

The `hyperref-generic` module

A generic driver for `hyperref`

The L^AT_EX Project*

Version 0.96b, released 2023-11-05

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

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

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

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

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

1 Avoiding transition problems

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

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

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

- This driver uses the `l3color` module for the colors. All colors defined with `\color_set:nn` or `\color_set:nnn` will work. Colors defined with `xcolor` will work if they don't use one of the special color models not supported by `l3color` as `pdfmanagement-firstaid` contains a patch for `xcolor`. If the package `color` is used it is currently recommended to define colors after `hyperref`.
- Load a color package or `graphicx` to get the right page sizes.
- Report problems! Only known problem can be resolved.

2 Bookmarks / outlines

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

3 “Metadata”

“Metadata”, informations about the document, are stored in a PDF in two places: The `/Info` dictionary and the XMP-metadata. `hyperref` only handles the `/Info` dictionary. The XMP-metadata are added by code from `l3pdfmeta`. (without the `pdfmanagement` the XMP-metadata can be added with packages like `pdfx` and `hyperxmp`).

The `/Info` dictionary can be filled with arbitrary keys, but the PDF viewer typically care only about a few, like `/Author`, `/Title` and `/Keywords`. A number of `/Info` keys, like dates and the producer, are added automatically by the engines and backends. Some of them can only be removed with special commands, some not at all. But—with the exception of `/Producer` when using the dvips backend—they can be overwritten.

The current handling of the metadata is problematic:

- External package like `hyperxmp` wants to access them too and for this had to patch a number of internal `hyperref` commands—which is a problem if the internal commands change (as happens with this new driver)
- `hyperref` (and also `hyperxmp`) tries to deduce some datas from document commands like `\title` or `\author`—something that worked reasonably well when only some standard classes with well-known definitions of these command existed, but gets problematic with classes and packages which define more powerful commands knowing a variety of optional arguments to set authors and affiliations and title information.

To resolve some of this problem the driver will

- *Not* try deduce author and title from documents. They have to be set in `\hypersetup` with `pdfauthor` and `pdftitle`. It is recommended to separate more than one author by commas, and to hide commas inside braces if needed:

```
pdfauthor = {Bär, Peter Anteater, {Riley, the sloth}}
```

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

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

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

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

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

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

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

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

4 Dates

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

5 PDF page size (mediabox)

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

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

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

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

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

6 Commands to create “external” references

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

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

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

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

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

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

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

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

6.1 Special problem: links to files

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

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

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

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

6.2 Splits

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

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

For this reason three new commands have been create:

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

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

6.3 Options

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

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

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

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

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

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

Some main changes are

- The default colors have been changed.

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

7.1 Background information

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

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

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

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

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

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

7.2 New Keys

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

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

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

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

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

The default is `phelype`.

7.3 Public interfaces

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

```
\l_hyp_annot_colorlink_bool,
\l_hyp_annot_colorurl_bool,
\l_hyp_annot_colorfile_bool,
\l_hyp_annot_colorrund_bool,
\l_hyp_annot_colormenu_bool,
\l_hyp_annot_ocgcolorlink_bool,
\l_hyp_annot_ocgcolorurl_bool,
\l_hyp_annot_ocgcolorfile_bool,
\l_hyp_annot_ocgcolorrund_bool,
\l_hyp_annot_ocgcolormenu_bool,
```

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

`colorlinks` uses the label `hyp/color`, and `ocgcolorlinks` the label `hyp/ocg`.

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

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

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

7.4 Changed behaviour

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

Color keys accept the following input syntax:

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

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

frenchlinks The option `frenchlinks` does nothing at all.

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

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

8 PDF strings

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

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

The new method is under heavy development!

Important differences here are

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

9 Package options from hyperref

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

Options that still must be set as package options are

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

Options that can be without problems set as package options are

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

Ignored options:

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

10 Disabling links

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

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

11 Draftmode

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

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

12 Dropped options

A number of options are ignored by this driver

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

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

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

unicode This is always true.

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

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

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

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

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

13 Destinations

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

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

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

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

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

The manual of `hyperref` gives a bit the impression as if this coordinates can be set manually by the user but as described above this is mostly wrong: It is for normal destination only possible with a dvi-backend like `dvips` which make use of `pdfmark.def`. `pdftex` and `luatex` can use manual coordinates only for `pdfstartview` and `pdfremotestartview`. As `dvips` was the first driver of `hyperref` the option `pdfview` was at first developed for it and then adapted to `pdftex`. But this had the effect that the handling of the option `pdfview` is inconsequent across the backend and engines: For example with `pdfview=FitH_100` `pdftex` ignores the number and calculates its own, while `dvips` sets the coordinate to the absolute 100. The zoom factor of `XYZ` is not supported by the `pdftex` driver at all, and `FitR` only partially.

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

<code>pdfview (setup key)</code> <code>pdfstartview (setup key)</code> <code>pdfremotestartview (setup key)</code>	<p>So with this driver the options <code>pdfview</code>, <code>pdfstartview</code> and <code>pdfremotestartview</code> take the following options:</p> <ul style="list-style-type: none"> • <code>Fit</code>, <code>FitB</code>, <code>FitH</code>, <code>FitV</code>, <code>FitBH</code>, <code>FitBV</code> which can be followed by a positive integer (separated by a space) or the keyword <code>null</code>. The number can be gives as a <i><dimension expression></i> surrounded with <code>\hypercalcbp</code>. The driver redefines this command to use <code>\dim_to_decimal_in_bp:n</code>.
--	--

- `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 `LaTeX` the automatic calculation of the coordinates from the encompassing box. With dvips and (x)dvipdfmx it will fall back to `Fit`. `pdfstartview` and `pdfremotestartview` will fallback to `Fit` too.

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

- `pdfview` will use the values to calculate coordinates relative to the current location. So `0_100_200_400` will give a “box” of width 200bp, height 400bp and depth 100dp that the destination should encompass. Missing numbers will be set to 0. But one should be aware that it is quite unpredictable how viewers which support `FitR` handles zero sizes.
- `pdfstartview` and `pdfremotestartview` will pass the values as absolute coordinates.

13.1 Names of destinations

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

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

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

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

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

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

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

14 Assorted key descriptions

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

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

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

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

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

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

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

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

will create

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

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

pdfborder (*setup key*) This key set accept as value three numbers or three numbers and an array describing a dash pattern, examples are `0_0_1` or `0_0_1_[3_2]`. The first two numbers should

linkborder (*setup key*)

urlborder (*setup key*)

runborder (*setup key*)

menuborder (*setup key*)

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

<code>pdfborderstyle</code> (<i>setup key</i>)	The value of this key is the content of the BS dictionary. As an example		
<code>linkborderstyle</code> (<i>setup key</i>)	<code>/Type/Border/W1/S/U/D[3,2]</code>		
<code>urlborderstyle</code> (<i>setup key</i>)	Key	Values	comment / example
<code>fileborderstyle</code> (<i>setup key</i>)	<code>/Type</code>	<code>/Border</code>	optional
<code>runborderstyle</code> (<i>setup key</i>)	<code>/W</code>	<code><number></code>	Width of border line
<code>menuborderstyle</code> (<i>setup key</i>)	<code>/S</code>	<code>/S</code>	solid (default)
		<code>/D</code>	dash pattern can be set with <code>/D</code>
		<code>/B</code>	beveled
		<code>/I</code>	inset
		<code>/U</code>	underline
	<code>/D</code>	<code><array of numbers></code>	dash pattern (lines/gaps) (default [3])
<code>pdfcreationdate</code> (<i>setup key</i>)	Setting these keys is normally not needed. If they are used the values of the first		
<code>pdfmoddate</code> (<i>setup key</i>)	two keys are stored directly in the Info dictionary for <code>/Creationdate</code> and <code>/ModDate</code> .		
<code>pdfmetadate</code> (<i>setup key</i>)	All three keys are used in XMP-metadata. The values are converted to strings but not processed further, so they should have the correct PDF format without the enclosing parentheses, e.g. <code>D:20200202111111+01'00'</code> .		
<code>pdflinkmargin</code> (<i>setup key</i>)	As described in the <code>hyperref</code> manual the behaviour differs between the backends: with dvips it is possible to change links locally, pdf _l atex and lua _l atex work by page, with dvipdfmx the setting is global (and has to be done in the preamble).		
<code>pdflang</code> (<i>setup key</i>)	The key will work, but it is recommended to set the language in <code>\DocumentMetadata</code> instead.		

File I

hyperref-generic driver implementation

```

1 <*package>
2 <@@=hyp>
3 \ProvidesFile{hgeneric-testphase.def}[2023-11-05 v0.96b %
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}
157 \cs_generate_variant:Nn\pdf_pageobject_ref:n {e}

```

3 Overwriting/providing commands from hyperref

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

```

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

159 %\cs_set_protected:Npn \PDF@SetupDoc{}
160 %\PDF@FinishDoc{}% dummy needed for hyperref ...

```

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

`\hypercalcbp`

```

161 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n

```

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

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

```

162 \providecommand\@pdfborder{}
163 \providecommand\@pdfborderstyle{}
164 \newcommand\OBJ@OCG@view {} % needed in hyperref
165 \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:

```

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

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

```

`\Acrobatmenu` should use the new internal link command

```

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

```

`\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!

```

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

```

```

185 % TODO for now unknown keys should only give warnings.
186 \keys_define:nn { hyp / setup }
187 {
188   unknown .code:n =
189   {
190     \msg_warning:nneee { hyp } { unknown-key }
191     { hyp~/~setup }{ \l_keys_key_str } { #1 }
192   }
193 }

```

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.

```

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

```

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`)

```

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

```

```

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

```

```

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

```

```

338
339 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
340 {
341   \exp_args:Nno\hyper@linklaunch{#1}{#2}{\l__hyp_href_run_parameter_tl}
342   \group_end:
343   \hook_use:n{cmd/hrefrun/after}
344 }
345
346
347 \hook_new_pair:nn{cmd/url/before}{cmd/url/after}
348
349 \DeclareRobustCommand*\url{[1] []}
350 {
351   \mode_leave_vertical:
352   \hook_use:n{cmd/url/before}
353   \group_begin:
354   \keys_set:nn {hyp / href } {#1}
355   \bool_if:NTF \l__hyp_href_url_encode_bool
356   {
357     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
358   }
359   {
360     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
361   }
362   \hyper@normalise\__hyp_href_url_aux:n
363 }
364
365 \cs_new_protected:Npn \__hyp_href_url_aux:n #1
366 {
367   \exp_args:Nno
368   \hyper@linkurl{\__hyp_href_url_format: {#1}}
369   {\l__hyp_href_url_protocol_tl#1}
370   \group_end:
371   \hook_use:n{cmd/url/after}
372 }
373

```

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

```

374
375 \def\urldef#1#2{\begingroup\def\url{\hyper@normalise\url@}\setbox\z@\hbox\bgroup
376   \def\url@HyperHook##1\endgroup{\url@def{#1}{#2}}%
377   % Because hyperref breaks \urldef and does not define its own (Grrrr!)...
378   \def\url@##1{\egroup\endgroup\DeclareRobustCommand#1{#2{##1}}}%
379   #2}
380

```

make the new commands compatible with `\pdfstringdef`:

```

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

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

4.2 citecolor

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

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

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

```

423 \bool_lazy_and:nnF
424 { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
425 { \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

```

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

```

6 Reference and label commands

The code uses in older LaTeX the l3ref-tmp package which must have been loaded as pdfmanagement is a requirement. In newer LaTeX it uses the in-built property module.

```

\__hyp_property_record:nn
  \__hyp_property_if_recorded:nn 448 %
\__hyp_property_ref_undefined_warn:nn 449 \cs_if_exist:NTF \property_new:nnnn
\__hyp_property_ref:nn 450 {

```

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

```

451 \cs_new_protected:Npn \__hyp_property_record:nn #1 #2 %label/attributes
452 {
453   \@bsphack
454   \property_record:nn{#1}{#2}
455   \@esphack
456 }

```


This provides a conditional which tests if a label/property combination is known

```

457 \prg_new_eq_conditional:Nn
458 \__hyp_property_if_recorded:nn \property_if_recorded:nn { T }

```

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

```

459 \cs_new_eq:NN \__hyp_property_ref_undefined_warn:nn \property_ref_undefined_warn:nn
460 \cs_new_eq:NN \__hyp_property_ref:nn \property_ref:nn
461 }

```

and now the same with the older l3ref-tmp. This can go once LaTeX has been released.

```

462 {
463 \cs_new_protected:Npn \__hyp_property_record:nn #1 #2 %label/attributes
464 {
465 \@bsphack
466 \ref_label:nn{#1}{#2}
467 \@esphack
468 }
469
470 \prg_new_eq_conditional:Nn \__hyp_property_if_recorded:nn \ref_if_exist:nn { T }
471
472 \cs_new_protected:Npn \__hyp_property_ref_undefined_warn:nn #1 #2 %label/attribute
473 {
474 \__hyp_property_if_recorded:nnF {#1}{#2}
475 {
476 \protect\G@refundefinedtrue
477 \@latex@warning
478 {
479 Reference~'\tl_to_str:n {#1}'~with~attribute~'\tl_to_str:n {#2}'~
480 on~page~\thepage~\space undefined
481 }
482 }
483 }
484
485 \cs_new:Npn \__hyp_property_ref:nn #1 #2
486 {
487 \ref_value:nn{#1}{#2}
488 }
489 }

```

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

```

490 \cs_generate_variant:Nn \__hyp_property_record:nn {ee}
491 \prg_generate_conditional_variant:Nnn \__hyp_property_if_recorded:nn {ee} { T }
492 \cs_generate_variant:Nn \__hyp_property_ref_undefined_warn:nn {ee}
493 \cs_generate_variant:Nn \__hyp_property_ref:nn {ee}

```

(End of definition for __hyp_property_record:nn and others.)

7 Variables

7.1 Private temporary variables

At first a few generic tmp variables

```

\l__hyp_tmpa_tl
\l__hyp_tmpa_seq
\l__hyp_tmpa_int
\l__hyp_tmpa_box
\l__hyp_tmpa_str

```

```

494 \box_new:N \l__hyp_tmpa_box
495 \tl_new:N \l__hyp_tmpa_tl
496 \seq_new:N \l__hyp_tmpa_seq
497 \int_new:N \l__hyp_tmpa_int
498 \str_new:N \l__hyp_tmpa_str

```

(End of definition for \l__hyp_tmpa_tl and others.)

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

```

\l__hyp_dest_name_tmpa_tl
\l__hyp_uri_tmpa_tl
\l__hyp_filename_tmpa_tl
\__hyp_text_tmpa_str__\g__hyp_text_tmpa_str

```

TODO: document and check use!

```

499 \tl_new:N \l__hyp_dest_name_tmpa_tl
500 \tl_new:N \l__hyp_uri_tmpa_tl
501 \tl_new:N \l__hyp_filename_tmpa_tl
502 \tl_new:N \l__hyp_para_tmpa_tl
503 \str_new:N \l__hyp_text_tmpa_str
504 \str_new:N \g__hyp_text_tmpa_str

```

(End of 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.

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

(End of definition for \c__hyp_dest_undefined_tl.)

```

\c__hyp_annot_types_seq
\c__hyp_map_annot_hyp_prop
\c__hyp_map_hyp_annot_prop

```

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

```

506 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
507 {url,link,file,menu,run}
508 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
509 {
510   URI    = url,
511   GoTo   = link,
512   GoToR  = file,
513   Named  = menu,
514   Launch = run
515 }
516 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
517 {
518   url    = URI,
519   link   = GoTo,
520   file   = GoToR,
521   menu   = Named,
522   run    = Launch
523 }
524

```

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

7.3 Variables

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

```
525 \tl_new:N \g__hyp_dest_pdfstartpage_tl
526 \tl_new:N \g__hyp_dest_pdfstartview_tl
527 \tl_new:N \l__hyp_dest_pdfremotestartview_tl
```

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

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

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

528 \tl_new:N \l__hyp_text_enc_uri_print_tl
529 \tl_new:N \l__hyp_text_enc_info_print_tl
530 \tl_new:N \l__hyp_text_enc_dest_tl
531 \tl_new:N \l__hyp_text_enc_dest_print_tl
532 \tl_new:N \l__hyp_text_enc_file_print_tl
533 \tl_new:N \l__hyp_text_enc_para_print_tl
534
535 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
536 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
537 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
538 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
539 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
540 \tl_set:Nn \l__hyp_text_enc_para_print_tl {utf8/string}
```

(End of definition for \l__hyp_text_enc_uri_print_tl and others.)

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

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

(End of definition for \l__hyp_dest_pdfview_tl.)

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

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

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

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

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

(End of definition for \g__hyp_bordercolormodel_str.)

7.4 Booleans

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

```
\l_hyp_annot_colorlink_bool
\l_hyp_annot_colorurl_bool
\l_hyp_annot_colorfile_bool
\l_hyp_annot_colorrund_bool
\l_hyp_annot_colormenu_bool
543 \seq_map_inline:Nn \c__hyp_annot_types_seq
544 {
545   \bool_new:c {l_hyp_annot_color#1_bool}
546 }
```

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

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

```
\l_hyp_annot_occolorlink_bool
\l_hyp_annot_occolorurl_bool
\l_hyp_annot_occolorfile_bool
\l_hyp_annot_occolorrund_bool
\l_hyp_annot_occolormenu_bool
547 \seq_map_inline:Nn \c__hyp_annot_types_seq
548 {
549   \bool_new:c {l_hyp_annot_occolor#1_bool}
550 }
```

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

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

```
551 \seq_map_inline:Nn \c_pdfannot_link_types_seq
552 {
553   \bool_new:c {l__hyp_annot_#1_bool}
554   \bool_set_true:c {l__hyp_annot_#1_bool}
555 }
```

(End of definition for `\l__hyp_annot_GoTo_bool` `\l__hyp_annot_URI_bool` `\l__hyp_annot_GoToR_bool` `\l__hyp_annot_Named_bool` `\l__hyp_annot_Launch_bool`.)

7.5 Boxes

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

```
556 \box_new:N \l__hyp_dest_box
```

(End of definition for `\l__hyp_dest_box`.)

7.6 Regex

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

```
557 \regex_const:Nn \c__hyp_dest_startview_regex
558 {
559   \A\ *
560   (?
561     (?:XYZ (?:\ +(?:?:\d+|\d*\.\d+)|null)){3}\ )
562     |
563     (?:Fit\b|FitB\b)
564     |
565     (?:?:FitH|FitV|FitBH|FitBV)(?:\ +(?:?:\d+|\d*\.\d+)|\ +null){1}
566     |
567     (?:FitR (?:\ +\d+|\ +\d*\.\d+)){4}\ )
568   )
569 }
```

(End of definition for \c__hyp_dest_startview_regex.)

7.7 PDF dictionaries

l__hyp_page/Trans This dictionary is used for page transitions.

```
570 \pdfdict_new:n {l__hyp_page/Trans}
571 \pdfdict_put:nnn {l__hyp_page/Trans}{Type}{/Trans}
```

(End of definition for l__hyp_page/Trans.)

8 PDF string conversion

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

hyp/text/pdfstring

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

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

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

__hyp_text_purify:nN

```
573 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
574 {
575   \str_set:Ne #2 {\text_purify:n { #1 } }
576 }
```

(End of definition for __hyp_text_purify:nN.)

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

__hyp_text_cleanup:N

```
577 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
578 {
579
580 }
```

(End of definition for __hyp_text_cleanup:N.)

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

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

`_hyp_text_string_from_unicode:nN`

```

581 \cs_new_protected:Npn \_hyp\_text\_string\_from\_unicode:nN #1 #2
582 {
583   \pdf\_string\_from\_unicode:nVN { #1 } #2 #2
584 }

```

(End of definition for _hyp_text_string_from_unicode:nN.)

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

`_hyp_text_pdfstring:nnN`

```

585 \cs_new_protected:Npn \_hyp\_text\_pdfstring:nnN #1 #2 #3
586 {
587   \group\_begin:
588   \Hy@pdfstringtrue
589   \hook\_use:n {hyp/text/pdfstring}
590   \_hyp\_text\_purify:nN { #1 } \l\_hyp\_text\_tmpa\_str
591   \_hyp\_text\_cleanup:N \l\_hyp\_text\_tmpa\_str
592   \_hyp\_text\_string\_from\_unicode:nN { #2 } \l\_hyp\_text\_tmpa\_str
593   \str\_gset\_eq:NN \g\_hyp\_text\_tmpa\_str\l\_hyp\_text\_tmpa\_str
594   \group\_end:
595   \str\_set\_eq:NN #3 \g\_hyp\_text\_tmpa\_str
596 }
597 \cs\_generate\_variant:Nn \_hyp\_text\_pdfstring:nnN {enN,onN,eoN,ooN,noN}

```

(End of definition for _hyp_text_pdfstring:nnN.)

!!! temporary until all instances are gone

```

598 \cs\_new\_protected:Npn \Hy@pstringdef #1 #2
599 { \_hyp\_text\_pdfstring:enN {#2} {utf8/string-raw}#1 }

```

This is a special version for info keys:

`_hyp_text_pdfstring_info:nN`

```

600 \cs\_new\_protected:Npn \_hyp\_text\_pdfstring\_info:nN #1 #2
601 {
602   \_hyp\_text\_pdfstring:noN { #1 } { \l\_hyp\_text\_enc\_info\_print\_tl } #2
603 }

```

(End of definition for _hyp_text_pdfstring_info:nN.)

9 Pagelabels

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

The current implementation in hyperref/hyperref drivers:

xetex: hxtex.def, line 80-110

```

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

```

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

dvipdfm: identical to xetex

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

luatex identical to pdftex

The code in `hyperref` inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-31 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_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
604 \cs_new_protected:Npn \__hyp_PageLabels_gpush:
605   {
606     \pdfmanagement_add:nne {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
607   }
608
609 \def\Hy@PutCatalog #1 {}
610
611
612 \legacy_if:nT { Hy@pdfpagelabels }
613 {
614   \cs_set_protected:Npn \HyPL@StorePageLabel #1
615   {
616     \tl_gput_right:Nx \HyPL@Labels { \the\Hy@abspage<<#1>> }
617     \__hyp_PageLabels_gpush:
618   }
619 }
```

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

10 Core Hyperref Commands

Every `hyperref` has to define eight core command:

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

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

10.1 Anchors / destinations

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

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

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

```
\__hyp_destination:nn
620 \Hy@WrapperDef \__hyp_destination:nn #1 #2
621 {
622   \mode_if_horizontal:T { \@savs\spacefactor }
623   \Hy@SaveLastskip      %defined in hyperref
624   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
625   \__hyp_text_pdfstring:eoN
626   { \HyperDestNameFilter{#1} }
627   { \l__hyp_text_enc_dest_tl }
628   \l__hyp_tmpa_tl
629   \str_if_eq:nnTF {#2} {fitrbx}
630   {
631     \exp_args:NV
632     \pdf_destination:nnnn \l__hyp_tmpa_tl
633     { \box_wd:N \l__hyp_dest_box }
634     { \box_ht:N \l__hyp_dest_box }
635     { \box_dp:N \l__hyp_dest_box }
636   }
637   {
638     \exp_args:NV
639     \pdf_destination:nf
640     { \l__hyp_tmpa_tl }
641     { #2 }
642   }
643   \Hy@RestoreLastskip %defined in hyperref
644   \mode_if_horizontal:T { \spacefactor\@savs }
645 }
```

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

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

```
\hyper@anchor
\hyper@anchorstart 646 \tl_new:N\l_hyp_current_dest_name_tl
\hyper@anchorend
hyp/anchor
\l_hyp_current_dest_name_tl
```



```

647 \hook_new:n{hyp/anchor}
648 \cs_new_protected:Npn \hyper@anchor #1
649 {
650   \exp_args:NnV
651     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
652   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
653   \hook_use:n{hyp/anchor}
654 }
655
656 \cs_new_protected:Npn \hyper@anchorstart #1
657 {
658   \Hy@activeanchortrue
659   \exp_args:NnV
660     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
661   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
662   \hook_use:n{hyp/anchor}
663 }
664
665 \cs_new_protected:Npn \hyper@anchorend
666 {
667   \Hy@activeanchorfalse
668 }

```

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

10.2 GoTo Links

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

At first the internal link commands:

```

669 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
670 {
671   \mode_leave_vertical:
672   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
673   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
674   {
675     \msg_warning:nne
676       { hyp }
677       { empty-destination-name }
678       { \c__hyp_dest_undefined_tl }
679     \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
680   }
681   {
682     \__hyp_text_pdfstring:eoN
683     { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
684     { \l__hyp_text_enc_dest_tl }
685     \l__hyp_dest_name_tmpa_tl
686   }
687   \exp_args:No
688   \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
689 }
690

```

```

691 \cs_new_protected:Npn \__hyp_link_goto_end:
692 {
693   \pdfannot_link_goto_end:
694 }

```

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

```

695
696 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
697 {
698   \bool_if:NTF \l__hyp_annot_GoTo_bool
699   {
700     \Hy@VerboseLinkStart{#1}{#2}
701     \group_begin:
702     \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
703     \hook_use:n {hyp/link/#1}
704     \__hyp_link_goto_begin:nw {#2} #3 \Hy@xspace@end
705     \__hyp_link_goto_end:
706     \group_end:
707     \Hy@VerboseLinkStop
708   }
709   {\let\protect\relax#3}
710 }
711
712 \cs_new_protected:Npn \hyper@linkstart #1 #2 % #1 context, #2=destination name
713 {
714   \bool_if:NT \l__hyp_annot_GoTo_bool
715   {
716     \Hy@VerboseLinkStart{#1}{#2}% only for debug
717     \group_begin:
718     \tl_set:Nn \l_hyp_current_dest_name_tl {#2}

```

```

719         \hook_use:n {hyp/link/#1}
720         \__hyp_link_goto_begin:nw {#2}
721     }
722 }
723
724 \cs_new_protected:Npn \hyper@linkend
725 {
726     \bool_if:NT \l__hyp_annot_GoTo_bool
727     {
728         \__hyp_link_goto_end:
729         \group_end:
730         \Hy@VerboseLinkStop
731     }
732 }

```

10.3 URI links

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

```

733 \pdfdict_new:n {l_hyp/annot/A/URI}
734 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
735 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
736
737 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
738 {
739     \bool_if:NTF \l__hyp_annot_URI_bool
740     {
741         \group_begin:
742         \__hyp_text_pdfstring:eoN
743         { #2}
744         { \l__hyp_text_enc_uri_print_tl }
745         \l__hyp_uri_tmpa_tl
746         \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
747         \bool_if:NT \l__hyp_href_url_ismap_bool
748         {
749             \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
750         }
751         \cs_set_eq:NN \# \c_hash_str
752         \cs_set_eq:NN \% \c_percent_str
753         \Hy@safe@activetrue
754         \mode_leave_vertical:
755         \pdfannot_dict_put:nne {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
756         \pdfannot_link:nen { URI }
757         {
758         }
759         {
760             \let\protect\relax
761             #1
762             \Hy@xspace@end
763             \Hy@VerboseLinkStop %where is the start??
764         }
765         \group_end:
766     }
767     {{\let\protect\relax#1}}
768 }

```

10.4 GoToR Links files

```

770 \pdfdict_new:n {l_hyp/annot/A/GoToR}
771 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
772 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
773
774 \cs_generate_variant:Nn \pdffile_embed_file:nnn {noe}
775 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
776 {
777   \bool_if:NTF \l__hyp_annot_GoToR_bool
778   {
779     \group_begin:
780     \tl_set:Nx \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
781     \exp_args:Nx
782     \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
783     {
784       \pdfdict_put:nne { l_pdffile/Filespec } { Subtype } { \pdf_name_from_unicode_e:n {ap
785       \pdffile_embed_file:noe
786       {}
787       {\l__hyp_filename_tmpa_tl }
788       {\__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
789     }
790     \pdfdict_put:nne
791     {l_hyp/annot/A/GoToR}
792     {F}
793     {\pdf_object_ref:e {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
794     \__hyp_text_pdfstring:nnN
795     { #3 }
796     { \l__hyp_text_enc_dest_print_tl }
797     \l__hyp_dest_name_tmpa_tl
798     \tl_if_blank:eTF {#3}
799     {
800       \pdfdict_put:nne {l_hyp/annot/A/GoToR}{D}
801       {
802         [
803           \int_eval:n
804           { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
805           /\l__hyp_dest_pdfremotestartview_tl
806         ]
807       }
808     }
809     {
810       \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
811     }
812     \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>

```

813   \pdf_object_unnamed_write:ne{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
814   \pdfannot_dict_put:nne {link/GoToR}{A}{\pdf_object_ref_last:}
815   \pdfannot_link:nnn %expansion??
816   { GoToR }

```

```

817         {
818         }
819         {
820             \let\protect\relax
821             #1\Hy@xspace@end
822             \Hy@VerboseLinkStop %where is the start??
823         }
824     \group_end:
825 }
826 {{\let\protect\relax#1}}
827 }

```

10.5 Launch links

We define \hyper@linklaunch for naming consistency

```

828 \pdfdict_new:n {l_hyp/annot/A/Launch}
829 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
830 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
831
832 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
833 {
834     \bool_if:NTF \l__hyp_annot_Launch_bool
835     {
836         \group_begin:
837         \__hyp_text_pdfstring:nnN
838             { #1 }
839             { \l__hyp_text_enc_file_print_tl }
840             \l__hyp_filename_tmpa_tl
841         \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
842         \__hyp_text_pdfstring:noN
843             { #3 }
844             { \l__hyp_text_enc_para_print_tl }
845             \l__hyp_para_tmpa_tl
846         \bool_if:nTF
847             {
848                 \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
849                 ||
850                 \pdf_version_compare_p:Nn > {1.9}
851             }
852             {
853                 \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
854             }
855             {
856                 \pdfdict_put:nne
857                     {l_hyp/annot/A/Launch}
858                     {Win}
859                     {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
860             }
861         \mode_leave_vertical:
862         \pdfannot_dict_put:nne {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}>>}
863         \pdfannot_link:nen
864             { Launch }
865             {
866                 % /A
867                 % <<

```

```

868 %           \pdfdict_use:n {l_hyp/annot/A/Launch}
869 %           >>
870     }
871     {
872       \let\protect\relax
873       #2\Hy@xspace@end
874       \Hy@VerboseLinkStop %where is the start??
875     }
876   \group_end:
877 }
878 {\let\protect\relax#2}}
879 }

```

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

```

880 \use:e
881 { % filename, anchor text, linkname
882   \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str #1 \exp_not:N \ \ #2 #3
883 }
884 {
885   \hyper@linklaunch {#1}{#2}{#3}
886 }

```

10.6 Named links (menu)

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

```

887 \pdfdict_new:n {l_hyp/annot/A/Named}
888 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
889 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
890
891 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
892 {
893   \bool_if:NTF \l__hyp_annot_Named_bool
894   {
895     \group_begin:
896     \pdfmeta_standard_verify:nnTF {named_actions}{#1}
897     {
898       \mode_leave_vertical:
899       \pdfdict_put:nne {l_hyp/annot/A/Named}{N}
900       {\pdf_name_from_unicode_e:n{#1}}
901       \pdfannot_dict_put:nne {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}>>}
902       \pdfannot_link:nnn { Named }
903       {
904         /A
905         <<
906         \pdfdict_use:n { l_hyp/annot/A/Named }
907         >>
908       }
909     }
910     #2
911     \Hy@xspace@end
912     \Hy@VerboseLinkStop
913   }
914 }

```

```

915         {
916         \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
917         #2
918         }
919     \group_end:
920 }
921 {{\let\protect\relax#2}}
922 }
923

```

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.

```

924 \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 \__hyp_color_select:n {<color>}
\__hyp_color_select_aux:wn \__hyp_color_select_aux:wn {<color>}

```

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

```

925 \cs_new_protected:Npn \__hyp_color_select:n #1
926 {
927     \tl_if_head_eq_charcode:nNTF {#1}[ %]
928     {
929         \__hyp_color_select_aux:wn #1
930     }
931     {
932         \color_select:n {#1}
933     }
934 }
935
936 \cs_new_protected:Npn \__hyp_color_select_aux:wn [#1] #2
937 {
938     \color_select:nn {#1}{#2}
939 }
940
941 \cs_generate_variant:Nn \__hyp_color_select:n {e}

```

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

```

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

```

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

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

```

942 \cs_new_protected:Npn \__hyp_color_set:nn #1 #2
943 {
944   \tl_if_head_eq_charcode:nNTF {#2}[ %]
945   {
946     \__hyp_color_set_aux:nwn { #1 } #2
947   }
948   {
949     \color_set:nn {#1} {#2}
950   }
951 }
952
953 \cs_new_protected:Npn \__hyp_color_set_aux:nwn #1 [#2] #3
954 {
955   \color_set:nnn {#1}{#2}{#3}
956 }
957
958 \cs_generate_variant:Nn \__hyp_color_set:nn {ne}

```

(End of definition for `__hyp_color_set:nn` and `__hyp_color_set_aux:nwn`.)

11.2 Textcolor of links

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

```

959 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
960 {
961   \hook_gput_code:nnn
962     {pdfannot/link/#2/begin}
963     {hyp/color}
964     {
965       \bool_if:cT { l_hyp_annot_color#1_bool }
966       {
967         \group_begin:
968         \color_select:n { hyp/color/#1}
969       }
970     }
971   \hook_gput_code:nnn
972     {pdfannot/link/#2/end}
973     {hyp/color}
974     {
975       \bool_if:cT { l_hyp_annot_color#1_bool }
976       {
977         \group_end:
978       }
979     }
980 }

```

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

```

981 \keys_define:nn { hyp / setup }
982 {

```



```

983     ,colorlinks .meta:n =
984     {
985         ,pdfborder={0~0~0}
986         ,pdfborderstyle=
987         ,colorurl  =#1
988         ,colorlink =#1
989         ,colorrun  =#1
990         ,colormenu =#1
991         ,colorfile =#1
992     }
993     ,colorlinks .default:n = {true}
994 }

colorurl (setup key)
colorlink (setup key) 995 \seq_map_inline:Nn \c__hyp_annot_types_seq
colorrun (setup key) 996 {
colormenu (setup key) 997     \keys_define:nn { hyp / setup }
colorfile (setup key) 998     {
    urlcolor (setup key) 999         ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
linkcolor (setup key) 1000         ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
    runcolor (setup key) 1001     }
menucolor (setup key) 1002 }
filecolor (setup key) 1003
allcolors (setup key) 1004 \keys_define:nn { hyp / setup }
1005 {
1006     ,allcolors .meta:n =
1007     {
1008         ,urlcolor=#1
1009         ,linkcolor=#1
1010         ,runcolor=#1
1011         ,filecolor=#1
1012         ,menucolor=#1
1013     }
1014     ,allcolors .value_required:n = true
1015 }

```

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)

1016 \keys_define:nn { hyp / setup }
1017 {
1018     ,bordercolormodel .choices:nn =
1019     {rgb,cmyk}
1020     { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
1021     ,bordercolormodel .initial:n ={rgb}
1022 }

1023 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1024 {

```

```

1025 \keys_define:nn { hyp / setup }
1026 {
1027   #1bordercolor .code:n =
1028   {
1029     \tl_if_empty:nTF { ##1 }
1030     {
1031       \pdfannot_dict_remove:nn
1032       {link/#2}
1033       { C }
1034     }
1035     {
1036       \__hyp_color_set:ne {hyp/color/#1border}{##1}
1037       \color_export:nVN
1038       {hyp/color/#1border}
1039       \g__hyp_bordercolormodel_str
1040       \l__hyp_tmpa_tl
1041       \pdfannot_dict_put:nne
1042       {link/#2}
1043       { C }
1044       { [\l__hyp_tmpa_tl] }
1045     }
1046   }
1047 }
1048 }
1049
1050 \keys_define:nn { hyp / setup }
1051 {
1052   ,allbordercolors .meta:n =
1053   {
1054     ,linkbordercolor=#1
1055     ,urlbordercolor =#1
1056     ,filebordercolor=#1
1057     ,menubordercolor=#1
1058     ,runbordercolor =#1
1059   }
1060   ,allbordercolors .value_required:n = true
1061 }
1062

```

11.3.2 Borderwidth and -arc

```

1063 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1064 {
1065   \keys_define:nn { hyp / setup }
1066   {
1067     #1border .code:n =
1068     {
1069       \tl_if_empty:nTF { ##1 }
1070       {
1071         \pdfannot_dict_remove:nn
1072         {link/#2}
1073         { Border }
1074       }
1075       {

```

```

1076         \pdfannot_dict_put:nnn
1077         {link/#2}
1078         { Border }
1079         { [##1] }
1080     }
1081 }
1082 }
1083 }
1084 \keys_define:nn { hyp / setup }
1085 {
1086     ,pdfborder .code:n =
1087     {
1088         \tl_if_empty:nTF { #1 }
1089         {
1090             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1091             {
1092                 \pdfannot_dict_remove:nn
1093                 {link/##2}
1094                 { Border }
1095             }
1096         }
1097         {
1098             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1099             {
1100                 \pdfannot_dict_put:nnn
1101                 {link/##2}
1102                 { Border }
1103                 { [#1] }
1104             }
1105         }
1106     }
1107     ,pdfborder .initial:n = {0~0~1},
1108 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1109 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1110 {
runborderstyle (setup key) 1111     \keys_define:nn { hyp / setup }
fileborderstyle (setup key) 1112     {
menuborderstyle (setup key) 1113         #1borderstyle .code:n =
1114         {
1115             \tl_if_empty:nTF { ##1 }
1116             {
1117                 \pdfannot_dict_remove:nn
1118                 {link/#2}
1119                 { BS }
1120             }
1121             {
1122                 \pdfannot_dict_put:nnn
1123                 {link/#2}
1124                 { BS }

```

```

1125         { <<#1>> }
1126     }
1127 }
1128 }
1129 }
1130 \keys_define:nn { hyp / setup }
1131 {
1132     ,pdfborderstyle .code:n =
1133     {
1134         \tl_if_empty:nTF { #1 }
1135         {
1136             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1137             {
1138                 \pdfannot_dict_remove:nn
1139                 {link/##2}
1140                 { BS }
1141             }
1142         }
1143         {
1144             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1145             {
1146                 \pdfannot_dict_put:nnn
1147                 {link/##2}
1148                 { BS }
1149                 { <<#1>> }
1150             }
1151         }
1152     }
1153     ,pdfborderstyle .initial:n = {},
1154 }

```

11.4 ocgcolorlinks

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

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

```

1155 \cs_new_protected:Npn \__hyp_ocg_init:
1156 {
1157     \pdf_object_new:n { __hyp/OCG/View }
1158     \pdf_object_new:n { __hyp/OCG/Print }
1159     \pdf_object_new:n { __hyp/OCG/config }
1160     \pdf_object_new:n { __hyp/OCG/refarray }
1161     \pdf_object_write:nne { __hyp/OCG/refarray } { array }
1162     {
1163         \pdf_object_ref:n { __hyp/OCG/View }
1164         \c_space_tl
1165         \pdf_object_ref:n { __hyp/OCG/Print }
1166     }
1167     \pdf_object_write:nnn { __hyp/OCG/View } { dict }
1168     {
1169         /Type/OCG
1170         /Name(View)

```

```

1171         /Usage
1172         <<
1173         /Print <</PrintState/OFF>>~
1174         /View <</ViewState/ON >>~
1175         >>
1176     }
1177     \pdf_object_write:nnn { __hyp/OCG/Print } { dict }
1178     {
1179         /Type/OCG
1180         /Name(Print)
1181         /Usage
1182         <<
1183         /Print <</PrintState/ON>>~
1184         /View <</ViewState/OFF>>~
1185         >>
1186     }
1187     \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n { __hyp/OCG/Print } }
1188     \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n { __hyp/OCG/Print } }
1189     \pdf_object_write:nne { __hyp/OCG/config } { dict }
1190     {
1191         /OFF[\pdf_object_ref:n { __hyp/OCG/Print } ]
1192         /AS[
1193             <<
1194             /Event/View
1195             /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1196             /Category[/View]
1197             >>
1198             <<
1199             /Event/Print
1200             /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1201             /Category[/Print]
1202             >>
1203             <<
1204             /Event/Export
1205             /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1206             /Category[/Print]
1207             >>
1208         ]
1209     }
1210     \pdfmanagement_add:nne { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG/Print } }
1211     \cs_gset:Npn \__hyp_ocg_init: { }
1212 }

```

(End of definition for __hyp_ocg_init:.)

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

```

1213 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1214 {
1215     \hook_gput_code:nnn
1216     {pdfannot/link/#2/begin}
1217     {hyp/ocg}
1218     {
1219         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1220         {

```

```

1221         \__hyp_ocg_init:
1222         \group_begin:
1223         \hbox_set:Nw \l__hyp_tmpa_box
1224     }
1225 }
1226 \hook_gput_code:nnn
1227 {pdfannot/link/#2/end}
1228 {hyp/ocg}
1229 {
1230     \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1231     {
1232         \hbox_set_end:
1233         \mbox
1234         {
1235             \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1236             \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1237             \pdf_emc:
1238             \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1239             \group_begin:
1240             \color_select:n { hyp/color/#1 }
1241             \box_use_drop:N \l__hyp_tmpa_box
1242             \group_end:
1243             \pdf_emc:
1244         }
1245         \group_end:
1246     }
1247 }
1248 }

```

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

```

1249 \bool_lazy_or:nnTF
1250 { \pdf_version_compare_p:Nn > {1.4} }
1251 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1252 {
1253     \keys_define:nn { hyp / setup }
1254     {
1255         ,_ocgcolorlinks .meta:n =
1256         {
1257             ocgcolorlink=#1,
1258             ocgcolorurl=#1,
1259             ocgcolorfile=#1,
1260             ocgcolorrun=#1,
1261             ocgcolormenu=#1
1262         }
1263         ,_ocgcolorlinks .default:n = true
1264     }
1265 }
1266 {
1267     \keys_define:nn { hyp / setup }
1268     {
1269         ,_ocgcolorlinks .code:n =
1270         {
1271             \msg_warning:nnee
1272             { hyp }

```

```

1273         { ignore-deprecated-or-unknown-option-in-pdf-version }
1274         { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1275     }
1276 }
1277 }
1278
1279 \keys_define:nn { hyp / setup }
1280 {
1281     ,ocgcolorlinks .choice:
1282     ,ocgcolorlinks / true .meta:n =
1283     {
1284         pdfborder      = {0~0~0},
1285         pdfborderstyle = {},
1286         colorlinks     = false,
1287         _ocgcolorlinks = true
1288     }
1289     ,ocgcolorlinks / false .meta:n =
1290     {
1291         _ocgcolorlinks = false
1292     }
1293     ,ocgcolorlinks .default:n = {true}
1294 }
1295
1296 \seq_map_inline:Nn \c__hyp_annot_types_seq
1297 {
1298     \bool_lazy_or:nnTF
1299     { \pdf_version_compare_p:Nn > {1.4} }
1300     { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1301     {
1302         \keys_define:nn { hyp / setup }
1303         {
1304             ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1305         }
1306     }
1307     {
1308         \keys_define:nn { hyp / setup }
1309         {
1310             ,ocgcolor#1 .code:n=
1311             {
1312                 \msg_warning:nnee
1313                 { hyp }
1314                 { ignore-deprecated-or-unknown-option-in-pdf-version }
1315                 { ocgcolor#1 }
1316                 { \pdf_version_major:.\pdf_version_minor: }
1317             }
1318         }
1319     }
1320 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1321 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1322 {

```

```

1323 \keys_define:nn { hyp / setup }
1324 {
1325   ,#1highlight .choices:nn =
1326   { /I, /N, /O, /P}
1327   {
1328     \pdfannot_dict_put:nnn
1329     {link/#2}
1330     { H }
1331     { ##1 }
1332   }
1333   ,#1highlight / .code:n =
1334   {
1335     \pdfannot_dict_remove:nn
1336     {link/#2}
1337     { H }
1338   }
1339   ,#1highlight / unknown .code:n =
1340   {
1341     \msg_warning:nneee { hyp } { unknown-choice+empty }
1342     { pdfhighlight }
1343     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1344     { \exp_not:n {##1} }
1345   }
1346 }
1347 }
1348 }
1349 }
1350
1351
1352 \keys_define:nn { hyp / setup }
1353 {
1354   ,pdfhighlight .choices:nn =
1355   { /I, /N, /O, /P}
1356   {
1357     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1358     {
1359       \pdfannot_dict_put:nnn
1360       {link/####2}
1361       { H }
1362       { #1 }
1363     }
1364   }
1365   ,pdfhighlight / .code:n =
1366   {
1367     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1368     {
1369       \pdfannot_dict_remove:nn
1370       {link/##2}
1371       { H }
1372     }
1373   }
1374   ,pdfhighlight .initial:n = {/I},
1375   ,pdfhighlight / unknown .code:n =
1376   {

```



```

1377     \msg_warning:nneee { hyp } { unknown-choice+empty }
1378     { pdfhighlight }
1379     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1380     { \exp_not:n {#1} }
1381   }
1382 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1383 \keys_define:nn { hyp / setup }
hideurl (setup key) 1384 {
hidefile (setup key) 1385   hidelinks .meta:n =
hiderun (setup key) 1386   {
hidemenu (setup key) 1387     ,colorlinks      = false
1388     ,ocgcolorlinks = false
1389     ,pdfborder      = { 0~0~0 }
1390     ,pdfborderstyle=
1391   }
1392 }
1393
1394 \seq_map_inline:Nn \c__hyp_annot_types_seq
1395 {
1396   \keys_define:nn { hyp / setup }
1397   {
1398     hide#1 .meta:n =
1399     {
1400       ,color#1      = false
1401       ,ocgcolor#1   = false
1402       ,#1border     = { 0~0~0 }
1403       ,#1borderstyle =
1404     }
1405   }
1406 }

```

11.7 color schemes and settings

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

```

colorscheme (setup key)
1407 \keys_define:nn { hyp / setup }
1408 {
1409   colorscheme .code:n =
1410   {
1411     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1412     {
1413       \keys_set:nn { hyp /setup }
1414       {
1415         ##1 = ##2
1416       }
1417     }
1418   }

```

```

1419 }
1420 \keys_set:nn { hyp / setup } {colorscheme=phetype}

```

12 Keys

12.1 Ignored keys

The following are ignored (with or without warnings)

```

unicode (setup key)
pdfencoding (setup key) 1421 \keys_define:nn { hyp / setup }
pdfversion (setup key) 1422 {
1423   ,unicode .code:n = {}
1424   ,pdfencoding .code:n = {}
1425   ,pdfversion .code:n =
1426   {
1427     \msg_warning:nn { hyp }{ pdfversion-disabled }
1428   }
1429 }
1430 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
debug (setup key) 1431 \keys_define:nn { hyp / setup }
draft (setup key) 1432 {
final (setup key) 1433   ,verbose .legacy_if_set:n = {Hy@verbose}
1434   ,debug .legacy_if_set:n = {Hy@verbose}
1435 }
1436 \keys_define:nn { hyp / setup }
1437 {
1438   ,draft .code:n =
1439   {
1440     \Hy@drafttrue
1441     \PassOptionsToPackage{draft}{bookmark}
1442   }
1443   ,final .code:n =
1444   {
1445     \Hy@finaltrue
1446     \PassOptionsToPackage{final}{bookmark}
1447   }
1448 }

extension (setup key)
hypertextnames (setup key) 1449 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1450 {
pageanchor (setup key) 1451   ,extension .tl_set:N = \XR@ext
linktoc (setup key) 1452   ,extension .initial:n= pdf
linktocpage (setup key) 1453   ,hypertextnames .legacy_if_set:n = {Hy@hypertextnames}
plainpages (setup key) 1454   ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
localanchorname (setup key) 1455   ,localanchorname .legacy_if_set:n = {Hy@localanchorname}
linkfileprefix (setup key)

```

```

1456 ,naturalnames .legacy_if_set:n = {Hy@naturalnames}
1457 ,pageanchor .legacy_if_set:n = {Hy@pageanchor}
1458 ,plainpages .legacy_if_set:n = {Hy@plainpages}
1459 }
1460
1461 \keys_define:nn { hyp / setup }
1462 {
1463 ,linktoc .choices:nn = { none, section, all, page }
1464 {
1465 \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1466 }
1467 ,linktoc / unknown .code:n =
1468 {
1469 \msg_warning:nneee { hyp } { unknown-choice }
1470 { linktoc }
1471 { none, section, all, page }
1472 { \exp_not:n {#1} }
1473 }
1474 ,linktocpage .choice:
1475 ,linktocpage / true .meta:n = {linktoc=page}
1476 ,linktocpage / false .meta:n = {linktoc=section}
1477 ,linktocpage .default:n = true
1478 }
1479

```

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

`url (setup key)` 1480 `\prop_map_inline:Nn \c__hyp_map_hyp_annot_prop`

`file (setup key)` 1481 {

`menu (setup key)` 1482 `\keys_define:nn { hyp / setup }`

`run (setup key)` 1483 {

1484 ,#1 .bool_set:c = {l__hyp_annot_#2_bool}

1485 }

1486 }

1487 `\keys_define:nn { hyp / setup }`

1488 {

1489 ,baseurl .code:n =

1490 {

1491 `__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl`

1492 `\tl_if_empty:NTF \l__hyp_tmpa_tl`

1493 {

1494 `\pdfmanagement_remove:nn {Catalog} { URI }`

1495 }

1496 {

1497 `\pdfmanagement_add:nne {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }`

1498 }

1499 `__hyp_store_metadata:nn {baseurl}{#1}`

1500 }

1501 `%only false does something ...`

1502 ,bookmarks .choice:

1503 ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}

1504 ,bookmarks / true .code:n = {}

1505 ,bookmarks .default:n = {true}

1506 ,bookmarksnumbered .legacy_if_set:n = {Hy@bookmarksnumbered}

```

1507 ,bookmarksopen .legacy_if_set:n = {Hy@bookmarksopen}
1508 ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1509 ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1510 ,pdfcenterwindow .choice:
1511 ,pdfcenterwindow / false .code:n =
1512 {
1513     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1514 }
1515 ,pdfcenterwindow / true .code:n =
1516 {
1517     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1518 }
1519 ,pdfcenterwindow / .code:n =
1520 {
1521     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1522 }
1523 ,pdfcenterwindow / unknown .code:n =
1524 {
1525     \msg_warning:nnee { hyp } { no-bool }
1526     { pdfcenterwindow }
1527     { \exp_not:n {#1} }
1528 }
1529 ,pdfcenterwindow .default:n = true
1530 ,pdfdirection .choice:
1531 ,pdfdirection / L2R .code:n =
1532 {
1533     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1534 }
1535 ,pdfdirection / R2L .code:n =
1536 {
1537     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1538 }
1539 ,pdfdirection / .code:n =
1540 {
1541     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1542 }
1543 ,pdfdirection / unknown .code:n =
1544 {
1545     \msg_warning:nneee { hyp } { unknown-choice+empty }
1546     { pdfdirection }
1547     { L2R , R2L }
1548     { \exp_not:n {#1} }
1549 }
1550 ,pdfdisplaydoctitle .choice:
1551 ,pdfdisplaydoctitle / false .code:n =
1552 {
1553     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1554 }
1555 ,pdfdisplaydoctitle / true .code:n =
1556 {
1557     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1558 }
1559 ,pdfdisplaydoctitle .default:n = true
1560 ,pdfduplex .choices:nn =

```

```

1561 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1562 {
1563   \pdf_version_compare:NnTF > {1.6}
1564   {
1565     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1566     { PrintDuplex } { /#1 }
1567   }
1568   {
1569     \msg_warning:nnee
1570     {hyp}
1571     {ignore-deprecated-or-unknown-option-in-pdf-version}
1572     {pdfduplex}
1573     {\pdf_version:}
1574   }
1575 }%
1576 ,pdfduplex / .code:n =
1577 {
1578   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1579 }
1580 ,pdfduplex / unknown .code:n =
1581 {
1582   \msg_warning:nneee { hyp } { unknown-choice+empty }
1583   { pdfduplex }
1584   { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1585   { \exp_not:n {#1} }
1586 }
1587 ,pdffitwindow .choice:
1588 ,pdffitwindow / false .code:n =
1589 {
1590   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1591 }
1592 ,pdffitwindow / true .code:n =
1593 {
1594   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1595 }
1596 ,pdffitwindow / .code:n =
1597 {
1598   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1599 }
1600 ,pdffitwindow .default:n = true
1601 ,pdffitwindow / unknown .code:n =
1602 {
1603   \msg_warning:nnee { hyp } { no-bool }
1604   { pdffitwindow }
1605   { \exp_not:n {#1} }
1606 }
1607 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1608 ,pdflinkmargin .initial:n = {1pt}
1609 ,pdfmenubar .choice:
1610 ,pdfmenubar / true .code:n =
1611 {
1612   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1613 }
1614 ,pdfmenubar / false .code:n =

```

```

1615     {
1616         \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1617         { HideMenubar } { true }
1618     }
1619 ,pdfmenubar / .code:n =
1620     {
1621         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1622     }
1623 ,pdfmenubar .default:n = true
1624 ,pdfmenubar / unknown .code:n =
1625     {
1626         \msg_warning:nnee { hyp } { no-bool }
1627         { pdfmenubar }
1628         { \exp_not:n {#1} }
1629     }
1630 ,pdfnewwindow .choice:
1631 ,pdfnewwindow / true .code:n =
1632     {
1633         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1634         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1635     }
1636 ,pdfnewwindow / false .code:n =
1637     {
1638         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1639         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1640     }
1641 ,pdfnewwindow / .code:n =
1642     {
1643         \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1644         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1645     }
1646 ,pdfnonfullscreenpagemode .choices:nn =
1647 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1648 {
1649     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1650     { NonFullScreenPageMode } {/#1}
1651 }
1652 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1653 {
1654     \pdf_version_compare:NnTF < {1.6}
1655     {
1656         %message
1657     }
1658     {
1659         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1660         {NonFullScreenPageMode}{/UseAttachments}
1661     }
1662 }
1663 ,pdfnonfullscreenpagemode / .code:n =
1664 {
1665     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1666 }
1667 ,pdfnonfullscreenpagemode / unknown .code:n =
1668 {

```

```

1669     \msg_warning:nneee { hyp } { unknown-choice+empty }
1670     { pdfnonfullscreenpagemode }
1671     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1672     { \exp_not:n {#1} }
1673 }
1674 ,pdfnumcopies .code:n =
1675 {
1676     \pdf_version_compare:NnTF > {1.6}
1677     {
1678         \tl_if_empty:nTF {#1}
1679         {
1680             \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1681         }
1682         {
1683             \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1684             {NumCopies}{#1}
1685         }
1686     }
1687     {
1688         \msg_warning:nnee
1689         {hyp}
1690         {ignore-deprecated-or-unknown-option-in-pdf-version}
1691         {pdfnumcopies}
1692         {\pdf_version:}
1693     }
1694 }
1695 ,pdfpagelayout .choices:nn =
1696 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1697 { \pdfmanagement_add:nne {Catalog} { PageLayout }{ /#1 } }
1698 ,pdfpagelayout / .code:n =
1699 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1700 ,pdfpagelayout / unknown .code:n =
1701 {
1702     \msg_warning:nneee { hyp } { unknown-choice+empty }
1703     { pdfpagelayout }
1704     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1705     { \exp_not:n {#1} }
1706 }
1707 ,pdfpagemode .choices:nn =
1708 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1709 { \pdfmanagement_add:nne {Catalog} { PageMode }{ /#1 } }
1710 ,pdfpagemode / UseAttachments .code:n =
1711 {
1712     \pdf_version_compare:NnTF > {1.5}
1713     {
1714         \pdfmanagement_add:nne {Catalog} { PageMode }{ /UseAttachments }
1715     }
1716     {
1717         \msg_warning:nnee
1718         {hyp}
1719         {ignore-deprecated-or-unknown-value-in-pdf-version}
1720         {UseAttachments}
1721         {\pdf_version:}
1722     }

```

```

1723     }
1724 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1725 ,pdfpagemode / unknown .code:n =
1726 {
1727     \msg_warning:nneee { hyp } { unknown-choice+empty }
1728     { pdfpagemode }
1729     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1730     { \exp_not:n {#1} }
1731 }
1732 ,pdfpagescrop .code:n =
1733 {
1734     \tl_if_empty:nTF {#1} %or blank?
1735     {
1736         \pdfmanagement_remove:nn {Pages} { CropBox }
1737     }
1738     {
1739         \pdfmanagement_add:nne {Pages} { CropBox } { [#1] }
1740     }
1741 }
1742 ,pdfpicktraybypdfsize .choice:
1743 ,pdfpicktraybypdfsize / true .code:n =
1744 {
1745     \pdf_version_compare:NnTF > {1.6}
1746     {
1747         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1748         { PickTrayByPDFSize } { true }
1749     }
1750     {
1751         \msg_warning:nnee
1752         {hyp}
1753         {ignore-deprecated-or-unknown-option-in-pdf-version}
1754         {pdfpicktraybypdfsize}
1755         {\pdf_version:}
1756     }
1757 }
1758 ,pdfpicktraybypdfsize / false .code:n =
1759 {
1760     \pdf_version_compare:NnTF > {1.6}
1761     {
1762         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1763         { PickTrayByPDFSize } { false }
1764     }
1765     {
1766         \msg_warning:nnee
1767         {hyp}
1768         {ignore-deprecated-or-unknown-option-in-pdf-version}
1769         {pdfpicktraybypdfsize}
1770         {\pdf_version:}
1771     }
1772 }
1773 ,pdfpicktraybypdfsize / .code:n =
1774 {
1775     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1776 }

```



```

1777 ,pdfpicktraybypdfsize / unknown .code:n =
1778 {
1779     \msg_warning:nnee { hyp } { no-bool }
1780     { picktraybypdfsize }
1781     { \exp_not:n {#1} }
1782 }
1783 ,pdfprintarea .choices:nn =
1784 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1785 {
1786     \pdf_version_compare:NnTF < {2.0}
1787     {
1788         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1789         { PrintArea } { /#1 }
1790     }
1791     {
1792         \msg_warning:nnee
1793         {hyp}
1794         {ignore-deprecated-or-unknown-option-in-pdf-version}
1795         {pdfprintarea}
1796         {\pdf_version:}
1797     }
1798 }%
1799 ,pdfprintarea / .code:n =
1800 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1801 ,pdfprintarea / unknown .code:n =
1802 {
1803     \msg_warning:nneee { hyp } { unknown-choice+empty }
1804     { pdfprintarea }
1805     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1806     { \exp_not:n {#1} }
1807 }
1808 ,pdfprintclip .choices:nn =
1809 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1810 {
1811     \pdf_version_compare:NnTF < {2.0}
1812     {
1813         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1814         { PrintClip } { /#1 }
1815     }
1816     {
1817         \msg_warning:nnee
1818         {hyp}
1819         {ignore-deprecated-or-unknown-option-in-pdf-version}
1820         {pdfprintclip}
1821         {\pdf_version:}
1822     }
1823 }%
1824 ,pdfprintclip / .code:n =
1825 {
1826     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1827 }
1828 ,pdfprintclip / unknown .code:n =
1829 {
1830     \msg_warning:nneee

```

```

1831     { hyp }
1832     { unknown-choice+empty }
1833     { pdfprintclip }
1834     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1835     { \exp_not:n {#1} }
1836   }
1837 ,pdfprintpagerange .code:n =
1838   {
1839     \pdf_version_compare:NnTF > {1.6}
1840     {
1841       \tl_if_empty:nTF { #1}
1842       {
1843         \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1844         { PrintPageRange }
1845       }
1846       {
1847         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1848         {PrintPageRange}{[#1]}
1849       }
1850     }
1851     {
1852       \msg_warning:nnee
1853       {hyp}
1854       {ignore-deprecated-or-unknown-option-in-pdf-version}
1855       {pdfprintpagerange}
1856       {\pdf_version:}
1857     }
1858   }
1859 ,pdfprintscaling .choices:nn =
1860   { None, AppDefault }
1861   {
1862     \pdf_version_compare:NnTF > {1.5}
1863     {
1864       \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1865       { PrintScaling } { /#1 }
1866     }
1867     {
1868       \msg_warning:nnee
1869       {hyp}
1870       {ignore-deprecated-or-unknown-option-in-pdf-version}
1871       {pdfprintscaling}
1872       {\pdf_version:}
1873     }
1874   }%
1875 ,pdfprintscaling / .code:n =
1876   {
1877     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1878   }
1879 ,pdfprintscaling / unknown .code:n =
1880   {
1881     \msg_warning:nneee { hyp } { unknown-choice+empty }
1882     { pdfprintarea }
1883     { None, AppDefault }
1884     { \exp_not:n {#1} }

```

```

1885     }
1886 ,pdfremotestartview .code:n =
1887 {
1888     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
1889     \exp_args:NNV
1890     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1891     {
1892         \tl_set:Ne \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1893     }
1894     {
1895         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1896         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1897     }
1898 }
1899 ,pdfremotestartview .initial:n = {Fit}
1900 % pdfstartpage is special as it shares code with pdfstartview
1901 ,pdfstartpage .code:n =
1902 {
1903     \tl_gset:Ne \g__hyp_dest_pdfstartpage_tl { #1 }
1904     \bool_if:nTF
1905     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1906     {
1907         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1908     }
1909     {
1910         \pdfmanagement_add:nne {Catalog} { OpenAction }
1911         {
1912             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1913         }
1914     }
1915 }
1916 ,pdfstartpage .initial:n =1
1917 ,pdfstartview .code:n =
1918 {
1919     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
1920     \exp_args:NNV
1921     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1922     {
1923         \tl_gset:Ne \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1924     }
1925     {
1926         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1927         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1928     }
1929     \bool_if:nTF
1930     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1931     {
1932         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1933     }
1934     {
1935         \pdfmanagement_add:nne {Catalog} { OpenAction }
1936         {
1937             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1938         }

```

```

1939     }
1940 }
1941 ,pdfstartview .initial:n = Fit
1942 ,pdftoolbar .choice:
1943 ,pdftoolbar / true .code:n =
1944 {
1945     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1946 }
1947 ,pdftoolbar / false .code:n =
1948 {
1949     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1950     { HideToolbar } { true }
1951 }
1952 ,pdftoolbar / true .code:n =
1953 {
1954     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1955 }
1956 ,pdftoolbar .default:n = true
1957 ,pdftoolbar / unknown .code:n =
1958 {
1959     \msg_warning:nnee { hyp } { no-bool }
1960     { pdftoolbar }
1961     { \exp_not:n {#1} }
1962 }
1963 % pdfview see below.
1964 ,pdfviewarea .choices:nn =
1965 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1966 {
1967     \pdf_version_compare:NnTF < {2.0}
1968     {
1969         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1970         { ViewArea } { /#1 }
1971     }
1972     {
1973         \msg_warning:nnee
1974         {hyp}
1975         {ignore-deprecated-or-unknown-option-in-pdf-version}
1976         {pdfviewarea}
1977         {\pdf_version:}
1978     }
1979 }%
1980 ,pdfviewarea / .code:n =
1981 {
1982     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1983 }
1984 ,pdfviewarea / unknown .code:n =
1985 {
1986     \msg_warning:nnee { hyp } { unknown-choice+empty }
1987     { pdfviewarea }
1988     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1989     { \exp_not:n {#1} }
1990 }
1991 ,pdfviewclip .choices:nn =
1992 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }

```

```

1993 {
1994   \pdf_version_compare:NnTF < {2.0}
1995   {
1996     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1997     { ViewClip } { /#1 }
1998   }
1999   {
2000     \msg_warning:nnee
2001     {hyp}
2002     {ignore-deprecated-or-unknown-option-in-pdf-version}
2003     {pdfviewclip}
2004     {\pdf_version:}
2005   }
2006 }%
2007 ,pdfviewclip / .code:n =
2008 {
2009   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
2010 }
2011 ,pdfviewclip / unknown .code:n =
2012 {
2013   \msg_warning:nneee { hyp } { unknown-choice+empty }
2014   { pdfviewclip }
2015   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2016   { \exp_not:n {#1} }
2017 }
2018 ,pdfwindowui .choice:
2019 ,pdfwindowui / true .code:n =
2020 {
2021   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2022 }
2023 ,pdfwindowui / false .code:n =
2024 {
2025   \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2026   { HideWindowUI } { true }
2027 }
2028 ,pdfwindowui / .code:n =
2029 {
2030   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2031 }
2032 ,pdfwindowui / unknown .code:n =
2033 {
2034   \msg_warning:nnee { hyp } { no-bool }
2035   { pdfwindowui }
2036   { \exp_not:n {#1} }
2037 }
2038 ,pdfwindowui .default:n = true
2039 }

```

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

```

2040 \keys_define:nn { hyp / setup }
2041 {
2042   ,pdfview .code:n =
2043   {
2044     \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}

```

```

2045 \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
2046 {
2047   { xyz }
2048   {
2049     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
2050     {
2051       \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2052       \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2053       {
2054         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2055       }
2056       {
2057         \tl_set:Ne \l__hyp_dest_pdfview_tl
2058         {
2059           \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2060         }
2061       }
2062     }
2063     {
2064       \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2065     }
2066   }
2067   { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2068   { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2069   { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2070   { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2071   { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2072   { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2073   { fitr }
2074   {
2075     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2076     {
2077       \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2078     }
2079     {
2080       %ensure 4 values ...
2081       \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2082       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2083       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2084       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2085       \hbox_set_to_wd:Nnn \l__hyp_dest_box
2086       {
2087         \fp_eval:n
2088         {
2089           round
2090           (
2091             abs
2092             (
2093               \seq_item:Nn\l__hyp_tmpa_seq{4}
2094               -
2095               (\seq_item:Nn\l__hyp_tmpa_seq{2})
2096             ),
2097             3
2098           )

```

```

2099         }bp
2100     }{}
2101     \box_set_dp:Nn \l__hyp_dest_box
2102     {
2103         \fp_eval:n
2104         {
2105             round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2106         }bp
2107     }
2108     \box_set_ht:Nn \l__hyp_dest_box
2109     {
2110         \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2111     }
2112 }
2113 }
2114 }
2115 {
2116     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2117     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2118 }
2119 }
2120 ,pdfview .initial:n = {xyz}
2121 }

```

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.

```

2122 \keys_define:nn { hyp / setup }
2123 {
2124     ,pdflang .code:n =
2125     {
2126         \tl_if_empty:nF { #1 }
2127         {
2128             \pdfmanagement_add:nne {Catalog} { Lang } { (#1) }
2129             \AddToDocumentProperties[document]{lang}{#1}
2130         }
2131     }
2132 }

```

12.3.1 “info keys”

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

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

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

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

pdfproducer (*setup key*)

pdfkeywords (*setup key*)

```

2133 \regex_new:N\l__hyp_optlang_regex
2134 \regex_set:Nn\l__hyp_optlang_regex {\A\([([A-Za-z\-\-])+\)\(.*)}
2135 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2136 {
2137     \keys_define:nn { hyp / setup }

```

```

2138 {
2139     pdf#1 .code:n =
2140     {
2141         \tl_if_blank:nTF {##1}
2142         {
2143             \str_case:nnF { #1 }
2144             {
2145                 {creator}
2146                 {
2147                     \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2148                     \pdfmanagement_add:nne {Info}{Creator}{()}
2149                 }
2150                 {producer}
2151                 {
2152                     \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2153                     \pdfmanagement_add:nne {Info}{Producer}{()}
2154                 }
2155             }
2156             {
2157                 \pdfmanagement_remove:nn {Info}{#2}
2158             }
2159         }
2160         {
2161             \tl_set:Ne\l__hyp_tmpa_tl {\clist_item:nn{##1}{1}}
2162             \exp_args:NNV
2163             \regex_extract_once:NnN \l__hyp_optlang_regex \l__hyp_tmpa_tl\l__hyp_tmpa_s
2164             \seq_if_empty:NTF\l__hyp_tmpa_seq
2165             {
2166                 \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2167             }
2168             {
2169                 \exp_args:Ne
2170                 \__hyp_text_pdfstring_info:nN {\seq_item:Nn \l__hyp_tmpa_seq{3}}\l__hyp_t
2171             }
2172             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2173             {
2174                 \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2175             }
2176         }
2177         \__hyp_store_metadata:nn {pdf#1}{##1}
2178     }
2179 }
2180 \keys_define:nn { hyp / info }
2181 {
2182     #2 .code:n =
2183     {
2184         \tl_if_blank:nTF {##1}
2185         {
2186             \pdfmanagement_remove:nn {Info}{#2}
2187         }
2188         {
2189             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2190             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2191             {

```



```

2192         \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2193     }
2194 }
2195 \exp_args:Nne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2196 }
2197 ,unknown .code:n =
2198 {
2199     \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2200     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2201     {
2202         \exp_args:Nno
2203         \pdfmanagement_add:nne {Info}
2204         { \l_keys_key_str } { \l__hyp_tmpa_str}
2205     }
2206 }
2207 }
2208 }
2209 \__hyp_setup_info_key:nn {author} {Author}
2210 \__hyp_setup_info_key:nn {title} {Title}
2211 \__hyp_setup_info_key:nn {producer} {Producer}
2212 \__hyp_setup_info_key:nn {creator} {Creator}
2213 % ignored key: addtopdfcreator
2214 \__hyp_setup_info_key:nn {subject} {Subject}
2215 \__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) 2216 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2217 {
2218     \keys_define:nn { hyp / setup }
2219     {
2220         pdf#1 .code:n =
2221         {
2222             \tl_if_blank:nTF {##1}
2223             {
2224                 \pdfmanagement_remove:nn {Info}{#2}
2225             }
2226             {
2227                 \pdfmanagement_add:nne {Info}{#2}{(##1)}
2228             }
2229             \__hyp_store_metadata:nn {pdf#1}{##1}
2230             \AddToDocumentProperties[document]{#1}{##1}
2231         }
2232     }
2233     \keys_define:nn { hyp / info }
2234     {
2235         #2 .code:n =
2236         {
2237             \tl_if_blank:nTF {##1}
2238             {
2239                 \pdfmanagement_remove:nn {Info}{#2}
2240             }
2241             {
2242                 \pdfmanagement_add:nne {Info}{#2}{(##1)}

```

```

2243     }
2244     \exp_args:Ne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2245   }
2246 }
2247 }
2248
2249 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2250 \__hyp_setup_info_date_key:nn {moddate} {ModDate}
2251 \keys_define:nn { hyp / setup }
2252 {
2253   pdfmetadate .code:n = { \__hyp_store_metadata:nn {pdfmetadate}{#1} }
2254 }

```

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

```

2255 \keys_define:nn { hyp / setup }
2256 {
2257   ,pdftrapped .code:n =
2258   {
2259     \exp_args:Nne
2260     \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2261   }
2262   ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2263   {
2264     \pdfmanagement_add:nne {Info}{Trapped}
2265     {/
2266       \str_uppercase:f { \str_head:n { #1 } }
2267       \str_lowercase:f { \str_tail:n { #1 } }
2268     }
2269     \__hyp_store_metadata:ne {pdftrapped}
2270     {
2271       \str_uppercase:f { \str_head:n { #1 } }
2272       \str_lowercase:f { \str_tail:n { #1 } }
2273     }
2274   }
2275   ,_pdftrapped / unknown .code:n =
2276   {
2277     \msg_warning:nneee { hyp } { unknown-choice }
2278     { pdftrapped }
2279     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2280     { \exp_not:n {#1} }
2281   }
2282 }

```

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

```

2283 \keys_define:nn { hyp / setup }
2284 {
2285   pdfinfo .code:n =
2286   {
2287     \keys_set:nn { hyp / info } { #1 }
2288   }
2289 }

```

Now we set some default values

```

2290 \keys_set:nn { hyp / setup } {pdfcreator = LaTeX-with-hyperref}

```

```

2291 \keys_set:nn { hyp / setup} {pdfauthor = }
2292 \keys_set:nn { hyp / setup} {pdftitle = }
2293 \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:

```

2294 \clist_map_inline:nn
2295 {
2296   ,pdfcopyright
2297   ,pdftype
2298   ,pdflicenseurl
2299   ,pdfauthortitle
2300   ,pdfcaptionwriter
2301   ,pdfmetalang
2302   ,pdfsource
2303   ,pdfdocumentid
2304   ,pdfinstanceid
2305   ,pdfversionid
2306   ,pdfrendition
2307   ,pdfpublication
2308   ,pdfpubtype
2309   ,pdfbytes
2310   ,pdfnumpages
2311   ,pdfissn
2312   ,pdfeissn
2313   ,pdfisbn
2314   ,pdfbookedition
2315   ,pdfpublisher
2316   ,pdfvolumenum
2317   ,pdfissuenum
2318   ,pdfpagerange
2319   ,pdfdoi
2320   ,pdfurl
2321   ,pdfidentifier
2322   ,pdfsubtitle
2323   ,pdfpubstatus
2324   ,pdfcontactaddress
2325   ,pdfcontactcity
2326   ,pdfcontactregion
2327   ,pdfcontactpostcode
2328   ,pdfcontactcountry
2329   ,pdfcontactphone
2330   ,pdfcontactemail
2331   ,pdfcontacturl
2332   ,pdfdate
2333 }
2334 {
2335   \keys_define:nn { hyp / setup }
2336   {
2337     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2338   }

```

```

2339 }
2340

```

12.5 Transitions

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

```

2341 \keys_define:nn { hyp / setup }
2342 {
2343   pdfpageduration .code:n =
2344   {
2345     \tl_if_blank:nTF { #1 }
2346     {
2347       \pdfmanagement_remove:nn {Page}{Dur}
2348     }
2349     {
2350       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2351     }
2352   }
2353 }

```

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

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

The keys allowed in the argument of `pdfpagetransition` are

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

```

2354 \keys_define:nn { hyp / setup }
2355 {
2356   pdfpagetransition .code:n =
2357   {
2358     \tl_if_blank:nTF {#1}

```

```

2359     {
2360         \pdfmanagement_remove:nn {Page}{Trans}
2361     }
2362     {
2363         \group_begin:
2364         \keys_set:nn { hyp / trans }{style=R,#1}
2365         \pdf_object_unnamed_write:ne { dict }
2366         {
2367             \pdfdict_use:n {l__hyp_page/Trans}
2368         }
2369         \pdfmanagement_add:nne {Page}{Trans}{\pdf_object_ref_last:}
2370         \group_end:
2371     }
2372 }
2373 }
2374 \keys_define:nn { hyp / trans }
2375 {
2376     ,style .choices:nn =
2377     {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2378     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2379     ,style / unknown .code:n =
2380     {
2381         \msg_warning:nneee { hyp } { unknown-choice }
2382         { trans / style }
2383         { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2384         { \exp_not:n {#1} }
2385     }
2386     ,duration .code:n =
2387     {
2388         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{#1}
2389     }
2390     ,direction .choices:nn =
2391     {H,V}
2392     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2393     ,direction .choices:nn =
2394     {0,90,180,270,315}
2395     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2396     ,direction / None .code:n =
2397     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2398     ,direction / unknown .code:n =
2399     {
2400         \msg_warning:nneee { hyp } { unknown-choice }
2401         { trans / direction }
2402         {
2403             H~(horizontal,~only~Split,~Blinds),
2404             V~(vertical,~only~Split,~Blinds),
2405             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2406             90~(bottom~to~top,~only~Wipe),
2407             180~(right~to~left,~only~Wipe),
2408             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2409             315~(top~left~to~bottom,~only~Glitter),
2410             None~(only~Fly)
2411         }
2412         { \exp_not:n {#1} }

```

```

2413     }
2414     ,motion .choices:nn =
2415     {I,0}
2416     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2417     ,motion / unknown .code:n =
2418     {
2419         \msg_warning:nneee { hyp } { unknown-choice }
2420         { trans / motion }
2421         { I~(inwards) , 0~(outwards) }
2422         { \exp_not:n {#1} }
2423     }
2424     ,scale .code:n =
2425     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2426     ,opaque .choices:nn = {true,false}
2427     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2428     ,opaque / unknown .code:n =
2429     {
2430         \msg_warning:nneee { hyp } { unknown-choice }
2431         { trans / B }
2432         { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2433         { \exp_not:n {#1} }
2434     }
2435     % try to set unknown keys as style
2436     ,unknown .code:n =
2437     {
2438         % warning ...
2439         \exp_args:Nne\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2440     }
2441 }

```

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

```

2442 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}

```

Unfinished Form field code

```

2443 \NewDocumentCommand \MakeFieldObject { m m }
2444 {
2445     \pdfxform_new:nnn { #2 }{ } { #1 }
2446 }
2447
2448
2449 \prop_new:N    \g__hyp_AcroForm_CoFields_prop
2450 \prop_new:N    \g__hyp_AcroForm_Fields_prop
2451
2452 \let\HyField@afields\ltx@empty
2453 \let\HyField@cofields\ltx@empty
2454 %% UF test for old pdftex removed
2455 %\let\HyField@AuxAddToFields\ltx@gobble
2456 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2457 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2458
2459 % the value doesn't matter, but with a prop we avoid duplicates and it is
2460 % clearly faster than removing them from a sequence
2461 \def\HyField@AuxAddToFields#1
2462 {
2463     \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}

```

```

2464 }%
2465
2466 %fields with empty key get a value too -- lets hope that
2467 %this give the expected behaviour
2468 \def\HyField@AuxAddToCoFields #1 #2
2469 {
2470   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2471 }
2472
2473 \Hy@AtBeginDocument
2474 {
2475   \if@files
2476     \immediate\write\@mainaux{%
2477       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2478     }%
2479     \immediate\write\@mainaux{%
2480       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2481     }%
2482   \fi
2483   \let\HyField@AfterAuxOpen\@firstofone
2484 }%
2485
2486 \def\HyField@AddToFields
2487 {
2488   \exp_args:Ne\HyField__hypAddToFields
2489   {
2490     \pdfannot_box_ref_last:
2491   }
2492   \ifx\Fld@calculate@code\ltx@empty
2493   \else
2494     \begingroup
2495     \Hy@safe@activetrue
2496     \edef\Hy@temp{%
2497       \endgroup
2498       \if@files
2499         \write\@mainaux
2500         {
2501           \string\HyField@AuxAddToCoFields
2502           {
2503             \Fld@calculate@sortkey
2504           }
2505           {
2506             \pdfannot_box_ref_last:
2507           }
2508         }
2509       \fi
2510     }%
2511     \Hy@temp
2512   \fi
2513 }%
2514
2515 \def\HyField__hypAddToFields#1{
2516   \HyField@AfterAuxOpen{%
2517     \if@files

```

```

2518     \write\@mainaux{%
2519         \string\HyField@AuxAddToFields{#1}%
2520     }%
2521     \fi
2522 }%
2523 }%
2524
2525 \ExplSyntaxOff
2526 \ExplSyntaxOn
2527
2528 \def\@Form[#1]
2529 {
2530     \kvsetkeys{Form}{#1}
2531     \pdf@ifdraftmode{}
2532     {
2533         \Hy@FormObjects
2534         \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2535         {
2536             \pdfmanagement_add:nne { Catalog / AcroForm } { Fields }{##1}
2537             %\pdfmanagement_show:n { Catalog / AcroForm }
2538         }
2539         \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2540         {
2541             \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2542             {
2543                 \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2544             }
2545             \seq_sort:Nn \l__hyp_tmpa_seq
2546             {
2547                 \str_compare:nNnTF {##1} > {##2}
2548                 { \sort_return_swapped: }
2549                 { \sort_return_same: }
2550             }
2551             \seq_map_inline:Nn \l__hyp_tmpa_seq
2552             {
2553                 \pdfmanagement_add:nne { Catalog / AcroForm }
2554                 { CO }
2555                 {
2556                     \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2557                 }
2558             }
2559         }
2560         \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2561         {ZaDb} {\pdf_object_ref:n {\__hyp/Font/ZaDb} }
2562         \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2563         {Helv} {\pdf_object_ref:n {\__hyp/Font/Helv} }
2564         \pdfmanagement_add:nne {Catalog /AcroForm}
2565         {DA}{(/Helv~10~Tf~0~g)}
2566         \pdfmeta_standard_verify:nTF {form_no_NeedAppearance}
2567         {
2568             \legacy_if:nT { HyField@NeedAppearances }
2569             {
2570                 \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2571             }

```



```

2572     }
2573     {
2574         \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2575     }
2576 }
2577 }
2578 \ExplSyntaxOff
2579 \let\@endForm\ltx@empty
2580 \let\HyAnn@AbsPageLabel\ltx@empty
2581 \let\Fld@pageobjref\ltx@empty
2582
2583 \ExplSyntaxOn
2584 \newcount\HyAnn@Count
2585 \HyAnn@Count=\ltx@zero
2586 \def\HyAnn@AbsPageLabel
2587 {
2588     \global\advance\HyAnn@Count by\ltx@one
2589     %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2590     %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2591     %\zref@refused{HyAnn@\the\HyAnn@Count}%
2592     \__hyp_property_record:ee {HyAnn@\the\HyAnn@Count}{abspage}
2593     \__hyp_property_ref_undefined_warn:ee {HyAnn@\the\HyAnn@Count}{abspage}
2594 }%
2595 \def\Fld@pageobjref
2596 {
2597     \__hyp_property_if_recorded:eeT {HyAnn@\the\HyAnn@Count}{abspage}
2598     {
2599         /P~\pdf_pageobject_ref:e
2600         {
2601             \__hyp_property_ref:ee{HyAnn@\the\HyAnn@Count}{abspage}
2602         }
2603     }
2604 }
2605 \ExplSyntaxOff
2606 \ExplSyntaxOn
2607 %% check if the attr should be set through
2608 %% hooks.
2609 %% check if options are missing.
2610 \def\@TextField[#1]#2{% parameters, label
2611     \def\Fld@name{#2}%
2612     \let\Fld@default\ltx@empty
2613     \let\Fld@value\@empty
2614     \def\Fld@width{\DefaultWidthofText}%
2615     \def\Fld@height{%
2616         \ifFld@multiline
2617             \DefaultHeightofTextMultiline
2618         \else
2619             \DefaultHeightofText
2620         \fi
2621     }%
2622     \begingroup
2623         \expandafter\HyField@SetKeys\expandafter{%
2624             \DefaultOptionsofText,#1%
2625         }%

```

```

2626 \PDFForm@Name
2627 \HyField@FlagsText
2628 \ifFld@hidden\def\Fld@width{1sp}\fi
2629 \ifx\Fld@value\empty\def\Fld@value{\Fld@default}\fi
2630 \LayoutTextField{#2}{%
2631   \leavevmode
2632   \HyAnn@AbsPageLabel
2633   \Hy@escapeform\PDFForm@Text
2634   \pdfannot_box:nnnn
2635   {\Fld@width}
2636   {\Fld@height}
2637   {Opt} %is this correct?
2638   {\PDFForm@Text}
2639   \MakeTextField{\Fld@width}{\Fld@height}
2640   \HyField@AddToFields
2641 }%
2642 \endgroup
2643 }
2644 \providecommand\@curropt{}
2645 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2646   \def\Fld@name{#2}
2647   \let\Fld@default\relax
2648   \let\Fld@value\relax
2649   \def\Fld@width{\DefaultWidthofChoiceMenu}
2650   \def\Fld@height{\DefaultHeightofChoiceMenu}
2651   \begingroup
2652     \Fld@menulength=0 %
2653     \@tempdima\z@
2654     \clist_map_variable:nNn { #3 } \@curropt
2655     %\@for\@curropt:=#3\do
2656     {%
2657       \expandafter\Fld@checkequals\@curropt==\%
2658       \Hy@StepCount\Fld@menulength
2659       \settowidth{\@tempdimb}{\@currDisplay}%
2660       \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2661     }%
2662     \advance\@tempdima by~15\p@
2663     \begingroup
2664       \HyField@SetKeys{#1}
2665     \edef\x{\endgroup
2666       \noexpand\expandafter
2667       \noexpand\HyField@SetKeys
2668       \noexpand\expandafter{%
2669         \expandafter\noexpand\csname DefaultOptionsof%
2670         \ifFld@radio
2671           Radio%
2672         \else
2673           \ifFld@combo
2674             \ifFld@popdown
2675               PopdownBox%
2676             \else
2677               ComboBox%
2678             \fi
2679             \else

```

```

2680         ListBox%
2681     \fi
2682 \fi
2683 \endcsname
2684 }%
2685 }\x
2686 \HyField@SetKeys{#1}%
2687 \PDFForm@Name
2688 \ifFld@hidden\def\Fld@width{1sp}\fi
2689 \ifx\Fld@value\relax
2690     \let\Fld@value\Fld@default
2691 \fi
2692 \LayoutChoiceField{#2}{%
2693     \ifFld@radio
2694         \HyField@FlagsRadioButton
2695         \__hypRadio{#3}%
2696     \else
2697         \beginngroup
2698         \HyField@FlagsChoice
2699         \ifdim\Fld@width<\@tempdima
2700             \ifdim\@tempdima<1cm\@tempdima1cm\fi
2701             \edef\Fld@width{\the\@tempdima}%
2702         \fi
2703         \ifFld@combo
2704         \else
2705             \@tempdima=\the\Fld@menulength\Fld@charsize
2706             \advance\@tempdima by~\Fld@borderwidth bp %
2707             \advance\@tempdima by~\Fld@borderwidth bp %
2708             \edef\Fld@height{\the\@tempdima}%
2709         \fi
2710         \__hypListbox{#3}%
2711         \endgroup
2712     \fi
2713 }%
2714 \endgroup
2715 }
2716
2717 \def\__hypRadio#1{%
2718     \Fld@listcount=0-%
2719     %\show\Fld@default
2720     \EdefEscapeName\Fld@default{\Fld@default}%
2721     \clist_map_variable:nNn { #1 } \@curropt
2722     %\@for\@curropt:=#1\do
2723     {%
2724         \expandafter\Fld@checkequals\@curropt==\\%
2725         \EdefEscapeName\@currValue{\@currValue}%
2726         \Hy@StepCount\Fld@listcount
2727         \@currDisplay\space
2728         \leavevmode
2729         \HyAnn@AbsPageLabel
2730         \Hy@escapeform\PDFForm@Radio
2731         \pdfxform_if_exist:nF { __hyp_xform_Ding }
2732         {
2733             \pdfxform_new:nnn { __hyp_xform_Ding } {}

```

```

2734     {
2735         \group_begin:
2736         \fontfamily{pzd}
2737         \fontencoding{U}
2738         \fontseries{m}
2739         \fontshape{n}
2740         \selectfont
2741         \char123
2742         \group_end:
2743     }
2744 }
2745 \pdfannot_box:nnne
2746 {\Fld@width}
2747 {\Fld@height}
2748 {Opt} %is this correct?
2749 {
2750     \PDFForm@Radio
2751     /AP
2752     <<
2753     /N
2754     <<
2755     /\@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2756     %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2757     >>
2758     >>
2759 }
2760 {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2761 \int_compare:nNnT { \Fld@listcount} = { 1 }
2762 { \HyField@AddToFields }
2763 \c_space_tl % deliberate space between radio buttons
2764           % to do: --> should be configurable
2765 }%
2766 }
2767
2768 \newcount\Fld@listcount
2769 \def\__hypListbox#1
2770 {
2771     \HyField@PDFChoices{#1}
2772     \mode_leave_vertical:
2773     \HyAnn@AbsPageLabel
2774     \Hy@escapeform\PDFForm@List
2775     \pdf_link_user:nnn
2776     {widget} %perhaps we need more types??
2777     {\PDFForm@List}
2778     {\MakeChoiceField{\Fld@width}{\Fld@height}}
2779     \HyField@AddToFields
2780 }
2781
2782
2783 \def\@PushButton[#1]#2{% parameters, label
2784     \def\Fld@name{#2}%
2785     \group_begin:
2786     \exp_args:No\HyField@SetKeys
2787     {

```

```

2788     \DefaultOptionsofPushButton,#1
2789 }
2790 \PDFForm@Name
2791 \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2792 {
2793     \HyField@FlagsPushButton
2794     \legacy_if:nT {Fld@hidden}
2795     {
2796         \def\Fld@width{1sp}
2797     }
2798     \LayoutPushButtonField
2799     {
2800         \mode_leave_vertical:
2801         \HyAnn@AbsPageLabel
2802         \Hy@escapeform\PDFForm@Push
2803         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2804         \pdfannot_box:nnnn
2805         {\box_wd:N\l_tmpa_box}
2806         {\box_ht:N\l_tmpa_box}
2807         {\box_dp:N\l_tmpa_box} %is this correct?
2808         {\PDFForm@Push}
2809         {\box_use:N\l_tmpa_box}
2810         \HyField@AddToFields
2811     }
2812 }
2813 {
2814     \msg_error:nn { hyp }{ pdfa-no-push-button }
2815     \LayoutPushButtonField
2816     {
2817         \mode_leave_vertical:
2818         \MakeButtonField{#2}
2819     }
2820 }
2821 \group_end:
2822 }
2823
2824 \def\@Submit[#1]#2
2825 {
2826     \def\Fld@width {\DefaultWidthofSubmit}
2827     \def\Fld@height{\DefaultHeightofSubmit}
2828     \group_begin:
2829     \exp_args:No\HyField@SetKeys
2830     {
2831         \DefaultOptionsofSubmit,#1
2832     }
2833     \HyField@FlagsPushButton
2834     \HyField@FlagsSubmit
2835     \legacy_if:nT { Fld@hidden }
2836     {
2837         \def\Fld@width{1sp}
2838     }
2839     \mode_leave_vertical:
2840     \HyAnn@AbsPageLabel
2841     \Hy@escapeform\PDFForm@Submit

```

```

2842 \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2843 \pdfxform_if_exist:nF
2844 { __hyp_xform_Submit }
2845 {
2846   \pdfxform_new:nnn { __hyp_xform_Submit }{}
2847   {
2848     \fbox{\color_select:n{yellow}\textsf{Submit}}
2849   }
2850   \pdfxform_new:nnn { __hyp_xform_SubmitP }{}
2851   {
2852     \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2853   }
2854 }
2855 \pdfannot_box:nnnn
2856 {\box_wd:N\l_tmpa_box}
2857 {\box_ht:N\l_tmpa_box}
2858 {\box_dp:N\l_tmpa_box} %is this correct?
2859 {
2860   \PDFForm@Submit
2861   /AP<<
2862     /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2863     /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2864   >>
2865 }
2866 \HyField@AddToFields
2867 \box_use:N\l_tmpa_box
2868
2869 \group_end:
2870 }
2871
2872 \def\@Reset[#1]#2
2873 {
2874   \def\Fld@width {\DefaultWidthofReset}
2875   \def\Fld@height{\DefaultHeightofReset}
2876   \group_begin:
2877   \exp_args:No\HyField@SetKeys
2878   {
2879     \DefaultOptionsofReset,#1
2880   }
2881   \mode_leave_vertical:
2882   \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2883   {
2884     \HyField@FlagsPushButton
2885     \legacy_if:nT { Fld@hidden }
2886     { \def\Fld@width{isp} }
2887     \HyAnn@AbsPageLabel
2888     \Hy@escapeform\PDFForm@Reset
2889     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2890     \pdfannot_box:nnnn
2891     {\box_wd:N\l_tmpa_box}
2892     {\box_ht:N\l_tmpa_box}
2893     {\box_dp:N\l_tmpa_box} %is this correct?
2894     { \PDFForm@Reset }
2895     \HyField@AddToFields

```

```

2896         \box_use:N \l_tmpa_box
2897     }
2898     {
2899         \msg_error:nn { hyp }{ pdfa-no-reset-button }
2900         \MakeButtonField{#2}
2901     }
2902     \group_end:
2903 }
2904
2905 \def\@CheckBox[#1]#2
2906 {% parameters, label
2907     \def\Fld@name{#2}
2908     \def\Fld@default{0}
2909     \group_begin:
2910         \def\Fld@width {\DefaultWidthofCheckBox}
2911         \def\Fld@height{\DefaultHeightofCheckBox}
2912         \exp_args:No\HyField@SetKeys
2913         {
2914             \DefaultOptionsofCheckBox,#1
2915         }
2916         \PDFForm@Name
2917         \HyField@FlagsCheckBox
2918         \legacy_if:nT { Fld@hidden }
2919         {
2920             \def\Fld@width{1sp}
2921         }
2922         \LayoutCheckField{#2}
2923         {
2924             \mode_leave_vertical:
2925             \HyAnn@AbsPageLabel
2926             \Hy@escapeform\PDFForm@Check
2927             \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
2928             {
2929                 \pdfxform_new:nnn
2930                 {__hyp_xform_CheckMarkYes}{-}
2931                 {
2932                     \group_begin:
2933                     \fontfamily{pzd}
2934                     \fontencoding{U}
2935                     \fontseries{m}
2936                     \fontshape{n}
2937                     \selectfont
2938                     \char51
2939                     \group_end:
2940                 }
2941                 \pdfxform_new:nnn
2942                 {__hyp_xform_CheckMarkOff}{-}
2943                 {
2944                     \group_begin:
2945                     \fontfamily{pzd}
2946                     \fontencoding{U}
2947                     \fontseries{m}
2948                     \fontshape{n}
2949                     \selectfont

```

```

2950         \phantom{\char51} %perhaps xetex needs some small glyph ..
2951     \group_end:
2952 }
2953 }
2954 \pdfannot_box:nnnn
2955     {\Fld@width}
2956     {\Fld@height}
2957     {Opt} %is this correct?
2958     {\PDFForm@Check}
2959     \HyField@AddToFields %check if this works with xelatex ...
2960 }
2961 \group_end:
2962 }
2963 \ExplSyntaxOff
2964
2965 %hm. Should a luatex driver use type1 fonts in fields????
2966 \ExplSyntaxOn
2967 \def\Hy@FormObjects
2968 {
2969     \pdf_object_new:n    {__hyp/Encoding/pdfdoc }
2970     \pdf_object_new:n    {__hyp/Font/ZaDb }
2971     \pdf_object_new:n    {__hyp/Font/Helv }
2972     \pdf_object_write:nne {__hyp/Encoding/pdfdoc } { dict }
2973     {
2974         /Type/Encoding
2975         /Differences[
2976             24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2977             /ring/tilde
2978             \c_space_tl
2979             39/quotesingle
2980             \c_space_tl
2981             96/grave %
2982             \iow_newline:
2983             128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2984             /fraction/guilsinglleft/guilsinglright/minus/perthousand
2985             /quotedblbase/quotedblleft/quotedblright/quoteleft
2986             /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2987             /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2988             \iow_newline:
2989             164/currency
2990             \c_space_tl
2991             166/brokenbar
2992             \c_space_tl
2993             168/dieresis/copyright/ordfeminine
2994             \c_space_tl
2995             172/logicalnot/.notdef/registered/macron/degree/plusminus
2996             /twosuperior/threesuperior/acute/mu
2997             \c_space_tl
2998             183/periodcentered/cedilla/onesuperior/ordmasculine
2999             \c_space_tl
3000             188/onequarter/onehalf/threequarters
3001             \iow_newline:
3002             192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
3003             /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave

```



```

3004         /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
3005         /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
3006         /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
3007         /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
3008         /cedilla/egrave/eacute/ecircumflex/edieresis/igrave
3009         /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
3010         /ocircumflex/otilde/odieresis/divide/oslash/ugrave
3011         /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
3012     ]
3013 }
3014 \pdf_object_write:nnn {__hyp/Font/ZaDb } { dict }
3015 {
3016     /Type/Font
3017     /Subtype/Type1
3018     /Name/ZaDb
3019     /BaseFont/ZapfDingbats
3020 }
3021 \pdf_object_write:nne {__hyp/Font/Helv } { dict }
3022 {
3023     /Type/Font
3024     /Subtype/Type1
3025     /Name/Helv
3026     /BaseFont/Helvetica
3027     /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
3028 }
3029 \global\let\Hy@FormObjects\relax
3030 }
3031 \ExplSyntaxOff
3032 \providecommand*{\Fld@pageobjref}{}
3033 \ifcsname pdf@escapestring\endcsname
3034   \def\Hy@escapeform#1{%
3035     \ifHy@pdfescapeform
3036       \let\Hy@escapestring\pdfescapestring
3037     \else
3038       \let\Hy@escapestring\@firstofone
3039     \fi
3040   }%
3041   \Hy@escapeform{}%
3042 \else
3043   \let\Hy@escapestring\@firstofone
3044   \def\Hy@escapeform#1{%
3045     \ifHy@pdfescapeform
3046       \def\Hy@escapestring##1{%
3047         \noexpand\Hy@escapestring{\noexpand##1}%
3048       }%
3049       \edef\Hy@temp{#1}%
3050       \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{}\@nil
3051       \def\Hy@escapestring##1{%
3052         \@ifundefined{Hy@esc@\string##1}{%
3053           ##1%
3054           \ThisShouldNotHappen
3055         }{%
3056           \csname Hy@esc@\string##1\endcsname
3057         }%

```

```

3058     }%
3059     \else
3060         \let\Hy@escapestring\@firstofone
3061     \fi
3062 }%
3063 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3064     \ifx\#3\%
3065     \else
3066         \expandafter
3067         \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3068         \ltx@ReturnAfterFi{%
3069             \Hy__hypescapeform#3\@nil
3070         }%
3071     \fi
3072 }%
3073 \fi
3074 \def\PDFForm@Name{%
3075     \PDFForm__hypName\Fld@name
3076     \ifx\Fld@altname\relax
3077     \else
3078         \PDFForm__hypName\Fld@altname
3079     \fi
3080     \ifx\Fld@mappingname\relax
3081     \else
3082         \PDFForm__hypName\Fld@mappingname
3083     \fi
3084 }
3085 \def\PDFForm__hypName#1{%
3086     \begingroup
3087         \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3088             \ifHy@unicode
3089                 \Hy@unicodedefalse
3090             \fi
3091         \fi
3092         \pdfstringdef\Hy@gtemp#1%
3093     \endgroup
3094     \let#1\Hy@gtemp
3095 }
3096 \def\Fld@X@additionalactions{%
3097     \ifx\Fld@keystroke@code\@empty
3098     \else
3099         /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3100     \fi
3101     \ifx\Fld@format@code\@empty
3102     \else
3103         /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3104     \fi
3105     \ifx\Fld@validate@code\@empty
3106     \else
3107         /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3108     \fi
3109     \ifx\Fld@calculate@code\@empty
3110     \else
3111         /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%

```

```

3112 \fi
3113 \ifx\Fld@onfocus@code\@empty
3114 \else
3115 /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3116 \fi
3117 \ifx\Fld@onblur@code\@empty
3118 \else
3119 /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3120 \fi
3121 \ifx\Fld@onmousedown@code\@empty
3122 \else
3123 /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3124 \fi
3125 \ifx\Fld@onmouseup@code\@empty
3126 \else
3127 /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3128 \fi
3129 \ifx\Fld@onenter@code\@empty
3130 \else
3131 /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3132 \fi
3133 \ifx\Fld@onexit@code\@empty
3134 \else
3135 /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3136 \fi
3137 }
3138 \ExplSyntaxOn
3139 \def\Fld@additionalactions
3140 {%
3141 \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{}
3142 {
3143 \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3144 {/AA<<\Fld@X@additionalactions>>}
3145 }
3146 }
3147 \ExplSyntaxOff
3148 \def\Fld@annotnames{%
3149 /T(\Fld@name)%
3150 \ifx\Fld@altname\relax
3151 \else
3152 /TU(\Fld@altname)%
3153 \fi
3154 \ifx\Fld@mappingname\relax
3155 \else
3156 /TM(\Fld@mappingname)%
3157 \fi
3158 }
3159 \ExplSyntaxOn
3160 \def\PDFForm@Check
3161 {
3162 /Subtype/Widget
3163 ~\Fld@annotflags
3164 ~\Fld@pageobjref
3165 ~\Fld@annotnames

```

```

3166     /FT/Btn
3167     \Fld@flags
3168     /Q~\Fld@align
3169     /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3170     /AP
3171     <<
3172     /N
3173     <<
3174         /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3175         /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3176     >>
3177     >>
3178     /MK<<
3179         \int_compare:nNnF {\Fld@rotation}={0}
3180         {
3181             /R~\Fld@rotation
3182         }
3183         \tl_if_empty:NF\Fld@bordercolor
3184         {
3185             /BC[\Fld@bordercolor]
3186         }
3187         \tl_if_empty:NF\Fld@bcolor
3188         {
3189             /BG[\Fld@bcolor]
3190         }
3191         /CA(\Hy@escapestring{\Fld@cbsymbol})%
3192     >>
3193     /DA
3194     (
3195         /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3196         \tl_if_empty:NF \Fld@color
3197         {
3198             \c_space_tl \Fld@color
3199         }
3200     )
3201     /H/P
3202     \legacy_if:nTF {\Fld@checked}
3203     {
3204         /V/Yes /AS/Yes
3205     }
3206     {
3207         /V/Off /AS/Off
3208     }
3209     \Fld@additionalactions
3210 }
3211 \ExplSyntaxOff
3212 \ExplSyntaxOn
3213 \def\PDFForm@Push
3214 {
3215     /Subtype/Widget
3216     ~\Fld@annotflags
3217     ~\Fld@pageobjref
3218     ~\Fld@annotnames
3219     /FT/Btn

```

```

3220 ~\Fld@flags
3221 /H/P
3222 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3223 \bool_if:nT
3224 {
3225     !\int_compare_p:nNn {\Fld@rotation} = {0}
3226     ||
3227     \tl_if_exist_p:N \Fld@bordercolor
3228 }
3229 {
3230     /MK
3231     <<
3232     \int_compare:nNnF {\Fld@rotation} = {0}
3233     {
3234         /R~\Fld@rotation
3235     }
3236     \tl_if_exist:NT \Fld@bordercolor
3237     {
3238         /BC[\Fld@bordercolor]
3239     }
3240     >>
3241 }
3242 /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3243 \Fld@additionalactions
3244 }
3245
3246 \ExplSyntaxOff
3247 \def\PDFForm@List{%
3248     /Subtype/Widget%
3249     \Fld@annotflags
3250     \Fld@pageobjref
3251     \Fld@annotnames
3252     /FT/Ch%
3253     \Fld@flags
3254     /Q \Fld@align
3255     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3256     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3257         \ifx\Fld@bordercolor\relax\else 1\fi
3258         \ifx\Fld@bcolor\relax \else 1\fi
3259     \space
3260 \else
3261     /MK<<%
3262     \ifnum\Fld@rotation=\z@
3263     \else
3264         /R \Fld@rotation
3265     \fi
3266     \ifx\Fld@bordercolor\relax
3267     \else
3268         /BC[\Fld@bordercolor]%
3269     \fi
3270     \ifx\Fld@bcolor\relax
3271     \else
3272         /BG[\Fld@bcolor]%
3273     \fi

```

```

3274     >>%
3275     \fi
3276     /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3277         \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3278     \Fld@choices
3279     \Fld@additionalactions
3280 }
3281 \ExplSyntaxOn
3282 \def\PDFForm@Radio
3283 {
3284     /Subtype/Widget
3285     ~\Fld@annotflags
3286     ~\Fld@pageobjref
3287     ~\Fld@annotnames
3288     /FT/Btn
3289     \Fld@flags
3290     /H/P
3291     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3292     /MK<<
3293     \ifnum\Fld@rotation=\z@
3294     \else
3295         /R~\Fld@rotation
3296     \fi
3297     \ifx\Fld@bordercolor\relax
3298     \else
3299         /BC[\Fld@bordercolor]%
3300     \fi
3301     \ifx\Fld@bcolor\relax
3302     \else
3303         /BG[\Fld@bcolor]%
3304     \fi
3305     /CA(\Hy@escapestring{\Fld@radiosymbol})%
3306     >>
3307     /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3308         \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3309     \ifx\Fld@default\@empty
3310         /V/Off%
3311         /DV/Off%
3312     \else
3313         /V/\Fld@default
3314         /DV/\Fld@default
3315     \fi
3316     \Fld@additionalactions
3317 }
3318 \ExplSyntaxOff
3319 \ExplSyntaxOn
3320 % Does an appeareance dict make sense here?
3321 \def\PDFForm@Text
3322 {
3323     /Subtype/Widget
3324     ~\Fld@annotflags
3325     ~\Fld@pageobjref
3326     ~\Fld@annotnames
3327     /FT/Tx

```

```

3328 ~\Fld@flags
3329 /Q~\Fld@align
3330 /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3331 \bool_if:nT
3332 {
3333   !\int_compare_p:nNn {\Fld@rotation} = {0}
3334   ||
3335   \tl_if_exist_p:N \Fld@bordercolor
3336   ||
3337   \tl_if_exist_p:N \Fld@bcolor
3338 }
3339 {
3340   /MK
3341   <<
3342     \int_compare:nNnF {\Fld@rotation} = {0}
3343     {
3344       /R~\Fld@rotation
3345     }
3346     \tl_if_exist:NT \Fld@bordercolor
3347     {
3348       /BC[\Fld@bordercolor]
3349     }
3350     \tl_if_exist:NT \Fld@bcolor
3351     {
3352       /BG[\Fld@bcolor]
3353     }
3354   >>
3355 }
3356 /DA
3357 (
3358   /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3359   \tl_if_empty:NF {\c_space_tl\Fld@color}
3360 )
3361 /DV(\Hy@escapestring{\Fld@default})
3362 /V(\Hy@escapestring{\Fld@value})
3363 ~\Fld@additionalactions
3364 \int_compare:nNnT { \Fld@maxlen}>{0}
3365 {
3366   /MaxLen~\Fld@maxlen
3367 }
3368 }
3369 \ExplSyntaxOff
3370
3371 \def\PDFForm@Submit{%
3372   /Subtype/Widget%
3373   \Fld@annotflags
3374   \Fld@pageobjref
3375   \Fld@annotnames
3376   /FT/Btn%
3377   \Fld@flags
3378   /H/P%
3379   /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3380   \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3381     \ifx\Fld@bordercolor\relax\else 1\fi

```

```

3382         \space
3383     \else
3384         /MK<<%
3385         \ifnum\Fld@rotation=\z@
3386         \else
3387             /R \Fld@rotation
3388         \fi
3389         \ifx\Fld@bordercolor\relax
3390         \else
3391             /BC[\Fld@bordercolor]%
3392         \fi
3393         >>%
3394     \fi
3395     /A<<%
3396         /S/SubmitForm%
3397         /F<<%
3398             /FS/URL%
3399             /F(\Hy@escapestring{\Form@action})%
3400         >>%
3401         \Fld@submitflags
3402     >>%
3403     \Fld@additionalactions
3404 }
3405 \ExplSyntaxOn
3406     \def\PDFForm@Reset{%
3407         /Subtype/Widget%
3408         \Fld@annotflags
3409         \Fld@pageobjref
3410         \Fld@annotnames
3411         /FT/Btn%
3412         \Fld@flags
3413         /H/P%
3414         /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3415         \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3416             \ifx\Fld@bordercolor\relax\else 1\fi
3417             \space
3418         \else
3419             /MK<<%
3420             \ifnum\Fld@rotation=\z@
3421             \else
3422                 /R~\Fld@rotation
3423             \fi
3424             \ifx\Fld@bordercolor\relax
3425             \else
3426                 /BC[\Fld@bordercolor]%
3427             \fi
3428             >>%
3429         \fi
3430         /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3431         /A<</S/ResetForm>>%
3432         \Fld@additionalactions
3433     }%
3434
3435

```



```

3436 %these patterns are used in hyperref checks.
3437 %it is unclear if they are really useful and if a backend support is
3438 %needed.
3439 \str_case:VnF \c_sys_backend_str
3440 {
3441   { pdfmode }
3442   {
3443     \def\HyPat@ObjRef
3444     {
3445       [0-9]*[1-9][0-9]*~0~R
3446     }
3447   }
3448   { dvipdfmx }
3449   {
3450     \def\HyPat@ObjRef
3451     {
3452       @[\~]+
3453     }
3454   }
3455   { xdvipdfmx }
3456   {
3457     \def\HyPat@ObjRef
3458     {
3459       @[\~]+
3460     }
3461   }
3462 }
3463 { %also set in hyperref sty, so probably not needed.
3464   \def\HyPat@ObjRef/{.+}
3465 }
3466
3467
3468 \ExplSyntaxOff
3469 % UF: removed Hy@writebookmark
3470 %   \Hy@currentbookmarklevel{0}
3471 %   \Hy@numberline
3472 %   \__hypwritetorep
3473 %   counter{bookmark@seq@number}
3474 % removed \HyPsd@SanitizeForOutFile, not needed
3475 % removed \currentpdfbookmark, defined by bookmark,
3476 % should use \newcommand there
3477 % removed \subpdfbookmark, defined by bookmark,
3478 % should use \newcommand there
3479 % removed \belowpdfbookmark, defined by bookmark,
3480 % should use \newcommand there
3481 % removed \pdfbookmark, defined by bookmark,
3482 % \BOOKMARK
3483 % \@BOOKMARK
3484 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3485 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3486 %% removed \ReadBookmarks / unneeded with bookmark.
3487 %% removed \Hy@OutlineName
3488 %% removed \check@bm@number
3489 %% removed \calc@bm@number

```

```

3490
3491 \ifHy@implicit
3492 \else
3493   \expandafter\endinput
3494 \fi
3495 \newlength\Hy@SectionHShift
3496 \def\Hy@SectionAnchorHref#1{%
3497   \ifx\protect\@typeset@protect
3498     \Hy__hypSectionAnchor{#1}%
3499   \fi
3500 }
3501 \DeclareRobustCommand*{\Hy__hypSectionAnchor}[1]{%
3502   \leavevmode
3503   \hbox to 0pt{%
3504     \kern-\Hy@SectionHShift
3505     \Hy@raisedlink{%
3506       \hyper@anchorstart{#1}\hyper@anchorend
3507     }%
3508     \hss
3509   }%
3510 }
3511 \@ifundefined{hyper@nopatch@sectioning}
3512 {
3513   \let\H@old@ssect\@ssect
3514   \def\@ssect#1#2#3#4#5{%
3515     \Hy@MakeCurrentHrefAuto{section*}%
3516     \setlength{\Hy@SectionHShift}{#1}%
3517     \begingroup
3518       \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3519       \toks\tw@\expandafter{%
3520         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3521         #5%
3522       }%
3523       \edef\x{\endgroup
3524         \the\toks@{\the\toks\tw@}%
3525       }\x
3526   }
3527   \let\H@old@schapter\@schapter
3528   \def\@schapter#1{%
3529     \begingroup
3530       \let\@mkboth\@gobbletwo
3531       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3532       \Hy@raisedlink{%
3533         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3534       }%
3535     \endgroup
3536     \H@old@schapter{#1}%
3537   }
3538   \ltx@ifundefined{chapter}{-}{%
3539     \let\Hy@org@chapter\@chapter
3540     \def\@chapter{%
3541       \def\Hy@next{%
3542         \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3543         \Hy@raisedlink{%

```

```

3544         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3545     }%
3546 }%
3547 \ifnum\c@secnumdepth>\m@ne
3548     \ltx@ifundefined{if@mainmatter}%
3549     \iftrue{\csname if@mainmatter\endcsname}%
3550     \let\Hy@next\relax
3551 \fi
3552 \fi
3553 \Hy@next
3554 \Hy@org@chapter
3555 }%
3556 }
3557 \let\H@old@part\@part
3558 \begingroup\expandafter\expandafter\expandafter\endgroup
3559 \expandafter\ifx\csname chapter\endcsname\relax
3560     \let\Hy@secnum@part\z@
3561 \else
3562     \let\Hy@secnum@part\m@ne
3563 \fi
3564 \def\@part{%
3565     \ifnum\Hy@secnum@part>\c@secnumdepth
3566         \phantomsection
3567     \fi
3568     \H@old@part
3569 }
3570 \let\H@old@spart\@spart
3571 \def\@spart#1{%
3572     \Hy@MakeCurrentHrefAuto{part*}%
3573     \Hy@raisedlink{%
3574         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3575     }%
3576     \H@old@spart{#1}%
3577 }
3578 \let\H@old@sect\@sect
3579 \def\@sect#1#2#3#4#5#6[#7]#8{%
3580     \ifnum #2>\c@secnumdepth
3581         \expandafter\@firstoftwo
3582     \else
3583         \expandafter\@secondoftwo
3584     \fi
3585     {%
3586         \Hy@MakeCurrentHrefAuto{section*}%
3587         \setlength{\Hy@SectionHShift}{#3}%
3588         \begingroup
3589             \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3590             \toks\tw@\expandafter{%
3591                 \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3592                 #8%
3593             }%
3594         \edef\x{\endgroup
3595             \the\toks@{\the\toks\tw@}%
3596         }\x
3597     }{%

```

```

3598     \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3599   }%
3600 }
3601 }{}
3602 \expandafter\def\csname Parent-4\endcsname{}
3603 \expandafter\def\csname Parent-3\endcsname{}
3604 \expandafter\def\csname Parent-2\endcsname{}
3605 \expandafter\def\csname Parent-1\endcsname{}
3606 \expandafter\def\csname Parent0\endcsname{}
3607 \expandafter\def\csname Parent1\endcsname{}
3608 \expandafter\def\csname Parent2\endcsname{}
3609 \expandafter\def\csname Parent3\endcsname{}
3610 \expandafter\def\csname Parent4\endcsname{}
3611 %%
3612 %% End of file 'hgeneric-testphase.def'.
3613 \</package>

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

```

```

3652     filebordercolor = [rgb]{0.8,0,0.8},
3653     urlbordercolor  = [rgb]{0.8,0,0.8},
3654     menubordercolor = [rgb]{0,0.2,0.6},
3655     runbordercolor  = [rgb]{0.8,0,0.8}
3656 }
3657
3658 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3659 { %two colors: intern/extern
3660     linkcolor      = [rgb]{0.79216, 0, 0.12549},
3661     filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3662     urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3663     menucolor      = [rgb]{0.79216, 0, 0.12549 },
3664     runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3665     %----- %-----
3666     linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3667     filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3668     urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3669     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3670     runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3671 }
3672
3673 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3674 { %all darkgray
3675     linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3676     filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3677     urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3678     menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3679     runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3680     %----- %-----
3681     linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3682     filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3683     urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3684     menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3685     runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3686 }
3687
3688 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3689 { %dvipsnam.def
3690     linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3691     filecolor      = [rgb]{1, 0, 0}, %Red
3692     urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3693     menucolor      = [rgb]{1, 0, 0}, %Red
3694     runcolor       = [rgb]{1, 0, 0}, %Red
3695     %----- %-----
3696     linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3697     filebordercolor = [rgb]{1, 0, 0}, %Red
3698     urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3699     menubordercolor = [rgb]{1, 0, 0}, %Red
3700     runbordercolor  = [rgb]{1, 0, 0} %Red
3701 }
3702
3703 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3704 { %dvipsnam.def
3705     linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed

```

```

3706     filecolor      = [rgb]{0, 1, 0},      %Green
3707     urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3708     menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3709     runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3710 %----- %-----
3711     linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3712     filebordercolor = [rgb]{0, 1, 0},      %Green
3713     urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3714     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3715     runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3716 }
3717
3718
3719 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3720 {
3721     linkcolor      = [rgb]{0.50196, 0, 0.02353},
3722     filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3723     urlcolor       = [rgb]{0.54118, 0, 0.52941},
3724     menucolor      = [rgb]{0.44706, 0.45882, 0},
3725     runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3726 %----- %-----
3727     linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3728     filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3729     urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3730     menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3731     runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3732 }
3733
3734 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3735 {
3736     linkcolor      = [rgb]{0,0,0},
3737     filecolor      = [rgb]{0,0,0},
3738     urlcolor       = [rgb]{0,0,0},
3739     menucolor      = [rgb]{0,0,0},
3740     runcolor       = [rgb]{0,0,0},
3741 %----- %-----
3742     linkbordercolor = [rgb]{0,0,0},
3743     filebordercolor = [rgb]{0,0,0},
3744     urlbordercolor  = [rgb]{0,0,0},
3745     menubordercolor = [rgb]{0,0,0},
3746     runbordercolor  = [rgb]{0,0,0}
3747 }
3748 </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	
\#	279, 751
\\$	278
\%	752
\-	2134
\.	561, 565, 567
@curropt commands:	
\@curropt:	2655, 2722
\[2134
\]	19, 20, 30, 31, 32, 40, 44, 54, 74, 81, 88, 95, 102, 117, 122, 123, 131, 132, 141, 142, 143, 144, 145, 152, 160, 280, 281, 882, 2657, 2724, 3064
_	559, 561, 565, 567
\]	2134
A	
\A	559, 2134
\Acrobatmenu	18, 173
\addcontentsline	13
\AddToDocumentProperties	388, 2129, 2230
\AddToHook	426, 439
\AddToHookNext	198
\advance	2588, 2662, 2706, 2707
allcolors (hypersetup key)	995
\author	2
B	
\b	563
\beginngroup	209, 277, 375, 2494, 2622, 2651, 2663, 2697, 3086, 3517, 3529, 3558, 3588
\belowpdfbookmark	3479
\bgroup	274, 375
\BOOKMARK	3482
bookmarkstype (hypersetup key)	13
bool commands:	
\bool_if:NTF	267, 296, 355, 698, 714, 726, 739, 747, 777, 834, 893, 965, 975, 1219, 1230
\bool_if:nTF	846, 1904, 1929, 3223, 3331
\bool_lazy_and:nnTF	423
\bool_lazy_or:nnTF	1249, 1298
\bool_new:N	203, 204, 545, 549, 553
\bool_set_true:N	554
bordercolormodel (hypersetup key)	13, 1016
box commands:	
\box_dp:N	635, 2807, 2858, 2893
\box_ht:N	634, 2806, 2857, 2892
\box_new:N	494, 556
\box_set_dp:Nn	2101
\box_set_ht:Nn	2108
\box_use:N	1236, 2809, 2867, 2896
\box_use_drop:N	1241
\box_wd:N	633, 2805, 2856, 2891
\l_tmpa_box	2803, 2805, 2806, 2807, 2809, 2842, 2856, 2857, 2858, 2867, 2889, 2891, 2892, 2893, 2896
C	
\catcode	278, 279
\char	2741, 2938, 2950
\chardef	158
\cite	33
clist commands:	
\clist_item:nn	2161
\clist_map_function:nN	124, 133
\clist_map_inline:nn	2294
\clist_map_variable:nNn	2654, 2721
color commands:	
\color_export:nnN	39, 431, 924, 1037
\color_select:n	932, 968, 1240, 2848, 2852
\color_select:nn	938
\color_set:nn	2, 393, 949
\color_set:nnn	2, 392, 955
color names:	
hyp/annot/file	542
hyp/annot/link	542
hyp/annot/menu	542
hyp/annot/run	542
hyp/annot/url	542
colorfile (hypersetup key)	995
colorlink (hypersetup key)	995
colorlinks (hypersetup key)	981
colormenu (hypersetup key)	995
colorrun (hypersetup key)	995
colorscheme (hypersetup key)	1, 1407
colorurl (hypersetup key)	995
cs commands:	
\cs_generate_variant:Nn	155, 156, 157, 390, 490, 492, 493, 597, 774, 924, 941, 958
\cs_gset:Npn	1211
\cs_gset_eq:NN	409, 420
\cs_if_exist:NTF	12, 449

\ExplSyntaxOn	hidefile (hypersetup key)	1383
... 7, 10, 2526, 2583, 2606, 2966,	hidelink (hypersetup key)	1383
3138, 3159, 3212, 3281, 3319, 3405	hidelinks (hypersetup key)	1383
extension (hypersetup key)	hidemenu (hypersetup key)	1383
13, 1449	hiderun (hypersetup key)	1383
	hideurl (hypersetup key)	1383
F	hook commands:	
\fbox	\hook_gput_code:nnn	
2760, 2848, 2852 402, 413, 961, 971, 1215, 1226	
\fi	\hook_new:n	391, 572, 647
2482,	\hook_new_pair:nn	
2509, 2512, 2521, 2620, 2628, 2629, 260, 288, 312, 329, 347	
2660, 2678, 2681, 2682, 2688, 2691,	\hook_use:n	
2700, 2702, 2709, 2712, 3039, 3061,	. 264, 284, 293, 309, 316, 326, 333,	
3071, 3073, 3079, 3083, 3090, 3091,	343, 352, 371, 589, 653, 662, 703, 719	
3100, 3104, 3108, 3112, 3116, 3120,	\href	4, 5, 19, 262
3124, 3128, 3132, 3136, 3153, 3157,	\hreflaunch	5
3256, 3257, 3258, 3265, 3269, 3273,	\hrefpdf	5, 19, 313, 383
3275, 3277, 3296, 3300, 3304, 3308,	\hrefrun	5, 330, 384
3315, 3380, 3381, 3388, 3392, 3394,	\hrefurl	5, 19, 290, 382
3415, 3416, 3423, 3427, 3429, 3494,	\hss	3508
3499, 3551, 3552, 3563, 3567, 3584	Hy internal commands:	
file (hypersetup key)	\Hy_hyescapeform ..	3050, 3063, 3069
10, 1480	\Hy_hypSectionAnchor ...	3498, 3501
file commands:	HyField internal commands:	
\file_input:n	\HyField_hypAddToFields .	2488, 2515
8	hyp commands:	
fileborderstyle (hypersetup key) 14, 1109	\l_hyp_annot_colorfile_bool ...	543
filecolor (hypersetup key)	\l_hyp_annot_colorlink_bool ...	543
995	\l_hyp_annot_colormenu_bool ...	543
final (hypersetup key)	\l_hyp_annot_colorrun_bool ...	543
1431	\l_hyp_annot_colorurl_bool ...	543
\fontencoding	\l_hyp_annot_ocgcolorfile_bool .	547
2737, 2934, 2946	\l_hyp_annot_ocgcolorlink_bool .	547
\fontfamily	\l_hyp_annot_ocgcolormenu_bool .	547
2736, 2933, 2945	\l_hyp_annot_ocgcolorrun_bool ..	547
\fontseries	\l_hyp_annot_ocgcolorurl_bool ..	547
2738, 2935, 2947	\l_hyp_annot_ocgcolorurl_bool ..	547
\fontshape	\l_hyp_current_dest_name_tl	
2739, 2936, 2948 646, 702, 718	
fp commands:	hyp internal commands:	
\fp_eval:n	\g__hyp_AcroForm_CoFields_prop ..	
2059, 2087, 2103 2449, 2470, 2539, 2541, 2556	
G	\g__hyp_AcroForm_Fields_prop ...	
\gdef 2450, 2463, 2534	
280, 281	\l__hyp_annot_GoTo_bool 698, 714, 726	
\GetDocumentProperties	\l__hyp_annot_GoTo_bool\l__-	
3	_hyp_annot_URI_bool\l__-	
\global	hyp_annot_GoToR_bool\l__-	
2588, 3029	hyp_annot_Named_bool\l__-	
group commands:	hyp_annot_Launch_bool	551
\group_begin: .. 265, 294, 317, 334,	\l__hyp_annot_GoToR_bool	777
353, 587, 701, 717, 741, 779, 836,	\l__hyp_annot_Launch_bool	834
895, 967, 1222, 1239, 2363, 2735,	\l__hyp_annot_Named_bool	893
2785, 2828, 2876, 2909, 2932, 2944		
\group_end:		
308, 325, 342,		
370, 594, 706, 729, 765, 824, 876,		
919, 977, 1242, 1245, 2370, 2742,		
2821, 2869, 2902, 2939, 2951, 2961		
H		
\hbox		
375, 3503		
hbox commands:		
\hbox_overlap_right:n		
1236		
\hbox_set:Nn		
2803, 2842, 2889		
\hbox_set:Nw		
1223		
\hbox_set_end:		
1232		
\hbox_set_to_wd:Nnn		
2085		

\c__hyp_annot_types_seq	\l__hyp_href_run_parameter_tl ...
..... 506, 543, 547, 995, 1296, 1394 208, 229, 341
\l__hyp_annot_URI_bool	__hyp_href_url_aux:n
739	362, 365
\g__hyp_bordercolormodel_str ...	__hyp_href_url_aux:nn
..... 433, 542, 1020, 1039	303, 305
__hyp_citebordercolor_hook_-	\l__hyp_href_url_encode_bool ...
init: 203, 214, 252, 267, 296, 355
398, 411, 420	__hyp_href_url_format:
__hyp_citecolor_hook_init: 209, 215, 257, 368
..... 396, 400, 409	\l__hyp_href_url_ismap_bool
__hyp_clist_display:n . 117, 124, 133 204, 228, 747
__hyp_color_select:n 39, 925, 925, 941	\l__hyp_href_url_protocol_tl ...
__hyp_color_select_aux:wn 205, 216, 255, 307, 369
..... 925, 929, 936	__hyp_link_goto_begin:nw
__hyp_color_set:nn 669, 704, 720
39, 396, 398, 430, 942, 942, 958, 1000, 1036	__hyp_link_goto_end: . 691, 705, 728
__hyp_color_set_aux:nnw 942, 946, 953	\c__hyp_map_annot_hyp_prop
\c__hyp_colorscheme_daleif_prop 3643	\c__hyp_map_hyp_annot_prop
\c__hyp_colorscheme_henryford_- 506, 959,
prop	1023, 1063, 1090, 1098, 1109, 1136,
3734	1144, 1213, 1321, 1357, 1367, 1480
\c__hyp_colorscheme_julian_prop 3658	__hyp_ocg_init: 1155, 1155, 1211, 1221
\c__hyp_colorscheme_phelype_prop	\l__hyp_optlang_regex 2133, 2134, 2163
..... 3719	l__hyp_page/Trans
\c__hyp_colorscheme_szabolcsA_-	570
prop	__hyp_PageLabels_gpush: 604, 604, 617
3688	\l__hyp_para_tmpa_tl 502, 845, 848, 859
\c__hyp_colorscheme_szabolcsB_-	\l__hyp_para_tmpa_tl_uuu\l__hyp_-
prop	text_tmpa_str_uuu\g__hyp_text_-
3703	tmpa_str
\c__hyp_colorscheme_tivv_prop . 3673 499
\l__hyp_dest_box	__hyp_property_if_recorded:nn ..
32, 556, 633, 634, 635, 2085, 2101, 2108 448, 458, 470, 491
\l__hyp_dest_name_tmpa_tl .. 499,	__hyp_property_if_recorded:nnTF
672, 673, 679, 683, 685, 688, 797, 810 474, 2597
\l__hyp_dest_pdfremotestartview_-	__hyp_property_record:nn
tl 448, 451, 463, 490, 2592
525, 805, 1892, 1896	__hyp_property_ref:nn
\g__hyp_dest_pdfstartpage_tl 448, 460, 485, 493, 2601
... 525, 1903, 1905, 1912, 1930, 1937	__hyp_property_ref_undefined_-
\g__hyp_dest_pdfstartview_tl ...	warn:nn ... 448, 459, 472, 492, 2593
525, 1905, 1912, 1923, 1927, 1930, 1937	__hyp_secondoftwowithopt:wnn ...
\l__hyp_dest_pdfview_tl 381, 382, 383, 384
..... 541, 651, 660,	__hyp_setup_info_date_key:nn ...
2054, 2057, 2064, 2067, 2068, 2069, 2216, 2249, 2250
2070, 2071, 2072, 2077, 2081, 2117	__hyp_setup_info_key:nn .. 2135,
\c__hyp_dest_startview_regex ...	2209, 2210, 2211, 2212, 2214, 2215
..... 557, 1890, 1921	__hyp_store_metadata:nn
\c__hyp_dest_undefined_tl 385, 390, 1499, 2177,
..... 505, 678, 679	2195, 2229, 2244, 2253, 2269, 2337
__hyp_destination:nn	__hyp_text_cleanup:N . 577, 577, 591
..... 32, 620, 620, 651, 660	\l__hyp_text_enc_dest_print_tl ..
\l__hyp_filename_tmpa_tl ... 499, 528, 796
780, 782, 787, 788, 793, 840, 841, 859	\l__hyp_text_enc_dest_tl
__hyp_href_pdf_aux:nn 32, 528, 627, 684
319, 322	\l__hyp_text_enc_file_print_tl ..
\l__hyp_href_pdf_destination_tl 528, 839
..... 206, 217, 256, 324	
\l__hyp_href_pdf_page_tl 207, 225, 804	
__hyp_href_run_aux:nn	
336, 339	

\l__hyp_text_enc_info_print_tl	528, 602	debug	1431
\l__hyp_text_enc_para_print_tl	528, 844	destlabel	13
\l__hyp_text_enc_uri_print_tl	269, 272, 298, 301, 357, 360, 528, 744, 1491	draft	1431
__hyp_text_pdfstring:nnN	585, 585, 597, 599, 602, 625, 682, 742, 794, 837, 842, 1491	extension	13, 1449
__hyp_text_pdfstring_info:nn	600, 600, 2166, 2170, 2189, 2199	file	10, 1480
__hyp_text_purify:nn	573, 573, 590	fileborderstyle	14, 1109
__hyp_text_string_from_unicode:nn	581, 581, 592	filecolor	995
\g_hyp_text_tmpa_str	504, 593, 595	final	1431
\l__hyp_text_tmpa_str	503, 590, 591, 592, 593	hidefile	1383
\l__hyp_tmpa_box	494, 1223, 1236, 1241	hidelink	1383
\l__hyp_tmpa_int	494	hidelinks	1383
\l__hyp_tmpa_seq	494, 1890, 1892, 1921, 1923, 2044, 2045, 2049, 2051, 2075, 2082, 2083, 2084, 2093, 2095, 2105, 2110, 2163, 2164, 2170, 2543, 2545, 2551	hidemenu	1383
\l__hyp_tmpa_str	494, 2166, 2170, 2172, 2174, 2189, 2190, 2192, 2199, 2200, 2204	hiderun	1383
\l__hyp_tmpa_tl	494, 628, 632, 640, 1040, 1044, 1491, 1492, 1497, 1888, 1890, 1919, 1921, 2051, 2052, 2059, 2161, 2163	hideurl	1383
\l__hyp_uri_tmpa_tl	499, 745, 746	hypertexnames	1449
hyp/anchor	646	link	10, 1480
hyp/annot/file (color name)	542	linkborder	13
hyp/annot/link (color name)	542	linkborderstyle	14, 1109
hyp/annot/menu (color name)	542	linkcolor	995
hyp/annot/run (color name)	542	linkfileprefix	1449
hyp/annot/url (color name)	542	linktoc	1449
hyp/text/pdfstring	572	linktocpage	1449
\hypercalcbp	11, 18, 161	localanchorname	1449
\HyperDestNameFilter	13, 626, 683	menu	10, 1480
\hypersetup 1, 2, 5, 9, 10, 13, 18, 67, 99, 180		menuborder	13
\hypersetup keys:		menuborderstyle	14, 1109
allcolors	995	menucolor	995
bookmarkstype	13	naturalnames	1449
bordercolormodel	13, 1016	nesting	13
colorfile	995	ocgcolorfile	1249
colorlink	995	ocgcolorlink	1249
colorlinks	981	ocgcolorlinks	1249
colormenu	995	ocgcolormenu	1249
colorrund	995	ocgcolorrun	1249
colorscheme	1, 1407	ocgcolorurl	1249
colorurl	995	pageanchor	1449
		pdfauthor	2133
		pdfborder	13
		pdfborderstyle	14, 1109
		pdfcreationdate	14, 2216
		pdfcreator	2133
		pdfencoding	1421
		pdfinfo	2283
		pdfkeywords	2133
		pdflang	14, 2122
		pdflinkmargin	14
		pdfmetadate	14, 2216
		pdfmoddate	14, 2216
		pdfproducer	2133
		pdfremotestartview	11
		pdfstartview	11
		pdfsubject	2133
		pdftitle	2133
		pdftrapped	2255

pdfversion	1421	2122, 2137, 2180, 2218, 2233, 2251, 2255, 2283, 2335, 2341, 2354, 2374
pdfview	11, 2040	
plainpages	1449	\l_keys_key_str 191, 2204, 2439
run	10, 1480	\keys_set:nn 183, 220, 239, 266, 295, 318, 335, 354, 404, 415, 1413, 1420, 2260, 2287, 2290, 2291, 2292, 2293, 2364, 2439
runborder	13	\keys_set_known:nn 2442
runborderstyle	14, 1109	\kvsetkeys 182, 2530
runcolor	995	
unicode	1421	
url	10, 1480	
urlborder	13	
urlborderstyle	14, 1109	
urlcolor	995	
verbose	1431	
hypertextnames (hypersetup key)	1449	
hypListbox internal commands:		
_hypListbox	2710, 2769	
hypRadio internal commands:		
_hypRadio	2695, 2717	
hypwritetorep internal commands:		
_hypwritetorep	3472	
I		
\ifcase	3256, 3380, 3415	
\ifcsname	3033	
\ifdim	2660, 2699, 2700	
\ifnum	3087, 3256, 3262, 3293, 3380, 3385, 3415, 3420, 3547, 3565, 3580	
\iftrue	3549	
\ifx	2492, 2629, 2689, 3064, 3076, 3080, 3097, 3101, 3105, 3109, 3113, 3117, 3121, 3125, 3129, 3133, 3150, 3154, 3257, 3258, 3266, 3270, 3277, 3297, 3301, 3308, 3309, 3381, 3389, 3416, 3424, 3497, 3559	
\immediate	2476, 2479	
int commands:		
\int_compare:nNnTF	2049, 2075, 2761, 3179, 3232, 3342, 3364	
\int_compare_p:nNn	3225, 3333	
\int_eval:n	803	
\int_max:nn	804	
\int_new:N	497	
iow commands:		
\iow_newline:	2982, 2988, 3001	
K		
\kern	3504	
keys commands:		
\keys_define:nn	186, 194, 212, 250, 394, 981, 997, 1004, 1016, 1025, 1050, 1065, 1084, 1111, 1130, 1253, 1267, 1279, 1302, 1308, 1323, 1352, 1383, 1396, 1407, 1421, 1431, 1436, 1449, 1461, 1482, 1487, 2040,	
L		
\label	9, 13	
\LayoutCheckField	2922	
\LayoutChoiceField	2692	
\LayoutPushButtonField	2798, 2815	
\LayoutTextField	2630	
\leavevmode	2631, 2728, 3502	
legacy commands:		
\legacy_if:nTF	168, 437, 444, 612, 2568, 2794, 2835, 2885, 2918, 3202	
\let	382, 383, 384, 709, 760, 767, 820, 826, 872, 878, 921, 2452, 2453, 2455, 2456, 2483, 2579, 2580, 2581, 2612, 2613, 2647, 2648, 2690, 3029, 3036, 3038, 3043, 3060, 3094, 3513, 3527, 3530, 3539, 3550, 3557, 3560, 3562, 3570, 3578	
link (hypersetup key)	10, 1480	
linkborder (hypersetup key)	13	
linkborderstyle (hypersetup key)	14, 1109	
linkcolor (hypersetup key)	995	
linkfileprefix (hypersetup key)	1449	
linktoc (hypersetup key)	1449	
linktocpage (hypersetup key)	1449	
localanchorname (hypersetup key)	1449	
M		
\MakeButtonField	2803, 2818, 2842, 2889, 2900	
\MakeChoiceField	2778	
\MakeFieldObject	2443	
\MakeRadioField	2760	
\MakeTextField	2639	
\mbox	1233	
menu (hypersetup key)	10, 1480	
menuborder (hypersetup key)	13	
menuborderstyle (hypersetup key)	14, 1109	
menucolor (hypersetup key)	995	
mode commands:		
\mode_if_horizontal:TF	622, 644	
\mode_leave_vertical:	263, 292, 315, 332, 351, 671, 754, 812, 861, 898, 2772, 2800, 2817, 2839, 2881, 2924	

msg commands:

- \msg_error:nn 425, 2814, 2899
- \msg_info:nnn 2147, 2152
- \msg_line_context: 82
- \g_msg_module_name_prop 11
- \msg_new:nnn 50, 57, 62, 66, 70, 77,
84, 91, 98, 105, 111, 118, 127, 137, 148
- \msg_new:nnnn 14, 25, 36
- \msg_warning:nn 170, 1427
- \msg_warning:nnn 179, 675, 916
- \msg_warning:nnnn
..... 1271, 1312, 1525, 1569,
1603, 1626, 1688, 1717, 1751, 1766,
1779, 1792, 1817, 1852, 1868, 1895,
1926, 1959, 1973, 2000, 2034, 2116
- \msg_warning:nnnn 190,
1343, 1377, 1469, 1545, 1582, 1669,
1702, 1727, 1803, 1830, 1881, 1986,
2013, 2277, 2381, 2400, 2419, 2430

N

- naturalnames (hypersetup key) 1449
- nesting (hypersetup key) 13
- \newcommand 164, 3476, 3478, 3480
- \newcount 2584, 2768
- \NewDocumentCommand 2443
- \NewExpandableDocumentCommand 381
- \newlength 3495
- \noexpand ... 2666, 2667, 2668, 2669, 3047
- \nolinkurl 4

O

- ocgcolorfile (hypersetup key) 1249
- ocgcolorlink (hypersetup key) 1249
- ocgcolorlinks (hypersetup key) 1249
- ocgcolormenu (hypersetup key) 1249
- ocgcolorrun (hypersetup key) 1249
- ocgcolorurl (hypersetup key) 1249

P

- pageanchor (hypersetup key) 1449
- \paperwidth 3
- \PassOptionsToPackage ... 446, 1441, 1446

pdf commands:

- \pdf_bdcobject:nn 1235, 1238
- \pdf_destination:nn 27, 155, 639
- \pdf_destination:nnnn 632
- \pdf_emc: 1237, 1243
- \pdf_link_user:nnn 2775
- \pdf_name_from_unicode_e:n
..... 245, 784, 900
- \pdf_object_if_exist:nTF 782
- \pdf_object_new:n 1157,
1158, 1159, 1160, 2969, 2970, 2971
- \pdf_object_ref:n 156, 793,
1163, 1165, 1187, 1188, 1191, 1195,
1200, 1205, 1210, 2561, 2563, 3027
- \pdf_object_ref_last: 814, 2369
- \pdf_object_unnamed_write:nn ...
..... 813, 2365
- \pdf_object_write:nnn 1161,
1167, 1177, 1189, 2972, 3014, 3021
- \pdf_pageobject_ref:n
..... 157, 1912, 1937, 2599
- \pdf_string_from_unicode:nnN .. 583
- \pdf_version:
... 1573, 1692, 1721, 1755, 1770,
1796, 1821, 1856, 1872, 1977, 2004
- \pdf_version_compare:NnTF
1563, 1654, 1676, 1712, 1745, 1760,
1786, 1811, 1839, 1862, 1967, 1994
- \pdf_version_compare_p:Nn
..... 850, 1250, 1299
- \pdf_version_major:
..... 167, 1251, 1274, 1300, 1316
- \pdf_version_minor: .. 166, 1274, 1316

pdfannot commands:

- \pdfannot_box:nnnn
... 2634, 2745, 2804, 2855, 2890, 2954
- \pdfannot_box_ref_last: .. 2490, 2506
- \pdfannot_dict_put:nnn
..... 755, 814, 862, 901, 1041,
1076, 1100, 1122, 1146, 1328, 1359
- \pdfannot_dict_remove:nn .. 1031,
1071, 1092, 1117, 1138, 1336, 1369
- \pdfannot_link:nnn . 756, 815, 863, 902
- \pdfannot_link_goto_begin:nw .. 688
- \pdfannot_link_goto_end: 693
- \pdfannot_link_margin:n 8, 1607
- \c_pdfannot_link_types_seq 551

pdfauthor (hypersetup key) 2133

\pdfbookmark 3481

pdfborder (hypersetup key) 13

pdfborderstyle (hypersetup key) . 14, 1109

pdfcreationdate (hypersetup key) 14, 2216

pdfcreator (hypersetup key) 2133

\pdfdest 11

pdfdict commands:

- \pdfdict_new:n 570, 733, 770, 828, 887
- \pdfdict_put:nnn
.... 237, 244, 571, 734, 735, 746,
749, 771, 772, 784, 790, 800, 810,
829, 830, 841, 856, 888, 889, 899,
1633, 1634, 1638, 1639, 2378, 2388,
2392, 2395, 2397, 2416, 2425, 2427
- \pdfdict_remove:nn 234, 853, 1643, 1644
- \pdfdict_use:n
... 755, 813, 862, 868, 901, 906, 2367

<code>\RequirePackage</code>	6, 31, 441, 3484	<code>\str_uppercase:n</code>	2260, 2266, 2271
<code>run</code> (hypersetup key)	10, 1480	<code>\string</code>	2477,
<code>runborder</code> (hypersetup key)	13		2480, 2501, 2519, 3052, 3056, 3067
<code>runborderstyle</code> (hypersetup key)	14, 1109	<code>\subpdfbookmark</code>	3477
<code>runcolor</code> (hypersetup key)	995	sys commands:	
		<code>\c_sys_backend_str</code>	3439
S			
<code>\selectfont</code>	2740, 2937, 2949	T	
seq commands:		T _E X and L ^A T _E X 2 _ε commands:	
<code>\seq_const_from_clist:Nn</code>	506	<code>\@BOOKMARK</code>	3483
<code>\seq_count:N</code>	2049, 2075	<code>\@CheckBox</code>	2905
<code>\seq_get_right:NN</code>	2051	<code>\@ChoiceMenu</code>	2645
<code>\seq_if_empty:NTF</code>	2164	<code>\@Form</code>	2528
<code>\seq_item:Nn</code>	1892, 1923,	<code>\@PushButton</code>	2783
	2045, 2093, 2095, 2105, 2110, 2170	<code>\@Reset</code>	2872
<code>\seq_map_inline:Nn</code>		<code>\@Submit</code>	2824
	543, 547, 551, 995, 1296, 1394, 2551	<code>\@TextField</code>	2610
<code>\seq_new:N</code>	496	<code>\@bookmarksopenlevel</code>	1508
<code>\seq_put_right:Nn</code>		<code>\@bsphack</code>	453, 465
	2082, 2083, 2084, 2543	<code>\@chapter</code>	3539, 3540
<code>\seq_set_split:Nnn</code>	2044	<code>\@currDisplay</code>	2659, 2727
<code>\seq_sort:Nn</code>	2545	<code>\@currValue</code>	2725, 2755
<code>\setbox</code>	375	<code>\@currentHref</code>	
<code>\setlength</code>	3516, 3587		3520, 3533, 3544, 3574, 3591
<code>\setpdflinkmargin</code>	8	<code>\@curropt</code>	2644, 2654, 2657, 2721, 2724
<code>\settowidth</code>	2659	<code>\@empty</code>	2613, 2629, 3097,
<code>\show</code>	2719		3101, 3105, 3109, 3113, 3117, 3121,
sort commands:			3125, 3129, 3133, 3277, 3308, 3309
<code>\sort_return_same:</code>	2549	<code>\@endForm</code>	2579
<code>\sort_return_swapped:</code>	2548	<code>\@esphack</code>	455, 467
<code>\space</code>	480, 2727, 3259, 3276,	<code>\@firstofone</code>	2483, 3038, 3043, 3060
	3277, 3307, 3308, 3382, 3414, 3417	<code>\@firstoftwo</code>	3581
<code>\spacefactor</code>	622, 644	<code>\@for</code>	2655, 2722
<code>\stockwidth</code>	3	<code>\@gobbletwo</code>	3530
str commands:		<code>\@hyper@launch</code>	38, 882
<code>\c_colon_str</code>	882	<code>\@ifnextchar</code>	274
<code>\c_hash_str</code>	751	<code>\@ifundefined</code>	3052, 3511
<code>\c_percent_str</code>	752	<code>\@latex@warning</code>	477
<code>\str_case:nnTF</code>	2143, 3439	<code>\@mainaux</code>	2476, 2479, 2499, 2518
<code>\str_case_e:nnTF</code>	2045	<code>\@mkboth</code>	3530
<code>\str_compare:nNnTF</code>	2547	<code>\@nil</code>	3050, 3063, 3069
<code>\str_gset:Nn</code>	1020	<code>\@part</code>	3557, 3564
<code>\str_gset_eq:NN</code>	593	<code>\@pdfauthor</code>	23
<code>\str_head:n</code>	2266, 2271	<code>\@pdfborder</code>	162
<code>\str_if_eq:nnTF</code>		<code>\@pdfborderstyle</code>	163
	629, 2172, 2190, 2200, 3141	<code>\@savsf</code>	622, 644
<code>\str_if_eq_p:nn</code>	848, 1251, 1300	<code>\@schapter</code>	3527, 3528
<code>\str_lowercase:n</code>		<code>\@secondoftwo</code>	3583
	2045, 2195, 2244, 2267, 2272	<code>\@sect</code>	3578, 3579
<code>\str_new:N</code>	498, 503, 504, 542	<code>\@spart</code>	3570, 3571
<code>\str_set:Nn</code>	575	<code>\@ssect</code>	3513, 3514
<code>\str_set_convert:Nnnn</code>	8	<code>\@tempdima</code>	2653, 2660, 2662, 2699,
<code>\str_set_eq:NN</code>	595		2700, 2701, 2705, 2706, 2707, 2708
<code>\str_tail:n</code>	2267, 2272	<code>\@tempdimb</code>	2659, 2660

\@typeset@protect	3497	\Fld@onmousedown@code ...	3121, 3123
\BKM@color	434	\Fld@onmouseup@code	3125, 3127
\c@secnumdepth	3547, 3565, 3580	\Fld@pageobjref	
\calc@bm@number	3489	2581, 2595, 3032, 3164,
\check@bm@number	3488		3217, 3250, 3286, 3325, 3374, 3409
\define@key	428	\Fld@radiosymbol	3305
\Fld@additionalactions	3139, 3209,	\Fld@rotation	3179,
	3243, 3279, 3316, 3363, 3403, 3432		3181, 3225, 3232, 3234, 3256, 3262,
\Fld@align	3168, 3254, 3329		3264, 3293, 3295, 3333, 3342, 3344,
\Fld@altname ..	3076, 3078, 3150, 3152		3380, 3385, 3387, 3415, 3420, 3422
\Fld@annotflags	3163,	\Fld@submitflags	3401
	3216, 3249, 3285, 3324, 3373, 3408	\Fld@validate@code	3105, 3107
\Fld@annotnames	3148, 3165,	\Fld@value	
	3218, 3251, 3287, 3326, 3375, 3410	..	2613, 2629, 2648, 2689, 2690, 3362
\Fld@bcolor	3187, 3189, 3270,	\Fld@width	2614,
	3272, 3301, 3303, 3337, 3350, 3352		2628, 2635, 2639, 2649, 2688, 2699,
\fld@bcolor	3258		2701, 2746, 2760, 2778, 2796, 2826,
\Fld@bordercolor	3183,		2837, 2874, 2886, 2910, 2920, 2955
	3185, 3227, 3236, 3238, 3257, 3266,	\Fld@X@additionalactions	
	3268, 3297, 3299, 3335, 3346, 3348,	3096, 3141, 3144
	3381, 3389, 3391, 3416, 3424, 3426	\Form@action	3399
\Fld@borderstyle	3169,	\G@refundefinedtrue	476
	3222, 3255, 3291, 3330, 3379, 3430	\H@old@part	3557, 3568
\Fld@borderwidth ..	2706, 2707, 3169,	\H@old@chapter	3527, 3536
	3222, 3255, 3291, 3330, 3379, 3430	\H@old@sect	3578, 3589, 3598
\Fld@calculate@code ..	2492, 3109, 3111	\H@old@spart	3570, 3576
\Fld@calculate@sortkey	2503	\H@old@ssect	3513, 3518
\Fld@cbsymbol	3191	\href@	274, 280
\Fld@charsize		\href@split	280, 281
..	2705, 3195, 3276, 3307, 3358, 3414	\Hy@abspage	616
\Fld@checkequals	2657, 2724	\Hy@activeanchorfalse	667
\Fld@choices	3278	\Hy@activeanchortrue	658
\Fld@color	3196, 3198, 3277, 3308, 3359	\Hy@AtBeginDocument	2457, 2473
\Fld@default		\Hy@bookmarkstyle	1509
...	2612, 2629, 2647, 2690, 2719,	\Hy@chapapp	3531, 3542
	2720, 2908, 3309, 3313, 3314, 3361	\Hy@colorlink	34
\Fld@flags	3167,	\Hy@currentbookmarklevel	3470
	3220, 3253, 3289, 3328, 3377, 3412	\Hy@DisableOption	172
\Fld@format@code	3101, 3103	\Hy@drafttrue	1440
\Fld@height		\Hy@escapeform	
	2615, 2636, 2639, 2650, 2708, 2747,	2633, 2730, 2774, 2802,
	2760, 2778, 2827, 2875, 2911, 2956		2841, 2888, 2926, 3034, 3041, 3044
\Fld@keystroke@code	3097, 3099	\Hy@escapestring	3036,
\Fld@listcount	2718, 2726, 2761, 2768		3038, 3043, 3046, 3047, 3050, 3051,
\Fld@mappingname	3080, 3082, 3154, 3156		3060, 3063, 3099, 3103, 3107, 3111,
\Fld@maxlen	3364, 3366		3115, 3119, 3123, 3127, 3131, 3135,
\Fld@menulength	2652, 2658, 2705		3191, 3242, 3305, 3361, 3362, 3399
\Fld@name		\Hy@finaltrue	1445
..	2611, 2646, 2784, 2907, 3075, 3149	\Hy@FormObjects	2533, 2967, 3029
\Fld@onblur@code	3117, 3119	\Hy@gttemp	3092, 3094
\Fld@onclick@code	3242	\Hy@href	274
\Fld@onenter@code	3129, 3131	\Hy@href@nextactionraw	238
\Fld@onexit@code	3133, 3135	\Hy@href@page	226
\Fld@onfocus@code	3113, 3115	\Hy@linkfileprefix	1454

\Hy@linktoc	1465	\HyField@PDFChoices	2771
\Hy@MakeCurrentHref	199	\HyField@SetKeys	2623, 2664,
\Hy@MakeCurrentHrefAuto		2667, 2686, 2786, 2829, 2877, 2912	
.....	3515, 3531, 3542, 3572, 3586	\HyPat@ObjRef ..	3443, 3450, 3457, 3464
\Hy@next	3541, 3550, 3553	\hyper@link	282
\Hy@numberline	165, 3471	\hyper@anchor	646
\Hy@org@chapter	3539, 3554	\hyper@anchorend	
\Hy@OutlineName	3487	646, 3506, 3533, 3544, 3574
\Hy@OutlineRerunCheck	3485	\hyper@anchorstart	
\Hy@pdfmajorversion	167	646, 3506, 3533, 3544, 3574
\Hy@pdfminorversion	166	\hyper@link	34, 696
\Hy@pdfstringtrue	30, 588	\hyper@linkend	34, 724
\Hy@pdfversion	3087	\hyper@linkfile	324, 775
\Hy@pstringdef	598, 3067	\hyper@linklaunch ..	37, 341, 832, 885
\Hy@PutCatalog	604	\hyper@linknamed	38, 175, 891
\Hy@raisedlink	3505, 3532, 3543, 3573	\hyper@linkstart	34, 712
\Hy@RestoreLastskip	643	\hyper@linkurl	307, 368, 737
\Hy@safe@activestruer	753, 2495	\hyper@normalise	
\Hy@SaveLastskip	623	274, 303, 319, 336, 362, 375
\Hy@secnum@part	3560, 3562, 3565	\HyPL@Labels	606, 616
\Hy@SectionAnchorHref	3496, 3520, 3591	\HyPL@StorePageLabel	604
\Hy@SectionHShift		\HyPsd@SanitizeForOutFile	3474
.....	3495, 3504, 3516, 3587	\if@filesw	2475, 2498, 2517
\Hy@StepCount	2658, 2726	\ifFld@combo	2673, 2703
\Hy@temp	2496, 2511, 3049, 3050	\ifFld@hidden	2628, 2688
\Hy@Unicodefalse	3089	\ifFld@multiline	2616
\Hy@VerboseAnchor	624	\ifFld@popdown	2674
\Hy@VerboseLinkStart	700, 716	\ifFld@radio	2670, 2693
\Hy@VerboseLinkStop		\ifHy@implicit	3491
.....	707, 730, 763, 822, 874, 912	\ifHy@pdfescapeform	3035, 3045
\Hy@VersionChecked	158	\ifHy@unicode	3088
\Hy@WrapperDef	620	\kv@set@family@handler	178
\Hy@xspace@end	704, 762, 821, 873, 911	\ltx@empty	2452,
\HyAnn@AbsPageLabel		2453, 2492, 2579, 2580, 2581, 2612	
.....	2580, 2586, 2632,	\ltx@gobble	2455
2729, 2773, 2801, 2840, 2887, 2925		\ltx@gobbletwo	2456
\HyAnn@Count	2584, 2585, 2588, 2589,	\ltx@ifUndefined	3538, 3548
2590, 2591, 2592, 2593, 2597, 2601		\ltx@one	2588
\HyField@AddToFields ..	2486, 2640,	\ltx@ReturnAfterFi	3068
2762, 2779, 2810, 2866, 2895, 2959		\ltx@zero	2585
\HyField@afields	2452	\m@ne	3547, 3562
\HyField@AfterAuxOpen	2457, 2483, 2516	\OBJ@OCG@view	164
\HyField@AuxAddToCoFields		\p@	2662
.....	2456, 2468, 2480, 2501	\pdf@ifdraftmode	2531
\HyField@AuxAddToFields		\PDF@SetupDoc	159
.....	2455, 2461, 2477, 2519	\PDFForm@Check	2926, 2958, 3160
\HyField@cofields	2453	\PDFForm@List	2774, 2777, 3247
\HyField@FlagsCheckBox	2917	\PDFForm@Name	
\HyField@FlagsChoice	2698	2626, 2687, 2790, 2916, 3074
\HyField@FlagsPushButton		\PDFForm@Push	2802, 2808, 3213
.....	2793, 2833, 2884	\PDFForm@Radio	2730, 2750, 3282
\HyField@FlagsRadioButton	2694	\PDFForm@Reset	2888, 2894, 3406
\HyField@FlagsSubmit	2834	\PDFForm@Submit	2841, 2860, 3371
\HyField@FlagsText	2627	\PDFForm@Text	2633, 2638, 3321

<code>\protected@edef</code>	672	<code>\tl_if_exist:N</code> TF	3236, 3346, 3350
<code>\strip@pt</code> .	3195, 3276, 3307, 3358, 3414	<code>\tl_if_exist_p:N</code>	3227, 3335, 3337
<code>\toks@</code>	3518, 3524, 3589, 3595	<code>\tl_if_head_eq_charcode:n</code> NTF	927, 944
<code>\tw@</code>	3519, 3524, 3590, 3595	<code>\tl_new:N</code> ..	205, 206, 207, 208, 495,
<code>\url@</code>	375, 378		499, 500, 501, 502, 525, 526, 527,
<code>\Url@def</code>	376		528, 529, 530, 531, 532, 533, 541, 646
<code>\Url@HyperHook</code>	376	<code>\tl_set:Nn</code>	
<code>\XR@ext</code>	1451	 225, 226, 238, 269, 272, 298,
<code>\z@</code>	375, 2653, 3256, 3262,		301, 357, 360, 387, 535, 536, 537,
	3293, 3380, 3385, 3415, 3420, 3560		538, 539, 540, 652, 661, 702, 718,
<code>\zref@labelbylist</code>	2590		780, 1888, 1892, 1896, 1919, 2054,
<code>\zref@labelbyprops</code>	2589		2057, 2064, 2067, 2068, 2069, 2070,
<code>\zref@refused</code>	2591		2071, 2072, 2077, 2081, 2117, 2161
<code>\texorpdfstring</code>	30	<code>\tl_set_eq:NN</code>	679
text commands:		<code>\tl_to_str:N</code>	782, 788, 793
<code>\text_expand:n</code>	780	<code>\tl_to_str:n</code>	20, 21, 31, 32, 33, 45, 479
<code>\text_purify:n</code>	8, 575	token commands:	
<code>\textsf</code>	2848, 2852	<code>\token_to_str:N</code>	41
<code>\the</code>	616,	<code>\toks</code>	3519, 3524, 3590, 3595
	2589, 2590, 2591, 2592, 2593, 2597,		
	2601, 2701, 2705, 2708, 3524, 3595		
<code>\theH...</code>	12		
<code>\thepage</code>	480		
<code>\ThisShouldNotHappen</code>	3054		
<code>\title</code>	2		
tl commands:			
<code>\c_space_tl</code>	165, 1164,		
	1195, 1200, 1205, 2755, 2756, 2763,		
	2978, 2980, 2990, 2992, 2994, 2997,		
	2999, 3195, 3198, 3330, 3358, 3359		
<code>\tl_const:Nn</code>	505		
<code>\tl_gput_right:Nn</code>	616		
<code>\tl_gset:Nn</code>	1903, 1923, 1927		
<code>\tl_if_blank:n</code> TF			
	798, 2141, 2184, 2222, 2237, 2345, 2358		
<code>\tl_if_empty:N</code> TF			
	... 673, 1492, 3183, 3187, 3196, 3359		
<code>\tl_if_empty:n</code> TF			
	... 232, 1029, 1069, 1088,		
	1115, 1134, 1678, 1734, 1841, 2126		
<code>\tl_if_empty_p:N</code>	1905, 1930		
<code>\tl_if_eq:Nn</code> TF	2052		

U

unicode (hypersetup key)	1421
<code>\Url</code>	4, 5, 209
url (hypersetup key)	10, 1480
<code>\url</code>	4–6, 22, 349, 375
urlborder (hypersetup key)	13
urlborderstyle (hypersetup key) .	14, 1109
urlcolor (hypersetup key)	995
<code>\urldef</code>	5, 22, 375, 377
<code>\Urlfont</code>	6
<code>\UrlSpecial</code>	4
use commands:	
<code>\use:n</code>	880

V

verbose (hypersetup key)	1431
--------------------------------	------

W

<code>\write</code>	2476, 2479, 2499, 2518
---------------------------	------------------------

X

<code>\x</code>	2665, 2685, 3523, 3525, 3594, 3596
-----------------------	------------------------------------