<code>\ifHy@unicode</code>	3066
<code>\Hy@xspace@end</code>	685, 743, 803, 853, 891	<code>\kv@set@family@handler</code>	177
<code>\HyAnn@AbsPageLabel</code>	2558, 2564, 2610, 2707, 2751, 2779, 2818, 2865, 2903	<code>\ltx@empty</code>	2430, 2431, 2470, 2557, 2558, 2559, 2590
<code>\HyAnn@Count</code>	2562, 2563, 2566, 2567, 2568, 2569, 2570, 2571, 2575, 2579	<code>\ltx@gobble</code>	2433
<code>\HyField@AddToFields</code>	2464, 2618, 2740, 2757, 2788, 2844, 2873, 2937	<code>\ltx@gobbletwo</code>	2434
<code>\HyField@afields</code>	2430	<code>\ltx@IfUndefined</code>	3516, 3526
<code>\HyField@AfterAuxOpen</code>	2435, 2461, 2494	<code>\ltx@one</code>	2566
<code>\HyField@AuxAddToCoFields</code>	2434, 2446, 2458, 2479	<code>\ltx@ReturnAfterFi</code>	3046
<code>\HyField@AuxAddToFields</code>	2433, 2439, 2455, 2497	<code>\ltx@zero</code>	2563
<code>\HyField@cofields</code>	2431	<code>\m@ne</code>	3525, 3540
<code>\HyField@FlagsCheckBox</code>	2895	<code>\OBJ@OCG@view</code>	163
<code>\HyField@FlagsChoice</code>	2676	<code>\p@</code>	2640
<code>\HyField@FlagsPushButton</code>	2771, 2811, 2862	<code>\pdf@ifdraftmode</code>	2509
<code>\HyField@FlagsRadioButton</code>	2672	<code>\PDF@SetupDoc</code>	158
<code>\HyField@FlagsSubmit</code>	2812	<code>\PDFForm@Check</code>	2904, 2936, 3138
<code>\HyField@FlagsText</code>	2605	<code>\PDFForm@List</code>	2752, 2755, 3225
<code>\HyField@PDFChoices</code>	2749	<code>\PDFForm@Name</code>	2604, 2665, 2768, 2894, 3052
<code>\HyField@SetKeys</code>	2601, 2642, 2645, 2664, 2764, 2807, 2855, 2890	<code>\PDFForm@Push</code>	2780, 2786, 3191
<code>\HyPat@ObjRef</code>	3421, 3428, 3435, 3442	<code>\PDFForm@Radio</code>	2708, 2728, 3260
<code>\hyper@@link</code>	281	<code>\PDFForm@Reset</code>	2866, 2872, 3384
<code>\hyper@anchor</code>	627	<code>\PDFForm@Submit</code>	2819, 2838, 3349
<code>\hyper@anchorend</code>	627, 3484, 3511, 3522, 3552	<code>\PDFForm@Text</code>	2611, 2616, 3299
<code>\hyper@anchorstart</code>	627, 3484, 3511, 3522, 3552	<code>\protected@edef</code>	653
		<code>\strip@pt</code>	3173, 3254, 3285, 3336, 3392
		<code>\toks@</code>	3496, 3502, 3567, 3573
		<code>\tw@</code>	3497, 3502, 3568, 3573
		<code>\url@</code>	374, 377
		<code>\Url@def</code>	375
		<code>\Url@HyperHook</code>	375
		<code>\XR@ext</code>	1431
		<code>\z@</code>	374, 2631, 3234, 3240, 3271, 3358, 3363, 3393, 3398, 3538

