

The hyperref-generic module

A generic driver for hyperref

The L^AT_EX Project*

Version 0.96a, released 2023-10-09

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 `cmymk`. `rgb` is the default. It does *not* change the model of text colors. Be aware that while the PDF format allows `cmymk` (4 numbers) in the `/C` key of an annotation, this is often ignored by pdf viewers and the colors can be wrong.
- The boolean keys `url`, `link`, `run`, `menu`, `file` allow to deactivate locally the link types.

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

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

The default is `phelype`.

7.3 Public interfaces

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

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

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

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

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

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

7.4 Changed behaviour

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

Color keys accept the following input syntax:

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

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

frenchlinks The option `frenchlinks` does nothing at all.

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

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

8 PDF strings

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

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

The new method is under heavy development!

Important differences here are

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

9 Package options from hyperref

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

Options that still must be set as package options are

- `backref`
- `CJKbookmarks` this key should not be used anymore. At some time it will be removed.
- `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{hggeneric-testphase.def}[2023-10-09 v0.96a %
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:nnx {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:nnxxx { 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:nnx
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\bgroup\Hy@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 {xn,nx,xx}
```

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` (*color name*)

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

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

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

```
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 ocgcolors. They are public so that other packages can query the state too.

```
\l_hyp_annot_ocgcolorlink_bool
\l_hyp_annot_ocgcolorurl_bool
\l_hyp_annot_ocgcolorfile_bool
\l_hyp_annot_ocgcolorrund_bool
\l_hyp_annot_ocgcolormenu_bool
547 \seq_map_inline:Nn \c__hyp_annot_types_seq
548 {
549   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
550 }
```

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

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

```
not_Named_bool_1111\l_hyp_annot_Launch_bool
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

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

```
\c__hyp_dest_startview_regex
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:Nx #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 {xnN,onN,xoN,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:xnN {#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:nnx {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:xoN
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:nnx
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:xoN
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:xoN
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@activestruer
754         \mode_leave_vertical:
755         \pdfannot_dict_put:nnx {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
756         \pdfannot_link:nxn { 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
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:nnx { l_pdffile/Filespec } { Subtype } { \pdf_name_from_unicode_e:n {ap
785       \exp_args:Nnox
786       \pdffile_embed_file:nnn
787       {}
788       {\l__hyp_filename_tmpa_tl }
789       {\__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
790     }
791     \pdfdict_put:nnx
792     {l_hyp/annot/A/GoToR}
793     {F}
794     {\pdf_object_ref:e {\__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
795     \__hyp_text_pdfstring:nnN
796     { #3 }
797     { \l__hyp_text_enc_dest_print_tl }
798     \l__hyp_dest_name_tmpa_tl
799     \tl_if_blank:eTF {#3}
800     {
801       \pdfdict_put:nnx {l_hyp/annot/A/GoToR}{D}
802       {
803         [
804           \int_eval:n
805           { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
806           /\l__hyp_dest_pdfremotestartview_tl
807         ]
808       }
809     }
810     {
811       \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
812     }
813     \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>

```

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

```

```

817         { GoToR }
818     {
819     }
820     {
821         \let\protect\relax
822         #1\Hy@xspace@end
823         \Hy@VerboseLinkStop %where is the start??
824     }
825     \group_end:
826 }
827 {{\let\protect\relax#1}}
828 }

```

10.5 Launch links

We define `\hyper@linklaunch` for naming consistency

```

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

```

```

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

```

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

```

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

```

10.6 Named links (menu)

We also define \hyper@linknamed for consistency.

```

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

```

```

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

```

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.

```

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

```

```

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

```

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

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

```

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

```

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

```

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

```

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

```

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

```



```

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

colorurl (setup key)
colorlink (setup key) 994 \seq_map_inline:Nn \c__hyp_annot_types_seq
colorrun (setup key) 995 {
colormenu (setup key) 996     \keys_define:nn { hyp / setup }
colorfile (setup key) 997     {
    urlcolor (setup key) 998         ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
linkcolor (setup key) 999         ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
    runcolor (setup key) 1000     }
menucolor (setup key) 1001 }
filecolor (setup key) 1002
allcolors (setup key) 1003 \keys_define:nn { hyp / setup }
1004 {
1005     ,allcolors .meta:n =
1006     {
1007         ,urlcolor=#1
1008         ,linkcolor=#1
1009         ,runcolor=#1
1010         ,filecolor=#1
1011         ,menucolor=#1
1012     }
1013     ,allcolors .value_required:n = true
1014 }

```

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)

1015 \keys_define:nn { hyp / setup }
1016 {
1017     ,bordercolormodel .choices:nn =
1018     {rgb,cmyk}
1019     { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
1020     ,bordercolormodel .initial:n ={rgb}
1021 }

1022 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1023 {

```

```

1024 \keys_define:nn { hyp / setup }
1025 {
1026   #1bordercolor .code:n =
1027   {
1028     \tl_if_empty:nTF { ##1 }
1029     {
1030       \pdfannot_dict_remove:nn
1031       {link/#2}
1032       { C }
1033     }
1034     {
1035       \__hyp_color_set:ne {hyp/color/#1border}{##1}
1036       \color_export:nVN
1037       {hyp/color/#1border}
1038       \g__hyp_bordercolormodel_str
1039       \l__hyp_tmpa_tl
1040       \pdfannot_dict_put:nnx
1041       {link/#2}
1042       { C }
1043       { [\l__hyp_tmpa_tl] }
1044     }
1045   }
1046 }
1047 }
1048
1049 \keys_define:nn { hyp / setup }
1050 {
1051   ,allbordercolors .meta:n =
1052   {
1053     ,linkbordercolor=#1
1054     ,urlbordercolor =#1
1055     ,filebordercolor=#1
1056     ,menubordercolor=#1
1057     ,runbordercolor =#1
1058   }
1059   ,allbordercolors .value_required:n = true
1060 }
1061

```

11.3.2 Borderwidth and -arc

```

1062 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1063 {
1064   \keys_define:nn { hyp / setup }
1065   {
1066     #1border .code:n =
1067     {
1068       \tl_if_empty:nTF { ##1 }
1069       {
1070         \pdfannot_dict_remove:nn
1071         {link/#2}
1072         { Border }
1073       }
1074       {

```

```

1075         \pdfannot_dict_put:nnn
1076         {link/#2}
1077         { Border }
1078         { [##1] }
1079     }
1080 }
1081 }
1082 }
1083 \keys_define:nn { hyp / setup }
1084 {
1085     ,pdfborder .code:n =
1086     {
1087         \tl_if_empty:nTF { #1 }
1088         {
1089             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1090             {
1091                 \pdfannot_dict_remove:nn
1092                 {link/##2}
1093                 { Border }
1094             }
1095         }
1096         {
1097             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1098             {
1099                 \pdfannot_dict_put:nnn
1100                 {link/##2}
1101                 { Border }
1102                 { [#1] }
1103             }
1104         }
1105     }
1106     ,pdfborder .initial:n = {0~0~1},
1107 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1108 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1109 {
runborderstyle (setup key) 1110     \keys_define:nn { hyp / setup }
fileborderstyle (setup key) 1111     {
menuborderstyle (setup key) 1112         #1borderstyle .code:n =
1113         {
1114             \tl_if_empty:nTF { ##1 }
1115             {
1116                 \pdfannot_dict_remove:nn
1117                 {link/#2}
1118                 { BS }
1119             }
1120             {
1121                 \pdfannot_dict_put:nnn
1122                 {link/#2}
1123                 { BS }

```

```

1124         { <<#1>> }
1125     }
1126 }
1127 }
1128 }
1129 \keys_define:nn { hyp / setup }
1130 {
1131     ,pdfborderstyle .code:n =
1132     {
1133         \tl_if_empty:nTF { #1 }
1134         {
1135             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1136             {
1137                 \pdfannot_dict_remove:nn
1138                 {link/##2}
1139                 { BS }
1140             }
1141         }
1142         {
1143             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1144             {
1145                 \pdfannot_dict_put:nnn
1146                 {link/##2}
1147                 { BS }
1148                 { <<#1>> }
1149             }
1150         }
1151     }
1152     ,pdfborderstyle .initial:n = {},
1153 }

```

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.

```

1154 \cs_new_protected:Npn \__hyp_ocg_init:
1155 {
1156     \pdf_object_new:n { __hyp/OCG/View }
1157     \pdf_object_new:n { __hyp/OCG/Print }
1158     \pdf_object_new:n { __hyp/OCG/config }
1159     \pdf_object_new:n { __hyp/OCG/refarray }
1160     \pdf_object_write:nx { __hyp/OCG/refarray } { array }
1161     {
1162         \pdf_object_ref:n { __hyp/OCG/View }
1163         \c_space_tl
1164         \pdf_object_ref:n { __hyp/OCG/Print }
1165     }
1166     \pdf_object_write:nnn { __hyp/OCG/View } { dict }
1167     {
1168         /Type/OCG
1169         /Name(View)

```

```

1170         /Usage
1171         <<
1172         /Print <</PrintState/OFF>>~
1173         /View <</ViewState/ON >>~
1174         >>
1175     }
1176     \pdf_object_write:nnn { __hyp/OCG/Print } { dict }
1177     {
1178         /Type/OCG
1179         /Name(Print)
1180         /Usage
1181         <<
1182         /Print <</PrintState/ON>>~
1183         /View <</ViewState/OFF>>~
1184         >>
1185     }
1186     \pdfmanagement_add:nnx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/Print} }
1187     \pdfmanagement_add:nnx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/Print} }
1188     \pdf_object_write:nnx { __hyp/OCG/config } { dict }
1189     {
1190         /OFF[\pdf_object_ref:n { __hyp/OCG/Print }]
1191         /AS[
1192             <<
1193             /Event/View
1194             /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1195             /Category[/View]
1196             >>
1197             <<
1198             /Event/Print
1199             /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1200             /Category[/Print]
1201             >>
1202             <<
1203             /Event/Export
1204             /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1205             /Category[/Print]
1206             >>
1207             ]
1208         }
1209     \pdfmanagement_add:nnx { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG/Print} }
1210     \cs_gset:Npn \__hyp_ocg_init: {}
1211 }

```

(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.

```

1212 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1213 {
1214     \hook_gput_code:nnn
1215     {pdfannot/link/#2/begin}
1216     {hyp/ocg}
1217     {
1218         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1219         {

```

```

1220         \__hyp_ocg_init:
1221         \group_begin:
1222         \hbox_set:Nw \l__hyp_tmpa_box
1223     }
1224 }
1225 \hook_gput_code:nnn
1226 {pdfannot/link/#2/end}
1227 {hyp/ocg}
1228 {
1229     \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1230     {
1231         \hbox_set_end:
1232         \mbox
1233         {
1234             \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1235             \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1236             \pdf_emc:
1237             \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1238             \group_begin:
1239             \color_select:n { hyp/color/#1 }
1240             \box_use_drop:N \l__hyp_tmpa_box
1241             \group_end:
1242             \pdf_emc:
1243         }
1244         \group_end:
1245     }
1246 }
1247 }

```

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

```

1248 \bool_lazy_or:nnTF
1249 { \pdf_version_compare_p:Nn > {1.4} }
1250 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1251 {
1252     \keys_define:nn { hyp / setup }
1253     {
1254         ,_ocgcolorlinks .meta:n =
1255         {
1256             ocgcolorlink=#1,
1257             ocgcolorurl=#1,
1258             ocgcolorfile=#1,
1259             ocgcolorruncolor=#1,
1260             ocgcolormenu=#1
1261         }
1262         ,_ocgcolorlinks .default:n = true
1263     }
1264 }
1265 {
1266     \keys_define:nn { hyp / setup }
1267     {
1268         ,_ocgcolorlinks .code:n =
1269         {
1270             \msg_warning:nnxx
1271             { hyp }

```

```

1272         { ignore-deprecated-or-unknown-option-in-pdf-version }
1273         { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1274     }
1275 }
1276 }
1277
1278 \keys_define:nn { hyp / setup }
1279 {
1280     ,ocgcolorlinks .choice:
1281     ,ocgcolorlinks / true .meta:n =
1282     {
1283         pdfborder      = {0~0~0},
1284         pdfborderstyle = {},
1285         colorlinks     = false,
1286         _ocgcolorlinks = true
1287     }
1288     ,ocgcolorlinks / false .meta:n =
1289     {
1290         _ocgcolorlinks = false
1291     }
1292     ,ocgcolorlinks .default:n = {true}
1293 }
1294
1295 \seq_map_inline:Nn \c__hyp_annot_types_seq
1296 {
1297     \bool_lazy_or:nnTF
1298     { \pdf_version_compare_p:Nn > {1.4} }
1299     { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1300     {
1301         \keys_define:nn { hyp / setup }
1302         {
1303             ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1304         }
1305     }
1306     {
1307         \keys_define:nn { hyp / setup }
1308         {
1309             ,ocgcolor#1 .code:n=
1310             {
1311                 \msg_warning:nnxx
1312                 { hyp }
1313                 { ignore-deprecated-or-unknown-option-in-pdf-version }
1314                 { ocgcolor#1 }
1315                 { \pdf_version_major:.\pdf_version_minor: }
1316             }
1317         }
1318     }
1319 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1320 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1321 {

```

```

1322 \keys_define:nn { hyp / setup }
1323 {
1324   ,#1highlight .choices:nn =
1325   { /I, /N, /O, /P}
1326   {
1327     \pdfannot_dict_put:nnn
1328     {link/#2}
1329     { H }
1330     { ##1 }
1331   }
1332   ,#1highlight / .code:n =
1333   {
1334     \pdfannot_dict_remove:nn
1335     {link/#2}
1336     { H }
1337   }
1338   ,#1highlight / unknown .code:n =
1339   {
1340     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1341     { pdfhighlight }
1342     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1343     { \exp_not:n {##1} }
1344   }
1345 }
1346 }
1347 }
1348 }
1349
1350
1351 \keys_define:nn { hyp / setup }
1352 {
1353   ,pdfhighlight .choices:nn =
1354   { /I, /N, /O, /P}
1355   {
1356     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1357     {
1358       \pdfannot_dict_put:nnn
1359       {link/####2}
1360       { H }
1361       { #1 }
1362     }
1363   }
1364   ,pdfhighlight / .code:n =
1365   {
1366     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1367     {
1368       \pdfannot_dict_remove:nn
1369       {link/##2}
1370       { H }
1371     }
1372   }
1373   ,pdfhighlight .initial:n = {/I},
1374   ,pdfhighlight / unknown .code:n =
1375   {

```



```

1376         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1377         { pdfhighlight }
1378         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1379         { \exp_not:n {#1} }
1380     }
1381 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1382 \keys_define:nn { hyp / setup }
hideurl (setup key) 1383 {
hidefile (setup key) 1384     hidelinks .meta:n =
hiderun (setup key) 1385     {
hidemenu (setup key) 1386         ,colorlinks      = false
1387         ,ocgcolorlinks = false
1388         ,pdfborder      = { 0~0~0 }
1389         ,pdfborderstyle=
1390     }
1391 }
1392
1393 \seq_map_inline:Nn \c__hyp_annot_types_seq
1394 {
1395     \keys_define:nn { hyp / setup }
1396     {
1397         hide#1 .meta:n =
1398         {
1399             ,color#1      = false
1400             ,ocgcolor#1   = false
1401             ,#1border     = { 0~0~0 }
1402             ,#1borderstyle =
1403         }
1404     }
1405 }

```

11.7 color schemes and settings

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

```

colorscheme (setup key)
1406 \keys_define:nn { hyp / setup }
1407 {
1408     colorscheme .code:n =
1409     {
1410         \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1411         {
1412             \keys_set:nn { hyp /setup }
1413             {
1414                 ##1 = ##2
1415             }
1416         }
1417     }

```

```

1418 }
1419 \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) 1420 \keys_define:nn { hyp / setup }
pdfversion (setup key) 1421 {
1422   ,unicode .code:n = {}
1423   ,pdfencoding .code:n = {}
1424   ,pdfversion .code:n =
1425   {
1426     \msg_warning:nn { hyp }{ pdfversion-disabled }
1427   }
1428 }
1429 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
debug (setup key) 1430 \keys_define:nn { hyp / setup }
draft (setup key) 1431 {
1432   ,verbose .legacy_if_set:n = {Hy@verbose}
1433   ,debug .legacy_if_set:n = {Hy@verbose}
1434 }
1435 \keys_define:nn { hyp / setup }
1436 {
1437   ,draft .code:n =
1438   {
1439     \Hy@drafttrue
1440     \PassOptionsToPackage{draft}{bookmark}
1441   }
1442   ,final .code:n =
1443   {
1444     \Hy@finaltrue
1445     \PassOptionsToPackage{final}{bookmark}
1446   }
1447 }

extension (setup key)
hypertextnames (setup key) 1448 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1449 {
1450   ,extension .tl_set:N = \XR@ext
1451   ,extension .initial:n= pdf
1452   ,hypertextnames .legacy_if_set:n = {Hy@hypertextnames}
1453   ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1454   ,localanchorname .legacy_if_set:n = {Hy@localanchorname}
}

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

```

```

1455 ,naturalnames .legacy_if_set:n = {Hy@naturalnames}
1456 ,pageanchor .legacy_if_set:n = {Hy@pageanchor}
1457 ,plainpages .legacy_if_set:n = {Hy@plainpages}
1458 }
1459
1460 \keys_define:nn { hyp / setup }
1461 {
1462 ,linktoc .choices:nn = { none, section, all, page }
1463 {
1464 \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1465 }
1466 ,linktoc / unknown .code:n =
1467 {
1468 \msg_warning:nnxxx { hyp } { unknown-choice }
1469 { linktoc }
1470 { none, section, all, page }
1471 { \exp_not:n {#1} }
1472 }
1473 ,linktocpage .choice:
1474 ,linktocpage / true .meta:n = {linktoc=page}
1475 ,linktocpage / false .meta:n = {linktoc=section}
1476 ,linktocpage .default:n = true
1477 }
1478

```

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

`url (setup key)` 1479 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop

`file (setup key)` 1480 {

`menu (setup key)` 1481 \keys_define:nn { hyp / setup }

`run (setup key)` 1482 {

```

1483 ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1484 }
1485 }

```

```

1486 \keys_define:nn { hyp / setup }
1487 {

```

```

1488 ,baseurl .code:n =
1489 {

```

```

1490 \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1491 \tl_if_empty:NTF \l__hyp_tmpa_tl

```

```

1492 {

```

```

1493 \pdfmanagement_remove:nn {Catalog} { URI }
1494 }

```

```

1495 {

```

```

1496 \pdfmanagement_add:nnx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1497 }

```

```

1498 \__hyp_store_metadata:nn {baseurl}{#1}
1499 }

```

```

1500 %only false does something ...

```

```

1501 ,bookmarks .choice:

```

```

1502 ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1503 ,bookmarks / true .code:n = {}

```

```

1504 ,bookmarks .default:n = {true}

```

```

1505 ,bookmarksnumbered .legacy_if_set:n = {Hy@bookmarksnumbered}

```

```

1506 ,bookmarksopen .legacy_if_set:n = {Hy@bookmarksopen}
1507 ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1508 ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1509 ,pdfcenterwindow .choice:
1510 ,pdfcenterwindow / false .code:n =
1511 {
1512   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1513 }
1514 ,pdfcenterwindow / true .code:n =
1515 {
1516   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1517 }
1518 ,pdfcenterwindow / .code:n =
1519 {
1520   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1521 }
1522 ,pdfcenterwindow / unknown .code:n =
1523 {
1524   \msg_warning:nnxx { hyp } { no-bool }
1525   { pdfcenterwindow }
1526   { \exp_not:n {#1} }
1527 }
1528 ,pdfcenterwindow .default:n = true
1529 ,pdfdirection .choice:
1530 ,pdfdirection / L2R .code:n =
1531 {
1532   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1533 }
1534 ,pdfdirection / R2L .code:n =
1535 {
1536   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1537 }
1538 ,pdfdirection / .code:n =
1539 {
1540   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1541 }
1542 ,pdfdirection / unknown .code:n =
1543 {
1544   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1545   { pdfdirection }
1546   { L2R , R2L }
1547   { \exp_not:n {#1} }
1548 }
1549 ,pdfdisplaydoctitle .choice:
1550 ,pdfdisplaydoctitle / false .code:n =
1551 {
1552   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1553 }
1554 ,pdfdisplaydoctitle / true .code:n =
1555 {
1556   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1557 }
1558 ,pdfdisplaydoctitle .default:n = true
1559 ,pdfduplex .choices:nn =

```

```

1560 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1561 {
1562   \pdf_version_compare:NnTF > {1.6}
1563   {
1564     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1565     { PrintDuplex } { /#1 }
1566   }
1567   {
1568     \msg_warning:nnxx
1569     {hyp}
1570     {ignore-deprecated-or-unknown-option-in-pdf-version}
1571     {pdfduplex}
1572     {\pdf_version:}
1573   }
1574 }%
1575 ,pdfduplex / .code:n =
1576 {
1577   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1578 }
1579 ,pdfduplex / unknown .code:n =
1580 {
1581   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1582   { pdfduplex }
1583   { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1584   { \exp_not:n {#1} }
1585 }
1586 ,pdffitwindow .choice:
1587 ,pdffitwindow / false .code:n =
1588 {
1589   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1590 }
1591 ,pdffitwindow / true .code:n =
1592 {
1593   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1594 }
1595 ,pdffitwindow / .code:n =
1596 {
1597   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1598 }
1599 ,pdffitwindow .default:n = true
1600 ,pdffitwindow / unknown .code:n =
1601 {
1602   \msg_warning:nnxx { hyp } { no-bool }
1603   { pdffitwindow }
1604   { \exp_not:n {#1} }
1605 }
1606 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1607 ,pdflinkmargin .initial:n = {1pt}
1608 ,pdfmenubar .choice:
1609 ,pdfmenubar / true .code:n =
1610 {
1611   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1612 }
1613 ,pdfmenubar / false .code:n =

```

```

1614     {
1615         \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1616         { HideMenubar } { true }
1617     }
1618 ,pdfmenubar / .code:n =
1619     {
1620         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1621     }
1622 ,pdfmenubar .default:n = true
1623 ,pdfmenubar / unknown .code:n =
1624     {
1625         \msg_warning:nnxx { hyp } { no-bool }
1626         { pdfmenubar }
1627         { \exp_not:n {#1} }
1628     }
1629 ,pdfnewwindow .choice:
1630 ,pdfnewwindow / true .code:n =
1631     {
1632         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1633         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1634     }
1635 ,pdfnewwindow / false .code:n =
1636     {
1637         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1638         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1639     }
1640 ,pdfnewwindow / .code:n =
1641     {
1642         \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1643         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1644     }
1645 ,pdfnonfullscreenpagemode .choices:nn =
1646     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1647     {
1648         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1649         { NonFullScreenPageMode } {/#1}
1650     }
1651 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1652     {
1653         \pdf_version_compare:NnTF < {1.6}
1654         {
1655             %message
1656         }
1657         {
1658             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1659             { NonFullScreenPageMode } {/UseAttachments}
1660         }
1661     }
1662 ,pdfnonfullscreenpagemode / .code:n =
1663     {
1664         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1665     }
1666 ,pdfnonfullscreenpagemode / unknown .code:n =
1667     {

```

```

1668     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1669     { pdfnonfullscreenpagemode }
1670     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1671     { \exp_not:n {#1} }
1672 }
1673 ,pdfnumcopies .code:n =
1674 {
1675     \pdf_version_compare:NnTF > {1.6}
1676     {
1677         \tl_if_empty:nTF {#1}
1678         {
1679             \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { NumCopies }
1680         }
1681         {
1682             \pdfmanagement_add:nnx {Catalog / ViewerPreferences}
1683             {NumCopies}{#1}
1684         }
1685     }
1686     {
1687         \msg_warning:nnxx
1688         {hyp}
1689         {ignore-deprecated-or-unknown-option-in-pdf-version}
1690         {pdfnumcopies}
1691         {\pdf_version:}
1692     }
1693 }
1694 ,pdfpagelayout .choices:nn =
1695 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1696 { \pdfmanagement_add:nnx {Catalog} { PageLayout }{ /#1 } }
1697 ,pdfpagelayout / .code:n =
1698 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1699 ,pdfpagelayout / unknown .code:n =
1700 {
1701     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1702     { pdfpagelayout }
1703     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1704     { \exp_not:n {#1} }
1705 }
1706 ,pdfpagemode .choices:nn =
1707 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1708 { \pdfmanagement_add:nnx {Catalog} { PageMode }{ /#1 } }
1709 ,pdfpagemode / UseAttachments .code:n =
1710 {
1711     \pdf_version_compare:NnTF > {1.5}
1712     {
1713         \pdfmanagement_add:nnx {Catalog} { PageMode }{ /UseAttachments }
1714     }
1715     {
1716         \msg_warning:nnxx
1717         {hyp}
1718         {ignore-deprecated-or-unknown-value-in-pdf-version}
1719         {UseAttachments}
1720         {\pdf_version:}
1721     }

```

```

1722     }
1723     ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1724     ,pdfpagemode / unknown .code:n =
1725     {
1726         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1727         { pdfpagemode }
1728         { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1729         { \exp_not:n {#1} }
1730     }
1731     ,pdfpagescrop .code:n =
1732     {
1733         \tl_if_empty:nTF {#1} %or blank?
1734         {
1735             \pdfmanagement_remove:nn {Pages} { CropBox }
1736         }
1737         {
1738             \pdfmanagement_add:nnx {Pages} { CropBox } { [#1] }
1739         }
1740     }
1741     ,pdfpicktraybypdfsize .choice:
1742     ,pdfpicktraybypdfsize / true .code:n =
1743     {
1744         \pdf_version_compare:NnTF > {1.6}
1745         {
1746             \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1747             { PickTrayByPDFSize } { true }
1748         }
1749         {
1750             \msg_warning:nnxx
1751             {hyp}
1752             {ignore-deprecated-or-unknown-option-in-pdf-version}
1753             {pdfpicktraybypdfsize}
1754             {\pdf_version:}
1755         }
1756     }
1757     ,pdfpicktraybypdfsize / false .code:n =
1758     {
1759         \pdf_version_compare:NnTF > {1.6}
1760         {
1761             \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1762             { PickTrayByPDFSize } { false }
1763         }
1764         {
1765             \msg_warning:nnxx
1766             {hyp}
1767             {ignore-deprecated-or-unknown-option-in-pdf-version}
1768             {pdfpicktraybypdfsize}
1769             {\pdf_version:}
1770         }
1771     }
1772     ,pdfpicktraybypdfsize / .code:n =
1773     {
1774         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1775     }

```



```

1776 ,pdfpicktraybypdfsize / unknown .code:n =
1777 {
1778     \msg_warning:nnxx { hyp } { no-bool }
1779     { picktraybypdfsize }
1780     { \exp_not:n {#1} }
1781 }
1782 ,pdfprintarea .choices:nn =
1783 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1784 {
1785     \pdf_version_compare:NnTF < {2.0}
1786     {
1787         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1788         { PrintArea } { /#1 }
1789     }
1790     {
1791         \msg_warning:nnxx
1792         {hyp}
1793         {ignore-deprecated-or-unknown-option-in-pdf-version}
1794         {pdfprintarea}
1795         {\pdf_version:}
1796     }
1797 }%
1798 ,pdfprintarea / .code:n =
1799 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1800 ,pdfprintarea / unknown .code:n =
1801 {
1802     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1803     { pdfprintarea }
1804     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1805     { \exp_not:n {#1} }
1806 }
1807 ,pdfprintclip .choices:nn =
1808 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1809 {
1810     \pdf_version_compare:NnTF < {2.0}
1811     {
1812         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1813         { PrintClip } { /#1 }
1814     }
1815     {
1816         \msg_warning:nnxx
1817         {hyp}
1818         {ignore-deprecated-or-unknown-option-in-pdf-version}
1819         {pdfprintclip}
1820         {\pdf_version:}
1821     }
1822 }%
1823 ,pdfprintclip / .code:n =
1824 {
1825     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1826 }
1827 ,pdfprintclip / unknown .code:n =
1828 {
1829     \msg_warning:nnxxx { hyp } { unknown-choice+empty }

```

```

1830     { pdfprintclip }
1831     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1832     { \exp_not:n {#1} }
1833   }
1834 ,pdfprintpagerange .code:n =
1835   {
1836     \pdf_version_compare:NnTF > {1.6}
1837     {
1838       \tl_if_empty:nTF { #1}
1839       {
1840         \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1841         { PrintPageRange }
1842       }
1843       {
1844         \pdfmanagement_add:nxx {Catalog / ViewerPreferences }
1845         {PrintPageRange}{[#1]}
1846       }
1847     }
1848     {
1849       \msg_warning:nxxx
1850       {hyp}
1851       {ignore-deprecated-or-unknown-option-in-pdf-version}
1852       {pdfprintpagerange}
1853       {\pdf_version:}
1854     }
1855   }
1856 ,pdfprintscaling .choices:nn =
1857   { None, AppDefault }
1858   {
1859     \pdf_version_compare:NnTF > {1.5}
1860     {
1861       \pdfmanagement_add:nxx {Catalog / ViewerPreferences }
1862       { PrintScaling } { /#1 }
1863     }
1864     {
1865       \msg_warning:nxxx
1866       {hyp}
1867       {ignore-deprecated-or-unknown-option-in-pdf-version}
1868       {pdfprintscaling}
1869       {\pdf_version:}
1870     }
1871   }%
1872 ,pdfprintscaling / .code:n =
1873   {
1874     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1875   }
1876 ,pdfprintscaling / unknown .code:n =
1877   {
1878     \msg_warning:nxxx { hyp } { unknown-choice+empty }
1879     { pdfprintarea }
1880     { None, AppDefault }
1881     { \exp_not:n {#1} }
1882   }
1883 ,pdfremotestartview .code:n =

```

```

1884 {
1885     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1886     \exp_args:NNV
1887     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1888     {
1889         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1890     }
1891     {
1892         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1893         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1894     }
1895 }
1896 ,pdfremotestartview .initial:n = {Fit}
1897 % pdfstartpage is special as it shares code with pdfstartview
1898 ,pdfstartpage .code:n =
1899 {
1900     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1901     \bool_if:nTF
1902     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1903     {
1904         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1905     }
1906     {
1907         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1908         {
1909             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1910         }
1911     }
1912 }
1913 ,pdfstartpage .initial:n =1
1914 ,pdfstartview .code:n =
1915 {
1916     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1917     \exp_args:NNV
1918     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1919     {
1920         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1921     }
1922     {
1923         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1924         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1925     }
1926     \bool_if:nTF
1927     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1928     {
1929         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1930     }
1931     {
1932         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1933         {
1934             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1935         }
1936     }
1937 }

```

```

1938 ,pdfstartview .initial:n = Fit
1939 ,pdftoolbar .choice:
1940 ,pdftoolbar / true .code:n =
1941 {
1942   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1943 }
1944 ,pdftoolbar / false .code:n =
1945 {
1946   \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1947   { HideToolbar } { true }
1948 }
1949 ,pdftoolbar / true .code:n =
1950 {
1951   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1952 }
1953 ,pdftoolbar .default:n = true
1954 ,pdftoolbar / unknown .code:n =
1955 {
1956   \msg_warning:nnxx { hyp } { no-bool }
1957   { pdftoolbar }
1958   { \exp_not:n {#1} }
1959 }
1960 % pdfview see below.
1961 ,pdfviewarea .choices:nn =
1962 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1963 {
1964   \pdf_version_compare:NnTF < {2.0}
1965   {
1966     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1967     { ViewArea } { /#1 }
1968   }
1969   {
1970     \msg_warning:nnxx
1971     {hyp}
1972     {ignore-deprecated-or-unknown-option-in-pdf-version}
1973     {pdfviewarea}
1974     {\pdf_version:}
1975   }
1976 }%
1977 ,pdfviewarea / .code:n =
1978 {
1979   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1980 }
1981 ,pdfviewarea / unknown .code:n =
1982 {
1983   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1984   { pdfviewarea }
1985   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1986   { \exp_not:n {#1} }
1987 }
1988 ,pdfviewclip .choices:nn =
1989 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1990 {
1991   \pdf_version_compare:NnTF < {2.0}

```

```

1992     {
1993         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1994         { ViewClip } { /#1 }
1995     }
1996     {
1997         \msg_warning:nnxx
1998         {hyp}
1999         {ignore-deprecated-or-unknown-option-in-pdf-version}
2000         {pdfviewclip}
2001         {\pdf_version:}
2002     }
2003     }%
2004     ,pdfviewclip / .code:n =
2005     {
2006         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
2007     }
2008     ,pdfviewclip / unknown .code:n =
2009     {
2010         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
2011         { pdfviewclip }
2012         { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2013         { \exp_not:n {#1} }
2014     }
2015     ,pdfwindowui .choice:
2016     ,pdfwindowui / true .code:n =
2017     {
2018         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2019     }
2020     ,pdfwindowui / false .code:n =
2021     {
2022         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2023         { HideWindowUI } { true }
2024     }
2025     ,pdfwindowui / .code:n =
2026     {
2027         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2028     }
2029     ,pdfwindowui / unknown .code:n =
2030     {
2031         \msg_warning:nnxx { hyp } { no-bool }
2032         { pdfwindowui }
2033         { \exp_not:n {#1} }
2034     }
2035     ,pdfwindowui .default:n = true
2036 }

```

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

```

2037 \keys_define:nn { hyp / setup }
2038 {
2039     ,pdfview .code:n =
2040     {
2041         \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
2042         \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } } {
2043             {

```

```

2044 { xyz }
2045 {
2046   \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
2047   {
2048     \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2049     \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2050     {
2051       \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2052     }
2053     {
2054       \tl_set:Nx \l__hyp_dest_pdfview_tl
2055       {
2056         \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2057       }
2058     }
2059   }
2060   {
2061     \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2062   }
2063 }
2064 { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2065 { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2066 { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2067 { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2068 { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2069 { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2070 { fitr }
2071 {
2072   \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2073   {
2074     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2075   }
2076   {
2077     %ensure 4 values ...
2078     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2079     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2080     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2081     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2082     \hbox_set_to_wd:Nnn \l__hyp_dest_box
2083     {
2084       \fp_eval:n
2085       {
2086         round
2087         (
2088           abs
2089           (
2090             \seq_item:Nn\l__hyp_tmpa_seq{4}
2091             -
2092             (\seq_item:Nn\l__hyp_tmpa_seq{2})
2093           ),
2094           3
2095         )
2096       }bp
2097     }{}

```

```

2098         \box_set_dp:Nn \l__hyp_dest_box
2099         {
2100             \fp_eval:n
2101             {
2102                 round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2103             }bp
2104         }
2105         \box_set_ht:Nn \l__hyp_dest_box
2106         {
2107             \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2108         }
2109     }
2110 }
2111 }
2112 {
2113     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2114     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2115 }
2116 }
2117 ,pdfview .initial:n = {xyz}
2118 }

```

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.

```

2119 \keys_define:nn { hyp / setup }
2120 {
2121     ,pdflang .code:n =
2122     {
2123         \tl_if_empty:nF { #1 }
2124         {
2125             \pdfmanagement_add:nnx {Catalog} { Lang } { (#1) }
2126             \AddToDocumentProperties[document]{lang}{#1}
2127         }
2128     }
2129 }

```

12.3.1 “info keys”

pdfauthor (*setup key*) The keys store their value also in the metadata container, so that hyperxmp can use them.
pdftitle (*setup key*) Creator and Producer can’t be removed with the pdfmanagement, but we allow to set an empty value. If the value begin with an optional argument, we assume a multilanguage
pdfcreator (*setup key*) empty value. If the value begin with an optional argument, we assume a multilanguage
pdfsubject (*setup key*) clist and use only the first value.

```

pdfproducer (setup key) 2130 \regex_new:N\l__hyp_optlang_regex
pdfkeywords (setup key) 2131 \regex_set:Nn\l__hyp_optlang_regex {\A\[([A-Za-z\-\-]+)\](.*)}
2132 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2133 {
2134     \keys_define:nn { hyp / setup }
2135     {
2136         pdf#1 .code:n =

```

```

2137 {
2138     \tl_if_blank:nTF {##1}
2139     {
2140         \str_case:nnF { #1 }
2141         {
2142             {creator}
2143             {
2144                 \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2145                 \pdfmanagement_add:nnx {Info}{Creator}{()}
2146             }
2147             {producer}
2148             {
2149                 \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2150                 \pdfmanagement_add:nnx {Info}{Producer}{()}
2151             }
2152         }
2153         {
2154             \pdfmanagement_remove:nn {Info}{#2}
2155         }
2156     }
2157     {
2158         \tl_set:Nx\l__hyp_tmpa_tl {\clist_item:nn{##1}{1}}
2159         \exp_args:NNV
2160         \regex_extract_once:NnN \l__hyp_optlang_regex \l__hyp_tmpa_tl\l__hyp_tmpa_seq
2161         \seq_if_empty:NTF\l__hyp_tmpa_seq
2162         {
2163             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2164         }
2165         {
2166             \exp_args:Nx
2167             \__hyp_text_pdfstring_info:nN {\seq_item:Nn \l__hyp_tmpa_seq{3}}\l__hyp_t
2168         }
2169         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2170         {
2171             \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2172         }
2173     }
2174     \__hyp_store_metadata:nn {pdf#1}{##1}
2175 }
2176 }
2177 \keys_define:nn { hyp / info }
2178 {
2179     #2 .code:n =
2180     {
2181         \tl_if_blank:nTF {##1}
2182         {
2183             \pdfmanagement_remove:nn {Info}{#2}
2184         }
2185         {
2186             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2187             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2188             {
2189                 \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2190             }
2191         }
2192     }
2193 }

```



```

2191     }
2192     \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2193   }
2194   ,unknown .code:n =
2195   {
2196     \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2197     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2198     {
2199       \exp_args:Nno
2200       \pdfmanagement_add:nnx {Info}
2201       { \l_keys_key_str } {\l__hyp_tmpa_str}
2202     }
2203   }
2204 }
2205 }
2206 \__hyp_setup_info_key:nn {author} {Author}
2207 \__hyp_setup_info_key:nn {title} {Title}
2208 \__hyp_setup_info_key:nn {producer} {Producer}
2209 \__hyp_setup_info_key:nn {creator} {Creator}
2210 % ignored key: addtopdfcreator
2211 \__hyp_setup_info_key:nn {subject} {Subject}
2212 \__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)
2213 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2214 {
2215   \keys_define:nn { hyp / setup }
2216   {
2217     pdf#1 .code:n =
2218     {
2219       \tl_if_blank:nTF {##1}
2220       {
2221         \pdfmanagement_remove:nn {Info}{#2}
2222       }
2223       {
2224         \pdfmanagement_add:nnx {Info}{#2}{(##1)}
2225       }
2226       \__hyp_store_metadata:nn {pdf#1}{##1}
2227       \AddToDocumentProperties[document]{#1}{##1}
2228     }
2229   }
2230   \keys_define:nn { hyp / info }
2231   {
2232     #2 .code:n =
2233     {
2234       \tl_if_blank:nTF {##1}
2235       {
2236         \pdfmanagement_remove:nn {Info}{#2}
2237       }
2238       {
2239         \pdfmanagement_add:nnx {Info}{#2}{(##1)}
2240       }
2241       \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}

```

```

2242     }
2243   }
2244 }
2245
2246 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2247 \__hyp_setup_info_date_key:nn {moddate} {ModDate}
2248 \keys_define:nn { hyp / setup }
2249 {
2250     pdfmetadate .code:n = { \__hyp_store_metadata:nn {pdfmetadate}{#1} }
2251 }

```

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

```

2252 \keys_define:nn { hyp / setup }
2253 {
2254     ,pdftrapped .code:n =
2255     {
2256         \exp_args:Nne
2257         \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2258     }
2259     ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2260     {
2261         \pdfmanagement_add:nnx {Info}{Trapped}
2262         {/
2263             \str_uppercase:f { \str_head:n { #1 } }
2264             \str_lowercase:f { \str_tail:n { #1 } }
2265         }
2266         \__hyp_store_metadata:nx {pdftrapped}
2267         {
2268             \str_uppercase:f { \str_head:n { #1 } }
2269             \str_lowercase:f { \str_tail:n { #1 } }
2270         }
2271     }
2272     ,_pdftrapped / unknown .code:n =
2273     {
2274         \msg_warning:nnxxx { hyp } { unknown-choice }
2275         { pdftrapped }
2276         { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2277         { \exp_not:n {#1} }
2278     }
2279 }

```

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

```

2280 \keys_define:nn { hyp / setup }
2281 {
2282     pdfinfo .code:n =
2283     {
2284         \keys_set:nn { hyp / info } { #1 }
2285     }
2286 }

```

Now we set some default values

```

2287 \keys_set:nn { hyp / setup } {pdfcreator = LaTeX-with-hyperref}
2288 \keys_set:nn { hyp / setup } {pdfauthor = }
2289 \keys_set:nn { hyp / setup } {pdftitle = }
2290 \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:

```
2291 \clist_map_inline:nn
2292 {
2293   ,pdfcopyright
2294   ,pdftype
2295   ,pdflicenseurl
2296   ,pdfauthortitle
2297   ,pdfcaptionwriter
2298   ,pdfmetalang
2299   ,pdfsource
2300   ,pdfdocumentid
2301   ,pdfinstanceid
2302   ,pdfversionid
2303   ,pdfrendition
2304   ,pdfpublication
2305   ,pdfpubtype
2306   ,pdfbytes
2307   ,pdfnumpages
2308   ,pdfissn
2309   ,pdfeissn
2310   ,pdfisbn
2311   ,pdfbookedition
2312   ,pdfpublisher
2313   ,pdfvolumenum
2314   ,pdfissuenum
2315   ,pdfpagerange
2316   ,pdfdoi
2317   ,pdfurl
2318   ,pdfidentifier
2319   ,pdfsubtitle
2320   ,pdfpubstatus
2321   ,pdfcontactaddress
2322   ,pdfcontactcity
2323   ,pdfcontactregion
2324   ,pdfcontactpostcode
2325   ,pdfcontactcountry
2326   ,pdfcontactphone
2327   ,pdfcontactemail
2328   ,pdfcontacturl
2329   ,pdfdate
2330 }
2331 {
2332   \keys_define:nn { hyp / setup }
2333   {
2334     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2335   }
2336 }
2337
```

12.5 Transitions

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

```

2338 \keys_define:nn { hyp / setup }
2339 {
2340   pdfpageduration .code:n =
2341   {
2342     \tl_if_blank:nTF { #1 }
2343     {
2344       \pdfmanagement_remove:nn {Page}{Dur}
2345     }
2346     {
2347       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2348     }
2349   }
2350 }

```

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

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

The keys allowed in the argument of `pdfpagetransition` are

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

```

2351 \keys_define:nn { hyp / setup }
2352 {
2353   pdfpagetransition .code:n =
2354   {
2355     \tl_if_blank:nTF {#1}
2356     {
2357       \pdfmanagement_remove:nn {Page}{Trans}
2358     }

```

```

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

```

```

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

```

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

```

2439 \keys_set:known:nv{hyp/setup}{opt@hyperref.sty}

```

Unfinished Form field code

```

2440 \NewDocumentCommand \MakeFieldObject { m m }
2441 {
2442   \pdfxform_new:nnn { #2 }{ } { #1 }
2443 }
2444
2445
2446 \prop_new:N \g__hyp_AcroForm_CoFields_prop
2447 \prop_new:N \g__hyp_AcroForm_Fields_prop
2448
2449 \let\HyField@afields\ltx@empty
2450 \let\HyField@cofields\ltx@empty
2451 %% UF test for old pdftex removed
2452 %\let\HyField@AuxAddToFields\ltx@gobble
2453 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2454 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2455
2456 % the value doesn't matter, but with a prop we avoid duplicates and it is
2457 % clearly faster than removing them from a sequence
2458 \def\HyField@AuxAddToFields#1
2459 {
2460   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2461 }%
2462
2463 %fields with empty key get a value too -- lets hope that

```

```

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

```

```

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

```



```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```



```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```



```

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

```

```

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

```

```

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

```

```

3598 }{}
3599 \expandafter\def\csname Parent-4\endcsname{}
3600 \expandafter\def\csname Parent-3\endcsname{}
3601 \expandafter\def\csname Parent-2\endcsname{}
3602 \expandafter\def\csname Parent-1\endcsname{}
3603 \expandafter\def\csname Parent0\endcsname{}
3604 \expandafter\def\csname Parent1\endcsname{}
3605 \expandafter\def\csname Parent2\endcsname{}
3606 \expandafter\def\csname Parent3\endcsname{}
3607 \expandafter\def\csname Parent4\endcsname{}
3608 %%
3609 %% End of file 'hgeneric-testphase.def'.
3610 \</package>

3611 \<colorscheme>
3612 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3613 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3614 % linkcolor=
3615 %,filecolor=
3616 %,urlcolor=
3617 %,menucolor=
3618 %,runcolor=
3619 %,linkbordercolor=
3620 %,filebordercolor=
3621 %,urlbordercolor=
3622 %,menubordercolor=
3623 %,runbordercolor=
3624
3625 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }
3626 {
3627     linkcolor      = [rgb]{1,0,0}, %red
3628     filecolor      = [rgb]{0,1,1}, %cyan
3629     urlcolor       = [rgb]{1,0,1}, %magenta
3630     menucolor      = [rgb]{1, 0, 0}, %red
3631     runcolor       = [rgb]{0,1,1}, %cyan
3632     %-----
3633     linkbordercolor = [rgb]{1, 0 ,0 },
3634     filebordercolor = [rgb]{0, .5, .5},
3635     urlbordercolor  = [rgb]{0, 1, 1},
3636     menubordercolor = [rgb]{1, 0, 0},
3637     runbordercolor  = [rgb]{0, .7, .7}
3638 }
3639
3640 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3641 {
3642     linkcolor      = [rgb]{0,0.2,0.6},
3643     filecolor      = [rgb]{0.8,0,0.8},
3644     urlcolor       = [rgb]{0.8,0,0.8},
3645     menucolor      = [rgb]{0,0.2,0.6},
3646     runcolor       = [rgb]{0.8,0,0.8},
3647     %----- %-----
3648     linkbordercolor = [rgb]{0,0.2,0.6},
3649     filebordercolor = [rgb]{0.8,0,0.8},
3650     urlbordercolor  = [rgb]{0.8,0,0.8},
3651     menubordercolor = [rgb]{0,0.2,0.6},

```

```

3652     runbordercolor = [rgb]{0.8,0,0.8}
3653 }
3654
3655 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3656 { %two colors: intern/extern
3657     linkcolor      = [rgb]{0.79216, 0, 0.12549},
3658     filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3659     urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3660     menucolor      = [rgb]{0.79216, 0, 0.12549 },
3661     runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3662     %----- %-----
3663     linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3664     filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3665     urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3666     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3667     runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3668 }
3669
3670 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3671 { %all darkgray
3672     linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3673     filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3674     urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3675     menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3676     runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3677     %----- %-----
3678     linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3679     filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3680     urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3681     menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3682     runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3683 }
3684
3685 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3686 { %dvipsnam.def
3687     linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3688     filecolor      = [rgb]{1, 0, 0}, %Red
3689     urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3690     menucolor      = [rgb]{1, 0, 0}, %Red
3691     runcolor       = [rgb]{1, 0, 0}, %Red
3692     %----- %-----
3693     linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3694     filebordercolor = [rgb]{1, 0, 0}, %Red
3695     urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3696     menubordercolor = [rgb]{1, 0, 0}, %Red
3697     runbordercolor  = [rgb]{1, 0, 0} %Red
3698 }
3699
3700 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3701 { %dvipsnam.def
3702     linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3703     filecolor      = [rgb]{0, 1, 0}, %Green
3704     urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3705     menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue

```

```

3706     runcolor          = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3707 %----- %-----
3708     linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3709     filebordercolor = [rgb]{0, 1, 0}, %Green
3710     urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3711     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3712     runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3713 }
3714
3715
3716 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3717 {
3718     linkcolor          = [rgb]{0.50196, 0, 0.02353},
3719     filecolor          = [rgb]{0.07451, 0.09412, 0.46667},
3720     urlcolor           = [rgb]{0.54118, 0, 0.52941},
3721     menucolor          = [rgb]{0.44706, 0.45882, 0},
3722     runcolor           = [rgb]{0.07451, 0.46667, 0.46275},
3723 %----- %-----
3724     linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3725     filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3726     urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3727     menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3728     runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3729 }
3730
3731 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3732 {
3733     linkcolor          = [rgb]{0,0,0},
3734     filecolor          = [rgb]{0,0,0},
3735     urlcolor           = [rgb]{0,0,0},
3736     menucolor          = [rgb]{0,0,0},
3737     runcolor           = [rgb]{0,0,0},
3738 %----- %-----
3739     linkbordercolor = [rgb]{0,0,0},
3740     filebordercolor = [rgb]{0,0,0},
3741     urlbordercolor  = [rgb]{0,0,0},
3742     menubordercolor = [rgb]{0,0,0},
3743     runbordercolor  = [rgb]{0,0,0}
3744 }
3745 </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
\-	2131
\.	561, 565, 567
@curropt commands:	
\@curropt:	2652, 2719
\[2131
\]	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, 881, 2654, 2721, 3061
_	559, 561, 565, 567
\]	2131
A	
\A	559, 2131
\Acrobatmenu	18, 173
\addcontentsline	13
\AddToDocumentProperties	388, 2126, 2227
\AddToHook	426, 439
\AddToHookNext	198
\advance	2585, 2659, 2703, 2704
allcolors (hypersetup key)	994
\author	2
B	
\b	563
\beginngroup	209, 277, 375, 2491, 2619, 2648, 2660, 2694, 3083, 3514, 3526, 3555, 3585
\belowpdfbookmark	3476
\bgroup	274, 375
\BOOKMARK	3479
bookmarkstype (hypersetup key)	13
bool commands:	
\bool_if:NTF	267, 296, 355, 698, 714, 726, 739, 747, 777, 835, 892, 964, 974, 1218, 1229
\bool_if:nTF	847, 1901, 1926, 3220, 3328
\bool_lazy_and:nnTF	423
\bool_lazy_or:nnTF	1248, 1297
\bool_new:N	203, 204, 545, 549, 553
\bool_set_true:N	554
bordercolormodel (hypersetup key)	13, 1015
box commands:	
\box_dp:N	635, 2804, 2855, 2890
\box_ht:N	634, 2803, 2854, 2889
\box_new:N	494, 556
\box_set_dp:Nn	2098
\box_set_ht:Nn	2105
\box_use:N	1235, 2806, 2864, 2893
\box_use_drop:N	1240
\box_wd:N	633, 2802, 2853, 2888
\l_tmpa_box	2800, 2802, 2803, 2804, 2806, 2839, 2853, 2854, 2855, 2864, 2886, 2888, 2889, 2890, 2893
C	
\catcode	278, 279
\char	2738, 2935, 2947
\chardef	158
\cite	33
clist commands:	
\clist_item:nn	2158
\clist_map_function:nN	124, 133
\clist_map_inline:nn	2291
\clist_map_variable:nNn	2651, 2718
color commands:	
\color_export:nnN	39, 431, 923, 1036
\color_select:n	931, 967, 1239, 2845, 2849
\color_select:nn	937
\color_set:nn	2, 393, 948
\color_set:nnn	2, 392, 954
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)	994
colorlink (hypersetup key)	994
colorlinks (hypersetup key)	980
colormenu (hypersetup key)	994
colorrun (hypersetup key)	994
colorscheme (hypersetup key)	1, 1406
colorurl (hypersetup key)	994
cs commands:	
\cs_generate_variant:Nn	155, 156, 157, 390, 490, 492, 493, 597, 923, 940, 957
\cs_gset:Npn	1210
\cs_gset_eq:NN	409, 420
\cs_if_exist:NTF	12, 449

\ExplSyntaxOff		\hbox_set:Nw	1222
..	9, 2522, 2575, 2602, 2960, 3028,		\hbox_set_end:	1231
	3144, 3208, 3243, 3315, 3366, 3465		\hbox_set_to_wd:Nnn	2082
\ExplSyntaxOn		hidefile (hypersetup key)	1382
...	7, 10, 2523, 2580, 2603, 2963,		hidelink (hypersetup key)	1382
	3135, 3156, 3209, 3278, 3316, 3402		hidelinks (hypersetup key)	1382
extension (hypersetup key)	13, 1448	hidemenu (hypersetup key)	1382
F					
\fbox	2757, 2845, 2849	hiderun (hypersetup key)	1382
\fi	2479,	hideurl (hypersetup key)	1382
	2506, 2509, 2518, 2617, 2625, 2626,		hook commands:		
	2657, 2675, 2678, 2679, 2685, 2688,		\hook_gput_code:nnn	
	2697, 2699, 2706, 2709, 3036, 3058,		402, 413, 960, 970, 1214, 1225	
	3068, 3070, 3076, 3080, 3087, 3088,		\hook_new:n	391, 572, 647
	3097, 3101, 3105, 3109, 3113, 3117,		\hook_new_pair:nn	
	3121, 3125, 3129, 3133, 3150, 3154,		260, 288, 312, 329, 347	
	3253, 3254, 3255, 3262, 3266, 3270,		\hook_use:n	
	3272, 3274, 3293, 3297, 3301, 3305,		. 264, 284, 293, 309, 316, 326, 333,		
	3312, 3377, 3378, 3385, 3389, 3391,		343, 352, 371, 589, 653, 662, 703, 719		
	3412, 3413, 3420, 3424, 3426, 3491,		\href	4, 5, 19, 262
	3496, 3548, 3549, 3560, 3564, 3581		\hreflaunch	5
file (hypersetup key)	10, 1479	\hrefpdf	5, 19, 313, 383
file commands:			\hrefrun	5, 330, 384
\file_input:n	8	\hrefurl	5, 19, 290, 382
fileborderstyle (hypersetup key)	14, 1108		\hss	3505
filecolor (hypersetup key)	994	Hy internal commands:		
final (hypersetup key)	1430	\Hy_hyescapeform	..	3047, 3060, 3066
\fontencoding	2734, 2931, 2943	\Hy_hypSectionAnchor	...	3495, 3498
\fontfamily	2733, 2930, 2942	HyField internal commands:		
\fontseries	2735, 2932, 2944	\HyField_hypAddToFields	..	2485, 2512
\fontshape	2736, 2933, 2945	hyp commands:		
fp commands:			\l_hyp_annot_colorfile_bool	...	543
\fp_eval:n	2056, 2084, 2100	\l_hyp_annot_colorlink_bool	...	543
G					
\gdef	280, 281	\l_hyp_annot_colormenu_bool	...	543
\GetDocumentProperties	3	\l_hyp_annot_colorrun_bool	...	543
\global	2585, 3026	\l_hyp_annot_colorurl_bool	...	543
group commands:			\l_hyp_annot_ocgcolorfile_bool	..	547
\group_begin:	..	265, 294, 317, 334,	\l_hyp_annot_ocgcolorlink_bool	..	547
	353, 587, 701, 717, 741, 779, 837,		\l_hyp_annot_ocgcolormenu_bool	..	547
	894, 966, 1221, 1238, 2360, 2732,		\l_hyp_annot_ocgcolorrun_bool	..	547
	2782, 2825, 2873, 2906, 2929, 2941		\l_hyp_annot_ocgcolorurl_bool	..	547
\group_end:	308, 325, 342,	\l_hyp_current_dest_name_tl	
	370, 594, 706, 729, 765, 825, 875,		646, 702, 718	
	918, 976, 1241, 1244, 2367, 2739,		hyp internal commands:		
	2818, 2866, 2899, 2936, 2948, 2958		\g__hyp_AcroForm_CoFields_prop	..	
H			2446, 2467, 2536, 2538, 2553	
\hbox	375, 3500	\g__hyp_AcroForm_Fields_prop	...	
hbox commands:			2447, 2460, 2531	
\hbox_overlap_right:n	1235	\l__hyp_annot_GoTo_bool	698, 714, 726	
\hbox_set:Nn	2800, 2839, 2886	\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	551
			\l__hyp_annot_GoToR_bool	777

\l__hyp_annot_Launch_bool	835	\l__hyp_href_pdf_page_tl 207, 225, 805	
\l__hyp_annot_Named_bool	892	__hyp_href_run_aux:nn	336, 339
\c__hyp_annot_types_seq		\l__hyp_href_run_parameter_tl	
.	506, 543, 547, 994, 1295, 1393	208, 229, 341
\l__hyp_annot_URI_bool	739	__hyp_href_url_aux:n	362, 365
\g__hyp_bordercolormodel_str		__hyp_href_url_aux:nn	303, 305
.	433, 542, 1019, 1038	\l__hyp_href_url_encode_bool	
__hyp_citebordercolor_hook_-		203, 214, 252, 267, 296, 355
init:	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, 924, 924, 940		\l__hyp_href_url_protocol_tl	
__hyp_color_select_aux:wn	205, 216, 255, 307, 369
.	924, 928, 935	__hyp_link_goto_begin:nw	
__hyp_color_set:nn	39,	669, 704, 720
396, 398, 430, 941, 941, 957, 999, 1035		__hyp_link_goto_end:	691, 705, 728
__hyp_color_set_aux:nwn 941, 945, 952		\c__hyp_map_annot_hyp_prop	506
\c__hyp_colorscheme_daleif_prop 3640		\c__hyp_map_hyp_annot_prop	
\c__hyp_colorscheme_henryford_-		506, 958,
prop	3731	1022, 1062, 1089, 1097, 1108, 1135,	
\c__hyp_colorscheme_julian_prop 3655		1143, 1212, 1320, 1356, 1366, 1479	
\c__hyp_colorscheme_phelype_prop		__hyp_ocg_init: 1154, 1154, 1210, 1220	
.	3716	\l__hyp_optlang_regex 2130, 2131, 2160	
\c__hyp_colorscheme_szabolcsA_-		l__hyp_page/Trans	570
prop	3685	__hyp_PageLabels_gpush: 604, 604, 617	
\c__hyp_colorscheme_szabolcsB_-		\l__hyp_para_tmpa_tl 502, 846, 849, 858	
prop	3700	\l__hyp_para_tmpa_tl_uuu\l__hyp_-	
\c__hyp_colorscheme_tivv_prop	3670	text_tmpa_str_uuu\g__hyp_text_-	
\l__hyp_dest_box	32,	tmpa_str	499
556, 633, 634, 635, 2082, 2098, 2105		__hyp_property_if_recorded:nn	
\l__hyp_dest_name_tmpa_tl	499,	448, 458, 470, 491
672, 673, 679, 683, 685, 688, 798, 811		__hyp_property_if_recorded:nnTF	
\l__hyp_dest_pdfremotestartview_-		474, 2594
tl	525, 806, 1889, 1893	__hyp_property_record:nn	
\g__hyp_dest_pdfstartpage_tl	448, 451, 463, 490, 2589
.	525, 1900, 1902, 1909, 1927, 1934	__hyp_property_ref:nn	
\g__hyp_dest_pdfstartview_tl	448, 460, 485, 493, 2598
.	525, 1902, 1909, 1920, 1924, 1927, 1934	__hyp_property_ref_undefined_-	
\l__hyp_dest_pdfview_tl		warn:nn	448, 459, 472, 492, 2590
.	541, 651, 660,	__hyp_secondoftwowithopt:wnn	
2051, 2054, 2061, 2064, 2065, 2066,		381, 382, 383, 384
2067, 2068, 2069, 2074, 2078, 2114		__hyp_setup_info_date_key:nn	
\c__hyp_dest_startview_regex	2213, 2246, 2247
.	557, 1887, 1918	__hyp_setup_info_key:nn	
\c__hyp_dest_undefined_tl	2206, 2207, 2208, 2209, 2211, 2212
.	505, 678, 679	__hyp_store_metadata:nn	
__hyp_destination:nn	385, 390, 1498, 2174,
.	32, 620, 620, 651, 660	2192, 2226, 2241, 2250, 2266, 2334	
\l__hyp_filename_tmpa_tl	499,	__hyp_text_cleanup:N	577, 577, 591
780, 782, 788, 789, 794, 841, 842, 858		\l__hyp_text_enc_dest_print_tl	
__hyp_href_pdf_aux:nn	319, 322	528, 797
\l__hyp_href_pdf_destination_tl		\l__hyp_text_enc_dest_tl	
.	206, 217, 256, 324	32, 528, 627, 684

\l__hyp_text_enc_file_print_tl ..	528, 840	colorscheme	1, 1406
\l__hyp_text_enc_info_print_tl ..	528, 602	colorurl	994
\l__hyp_text_enc_para_print_tl ..	528, 845	debug	1430
\l__hyp_text_enc_uri_print_tl ...	269,	destlabel	13
272, 298, 301, 357, 360, 528, 744, 1490		draft	1430
__hyp_text_pdfstring:nnN	585, 585, 597, 599,	extension	13, 1448
602, 625, 682, 742, 795, 838, 843, 1490		file	10, 1479
__hyp_text_pdfstring_info:nn	600, 600, 2163, 2167, 2186, 2196	fileborderstyle	14, 1108
__hyp_text_purify:nn	573, 573, 590	filecolor	994
__hyp_text_string_from_unicode:nn	581, 581, 592	final	1430
\g__hyp_text_tmpa_str	504, 593, 595	hidefile	1382
\l__hyp_text_tmpa_str	503, 590, 591, 592, 593	hidelink	1382
\l__hyp_tmpa_box	494, 1222, 1235, 1240	hidelinks	1382
\l__hyp_tmpa_int	494	hidemenu	1382
\l__hyp_tmpa_seq	494, 1887, 1889, 1918, 1920,	hiderun	1382
2041, 2042, 2046, 2048, 2072, 2079,		hideurl	1382
2080, 2081, 2090, 2092, 2102, 2107,		hypertextnames	1448
2160, 2161, 2167, 2540, 2542, 2548		link	10, 1479
\l__hyp_tmpa_str	494, 2163, 2167, 2169, 2171,	linkborder	13
2186, 2187, 2189, 2196, 2197, 2201		linkborderstyle	14, 1108
\l__hyp_tmpa_tl	494, 628, 632, 640, 1039, 1043,	linkcolor	994
1490, 1491, 1496, 1885, 1887, 1916,		linkfileprefix	1448
1918, 2048, 2049, 2056, 2158, 2160		linktoc	1448
\l__hyp_uri_tmpa_tl ...	499, 745, 746	linktocpage	1448
hyp/anchor	646	localanchorname	1448
hyp/annot/file (color name)	542	menu	10, 1479
hyp/annot/link (color name)	542	menuborder	13
hyp/annot/menu (color name)	542	menuborderstyle	14, 1108
hyp/annot/run (color name)	542	menucolor	994
hyp/annot/url (color name)	542	naturalnames	1448
hyp/text/pdfstring	572	nesting	13
\hypercalcbp	11, 18, 161	ocgcolorfile	1248
\HyperDestNameFilter	13, 626, 683	ocgcolorlink	1248
\hypersetup 1, 2, 5, 9, 10, 13, 18, 67, 99, 180		ocgcolorlinks	1248
\hypersetup keys:		ocgcolormenu	1248
allcolors	994	ocgcolorrun	1248
bookmarkstype	13	ocgcolorurl	1248
bordercolormodel	13, 1015	pageanchor	1448
colorfile	994	pdfauthor	2130
colorlink	994	pdfborder	13
colorlinks	980	pdfborderstyle	14, 1108
colormenu	994	pdfcreationdate	14, 2213
colorrun	994	pdfcreator	2130
		pdfencoding	1420
		pdfinfo	2280
		pdfkeywords	2130
		pdflang	14, 2119
		pdflinkmargin	14
		pdfmetadate	14, 2213
		pdfmoddate	14, 2213
		pdfproducer	2130
		pdfremotestartview	11
		pdfstartview	11
		pdfsubject	2130

pdftitle	2130	1351, 1382, 1395, 1406, 1420, 1430,
pdftrapped	2252	1435, 1448, 1460, 1481, 1486, 2037,
pdfversion	1420	2119, 2134, 2177, 2215, 2230, 2248,
pdfview	11, 2037	2252, 2280, 2332, 2338, 2351, 2371
plainpages	1448	\l_keys_key_str 191, 2201, 2436
run	10, 1479	\keys_set:nn 183,
runborder	13	220, 239, 266, 295, 318, 335, 354,
runborderstyle	14, 1108	404, 415, 1412, 1419, 2257, 2284,
runcolor	994	2287, 2288, 2289, 2290, 2361, 2436
unicode	1420	\keys_set_known:nn 2439
url	10, 1479	\kvsetkeys 182, 2527
urlborder	13	
urlborderstyle	14, 1108	
urlcolor	994	
verbose	1430	
hypertextnames (hypersetup key)	1448	
hypListbox internal commands:		
_hypListbox	2707, 2766	
hypRadio internal commands:		
_hypRadio	2692, 2714	
hypwritetorep internal commands:		
_hypwritetorep	3469	
I		
\ifcase	3253, 3377, 3412	
\ifcsname	3030	
\ifdim	2657, 2696, 2697	
\ifnum	3084, 3253, 3259, 3290, 3377,	
	3382, 3412, 3417, 3544, 3562, 3577	
\iftrue	3546	
\ifx	2489, 2626, 2686,	
	3061, 3073, 3077, 3094, 3098, 3102,	
	3106, 3110, 3114, 3118, 3122, 3126,	
	3130, 3147, 3151, 3254, 3255, 3263,	
	3267, 3274, 3294, 3298, 3305, 3306,	
	3378, 3386, 3413, 3421, 3494, 3556	
\immediate	2473, 2476	
int commands:		
\int_compare:nNnTF	2046,	
	2072, 2758, 3176, 3229, 3339, 3361	
\int_compare_p:nNn	3222, 3330	
\int_eval:n	804	
\int_max:nn	805	
\int_new:N	497	
iow commands:		
\iow_newline:	2979, 2985, 2998	
K		
\kern	3501	
keys commands:		
\keys_define:nn	186, 194,	
	212, 250, 394, 980, 996, 1003, 1015,	
	1024, 1049, 1064, 1083, 1110, 1129,	
	1252, 1266, 1278, 1301, 1307, 1322,	
		1351, 1382, 1395, 1406, 1420, 1430,
		1435, 1448, 1460, 1481, 1486, 2037,
		2119, 2134, 2177, 2215, 2230, 2248,
		2252, 2280, 2332, 2338, 2351, 2371
		\l_keys_key_str 191, 2201, 2436
		\keys_set:nn 183,
		220, 239, 266, 295, 318, 335, 354,
		404, 415, 1412, 1419, 2257, 2284,
		2287, 2288, 2289, 2290, 2361, 2436
		\keys_set_known:nn 2439
		\kvsetkeys 182, 2527
L		
\label	9, 13	
\LayoutCheckField	2919	
\LayoutChoiceField	2689	
\LayoutPushButtonField	2795, 2812	
\LayoutTextField	2627	
\leavevmode	2628, 2725, 3499	
legacy commands:		
\legacy_if:nTF	168, 437, 444,	
	612, 2565, 2791, 2832, 2882, 2915, 3199	
\let	382, 383, 384,	
	709, 760, 767, 821, 827, 871, 877,	
	920, 2449, 2450, 2452, 2453, 2480,	
	2576, 2577, 2578, 2609, 2610, 2644,	
	2645, 2687, 3026, 3033, 3035, 3040,	
	3057, 3091, 3510, 3524, 3527, 3536,	
	3547, 3554, 3557, 3559, 3567, 3575	
link (hypersetup key)	10, 1479	
linkborder (hypersetup key)	13	
linkborderstyle (hypersetup key)	14, 1108	
linkcolor (hypersetup key)	994	
linkfileprefix (hypersetup key)	1448	
linktoc (hypersetup key)	1448	
linktocpage (hypersetup key)	1448	
localanchorname (hypersetup key)	1448	
M		
\MakeButtonField	2800, 2815, 2839, 2886, 2897	
\MakeChoiceField	2775	
\MakeFieldObject	2440	
\MakeRadioField	2757	
\MakeTextField	2636	
\mbox	1232	
menu (hypersetup key)	10, 1479	
menuborder (hypersetup key)	13	
menuborderstyle (hypersetup key)	14, 1108	
menucolor (hypersetup key)	994	
mode commands:		
\mode_if_horizontal:TF	622, 644	
\mode_leave_vertical:	263, 292,	
	315, 332, 351, 671, 754, 813, 860,	
	897, 2769, 2797, 2814, 2836, 2878, 2921	

msg commands:

- \msg_error:nn 425, 2811, 2896
- \msg_info:nnn 2144, 2149
- \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, 1426
- \msg_warning:nnn 179, 675, 915
- \msg_warning:nnnn
..... 1270, 1311, 1524, 1568,
1602, 1625, 1687, 1716, 1750, 1765,
1778, 1791, 1816, 1849, 1865, 1892,
1923, 1956, 1970, 1997, 2031, 2113
- \msg_warning:nnnn 190,
1342, 1376, 1468, 1544, 1581, 1668,
1701, 1726, 1802, 1829, 1878, 1983,
2010, 2274, 2378, 2397, 2416, 2427

N

- naturalnames (hypersetup key) 1448
- nesting (hypersetup key) 13
- \newcommand 164, 3473, 3475, 3477
- \newcount 2581, 2765
- \NewDocumentCommand 2440
- \NewExpandableDocumentCommand 381
- \newlength 3492
- \noexpand ... 2663, 2664, 2665, 2666, 3044
- \nolinkurl 4

O

- ocgcolorfile (hypersetup key) 1248
- ocgcolorlink (hypersetup key) 1248
- ocgcolorlinks (hypersetup key) 1248
- ocgcolormenu (hypersetup key) 1248
- ocgcolorrun (hypersetup key) 1248
- ocgcolorurl (hypersetup key) 1248

P

- pageanchor (hypersetup key) 1448
- \paperwidth 3
- \PassOptionsToPackage ... 446, 1440, 1445

pdf commands:

- \pdf_bdcobject:nn 1234, 1237
- \pdf_destination:nn 27, 155, 639
- \pdf_destination:nnnn 632
- \pdf_emc: 1236, 1242
- \pdf_link_user:nnn 2772
- \pdf_name_from_unicode_e:n
..... 245, 784, 899
- \pdf_object_if_exist:nTF 782
- \pdf_object_new:n 1156,
1157, 1158, 1159, 2966, 2967, 2968
- \pdf_object_ref:n 156, 794,
1162, 1164, 1186, 1187, 1190, 1194,
1199, 1204, 1209, 2558, 2560, 3024
- \pdf_object_ref_last: 815, 2366
- \pdf_object_unnamed_write:nn ...
..... 814, 2362
- \pdf_object_write:nnn 1160,
1166, 1176, 1188, 2969, 3011, 3018
- \pdf_pageobject_ref:n
..... 157, 1909, 1934, 2596
- \pdf_string_from_unicode:nnN .. 583
- \pdf_version:
... 1572, 1691, 1720, 1754, 1769,
1795, 1820, 1853, 1869, 1974, 2001
- \pdf_version_compare:NnTF
1562, 1653, 1675, 1711, 1744, 1759,
1785, 1810, 1836, 1859, 1964, 1991
- \pdf_version_compare_p:Nn
..... 851, 1249, 1298
- \pdf_version_major:
..... 167, 1250, 1273, 1299, 1315
- \pdf_version_minor: .. 166, 1273, 1315

pdfannot commands:

- \pdfannot_box:nnnn
... 2631, 2742, 2801, 2852, 2887, 2951
- \pdfannot_box_ref_last: .. 2487, 2503
- \pdfannot_dict_put:nnn
..... 755, 815, 861, 900, 1040,
1075, 1099, 1121, 1145, 1327, 1358
- \pdfannot_dict_remove:nn .. 1030,
1070, 1091, 1116, 1137, 1335, 1368
- \pdfannot_link:nnn . 756, 816, 862, 901
- \pdfannot_link_goto_begin:nw .. 688
- \pdfannot_link_goto_end: 693
- \pdfannot_link_margin:n 8, 1606
- \c_pdfannot_link_types_seq 551

pdfauthor (hypersetup key) 2130

\pdfbookmark 3478

pdfborder (hypersetup key) 13

pdfborderstyle (hypersetup key) . 14, 1108

pdfcreationdate (hypersetup key) 14, 2213

pdfcreator (hypersetup key) 2130

\pdfdest 11

pdfdict commands:

- \pdfdict_new:n 570, 733, 770, 829, 886
- \pdfdict_put:nnn
.... 237, 244, 571, 734, 735, 746,
749, 771, 772, 784, 791, 801, 811,
830, 831, 842, 857, 887, 888, 898,
1632, 1633, 1637, 1638, 2375, 2385,
2389, 2392, 2394, 2413, 2422, 2424
- \pdfdict_remove:nn 234, 854, 1642, 1643
- \pdfdict_use:n
... 755, 814, 861, 867, 900, 905, 2364

<code>\RequirePackage</code>	6, 31, 441, 3481
<code>run</code> (hypersetup key)	10, 1479
<code>runborder</code> (hypersetup key)	13
<code>runborderstyle</code> (hypersetup key)	14, 1108
<code>runcolor</code> (hypersetup key)	994
S	
<code>\selectfont</code>	2737, 2934, 2946
seq commands:	
<code>\seq_const_from_clist:Nn</code>	506
<code>\seq_count:N</code>	2046, 2072
<code>\seq_get_right:NN</code>	2048
<code>\seq_if_empty:NTF</code>	2161
<code>\seq_item:Nn</code>	1889, 1920, 2042, 2090, 2092, 2102, 2107, 2167
<code>\seq_map_inline:Nn</code>	543, 547, 551, 994, 1295, 1393, 2548
<code>\seq_new:N</code>	496
<code>\seq_put_right:Nn</code>	2079, 2080, 2081, 2540
<code>\seq_set_split:Nnn</code>	2041
<code>\seq_sort:Nn</code>	2542
<code>\setbox</code>	375
<code>\setlength</code>	3513, 3584
<code>\setpdfmargin</code>	8
<code>\settowidth</code>	2656
<code>\show</code>	2716
sort commands:	
<code>\sort_return_same:</code>	2546
<code>\sort_return_swapped:</code>	2545
<code>\space</code>	480, 2724, 3256, 3273, 3274, 3304, 3305, 3379, 3411, 3414
<code>\spacefactor</code>	622, 644
<code>\stockwidth</code>	3
str commands:	
<code>\c_colon_str</code>	881
<code>\c_hash_str</code>	751
<code>\c_percent_str</code>	752
<code>\str_case:nnTF</code>	2140, 3436
<code>\str_case_e:nnTF</code>	2042
<code>\str_compare:nNnTF</code>	2544
<code>\str_gset:Nn</code>	1019
<code>\str_gset_eq:NN</code>	593
<code>\str_head:n</code>	2263, 2268
<code>\str_if_eq:nnTF</code>	629, 2169, 2187, 2197, 3138
<code>\str_if_eq_p:nn</code>	849, 1250, 1299
<code>\str_lowercase:n</code>	2042, 2192, 2241, 2264, 2269
<code>\str_new:N</code>	498, 503, 504, 542
<code>\str_set:Nn</code>	575
<code>\str_set_convert:Nnnn</code>	8
<code>\str_set_eq:NN</code>	595
<code>\str_tail:n</code>	2264, 2269
<code>\str_uppercase:n</code>	2257, 2263, 2268
<code>\string</code>	2474, 2477, 2498, 2516, 3049, 3053, 3064
<code>\subpdfbookmark</code>	3474
sys commands:	
<code>\c_sys_backend_str</code>	3436
T	
TeX and L ^A T _E X 2 _ε commands:	
<code>\BOOKMARK</code>	3480
<code>\@CheckBox</code>	2902
<code>\@ChoiceMenu</code>	2642
<code>\@Form</code>	2525
<code>\@PushButton</code>	2780
<code>\@Reset</code>	2869
<code>\@Submit</code>	2821
<code>\@TextField</code>	2607
<code>\@bookmarksopenlevel</code>	1507
<code>\@bsphack</code>	453, 465
<code>\@chapter</code>	3536, 3537
<code>\@currDisplay</code>	2656, 2724
<code>\@currValue</code>	2722, 2752
<code>\@currentHref</code>	3517, 3530, 3541, 3571, 3588
<code>\@curropt</code>	2641, 2651, 2654, 2718, 2721
<code>\@empty</code>	2610, 2626, 3094, 3098, 3102, 3106, 3110, 3114, 3118, 3122, 3126, 3130, 3274, 3305, 3306
<code>\@endForm</code>	2576
<code>\@esphack</code>	455, 467
<code>\@firstofone</code>	2480, 3035, 3040, 3057
<code>\@firstoftwo</code>	3578
<code>\@for</code>	2652, 2719
<code>\@gobbletwo</code>	3527
<code>\@hyper@launch</code>	38, 881
<code>\@ifnextchar</code>	274
<code>\@ifundefined</code>	3049, 3508
<code>\@latex@warning</code>	477
<code>\@mainaux</code>	2473, 2476, 2496, 2515
<code>\@mkboth</code>	3527
<code>\@nil</code>	3047, 3060, 3066
<code>\@part</code>	3554, 3561
<code>\@pdfauthor</code>	23
<code>\@pdfborder</code>	162
<code>\@pdfborderstyle</code>	163
<code>\@savsf</code>	622, 644
<code>\@schapter</code>	3524, 3525
<code>\@secondoftwo</code>	3580
<code>\@sect</code>	3575, 3576
<code>\@spart</code>	3567, 3568
<code>\@ssect</code>	3510, 3511
<code>\@tempdima</code>	2650, 2657, 2659, 2696, 2697, 2698, 2702, 2703, 2704, 2705
<code>\@tempdimb</code>	2656, 2657

\@typeset@protect	3494	\Fld@onmousedown@code ...	3118, 3120
\BKM@color	434	\Fld@onmouseup@code	3122, 3124
\c@secnumdepth	3544, 3562, 3577	\Fld@pageobjref	
\calc@bm@number	3486	2578, 2592, 3029, 3161,
\check@bm@number	3485		3214, 3247, 3283, 3322, 3371, 3406
\define@key	428	\Fld@radiosymbol	3302
\Fld@additionalactions	3136, 3206,	\Fld@rotation	3176,
	3240, 3276, 3313, 3360, 3400, 3429		3178, 3222, 3229, 3231, 3253, 3259,
\Fld@align	3165, 3251, 3326		3261, 3290, 3292, 3330, 3339, 3341,
\Fld@altname ..	3073, 3075, 3147, 3149		3377, 3382, 3384, 3412, 3417, 3419
\Fld@annotflags	3160,	\Fld@submitflags	3398
	3213, 3246, 3282, 3321, 3370, 3405	\Fld@validate@code	3102, 3104
\Fld@annotnames	3145, 3162,	\Fld@value	
	3215, 3248, 3284, 3323, 3372, 3407	..	2610, 2626, 2645, 2686, 2687, 3359
\Fld@bcolor	3184, 3186, 3267,	\Fld@width	2611,
	3269, 3298, 3300, 3334, 3347, 3349		2625, 2632, 2636, 2646, 2685, 2696,
\fld@bcolor	3255		2698, 2743, 2757, 2775, 2793, 2823,
\Fld@bordercolor	3180,		2834, 2871, 2883, 2907, 2917, 2952
	3182, 3224, 3233, 3235, 3254, 3263,	\Fld@X@additionalactions	
	3265, 3294, 3296, 3332, 3343, 3345,	3093, 3138, 3141
	3378, 3386, 3388, 3413, 3421, 3423	\Form@action	3396
\Fld@borderstyle	3166,	\G@refundefinedtrue	476
	3219, 3252, 3288, 3327, 3376, 3427	\H@old@part	3554, 3565
\Fld@borderwidth ..	2703, 2704, 3166,	\H@old@chapter	3524, 3533
	3219, 3252, 3288, 3327, 3376, 3427	\H@old@sect	3575, 3586, 3595
\Fld@calculate@code ..	2489, 3106, 3108	\H@old@spart	3567, 3573
\Fld@calculate@sortkey	2500	\H@old@ssect	3510, 3515
\Fld@cbsymbol	3188	\href@	274, 280
\Fld@charsize		\href@split	280, 281
..	2702, 3192, 3273, 3304, 3355, 3411	\Hy@abspage	616
\Fld@checkequals	2654, 2721	\Hy@activeanchorfalse	667
\Fld@choices	3275	\Hy@activeanchortrue	658
\Fld@color	3193, 3195, 3274, 3305, 3356	\Hy@AtBeginDocument	2454, 2470
\Fld@default		\Hy@bookmarkstyle	1508
...	2609, 2626, 2644, 2687, 2716,	\Hy@chapapp	3528, 3539
	2717, 2905, 3306, 3310, 3311, 3358	\Hy@colorlink	34
\Fld@flags	3164,	\Hy@currentbookmarklevel	3467
	3217, 3250, 3286, 3325, 3374, 3409	\Hy@DisableOption	172
\Fld@format@code	3098, 3100	\Hy@drafttrue	1439
\Fld@height		\Hy@escapeform	
	2612, 2633, 2636, 2647, 2705, 2744,	2630, 2727, 2771, 2799,
	2757, 2775, 2824, 2872, 2908, 2953		2838, 2885, 2923, 3031, 3038, 3041
\Fld@keystroke@code	3094, 3096	\Hy@escapestring	3033,
\Fld@listcount	2715, 2723, 2758, 2765		3035, 3040, 3043, 3044, 3047, 3048,
\Fld@mappingname	3077, 3079, 3151, 3153		3057, 3060, 3096, 3100, 3104, 3108,
\Fld@maxlen	3361, 3363		3112, 3116, 3120, 3124, 3128, 3132,
\Fld@menulength	2649, 2655, 2702		3188, 3239, 3302, 3358, 3359, 3396
\Fld@name		\Hy@finaltrue	1444
..	2608, 2643, 2781, 2904, 3072, 3146	\Hy@FormObjects	2530, 2964, 3026
\Fld@onblur@code	3114, 3116	\Hy@gttemp	3089, 3091
\Fld@onclick@code	3239	\Hy@href	274
\Fld@onenter@code	3126, 3128	\Hy@href@nextactionraw	238
\Fld@onexit@code	3130, 3132	\Hy@href@page	226
\Fld@onfocus@code	3110, 3112	\Hy@linkfileprefix	1453

\Hy@linktoc	1464	\HyField@PDFChoices	2768
\Hy@MakeCurrentHref	199	\HyField@SetKeys	2620, 2661,
\Hy@MakeCurrentHrefAuto		2664, 2683, 2783, 2826, 2874, 2909	
..... 3512, 3528, 3539, 3569, 3583		\HyPat@ObjRef	3440, 3447, 3454, 3461
\Hy@next	3538, 3547, 3550	\hyper@link	282
\Hy@numberline	165, 3468	\hyper@anchor	646
\Hy@org@chapter	3536, 3551	\hyper@anchorend	
\Hy@OutlineName	3484 646, 3503, 3530, 3541, 3571	
\Hy@OutlineRerunCheck	3482	\hyper@anchorstart	
\Hy@pdfmajorversion	167 646, 3503, 3530, 3541, 3571	
\Hy@pdfminorversion	166	\hyper@link	34, 696
\Hy@pdfstringtrue	30, 588	\hyper@linkend	34, 724
\Hy@pdfversion	3084	\hyper@linkfile	324, 775
\Hy@pstringdef	598, 3064	\hyper@linklaunch	37, 341, 833, 884
\Hy@PutCatalog	604	\hyper@linknamed	38, 175, 890
\Hy@raisedlink 3502, 3529, 3540, 3570		\hyper@linkstart	34, 712
\Hy@RestoreLastskip	643	\hyper@linkurl	307, 368, 737
\Hy@safe@activestruer	753, 2492	\hyper@normalise	
\Hy@SaveLastskip	623 274, 303, 319, 336, 362, 375	
\Hy@secnum@part	3557, 3559, 3562	\HyPL@Labels	606, 616
\Hy@SectionAnchorHref 3493, 3517, 3588		\HyPL@StorePageLabel	604
\Hy@SectionHShift		\HyPsd@SanitizeForOutFile	3471
..... 3492, 3501, 3513, 3584		\if@files	2472, 2495, 2514
\Hy@StepCount	2655, 2723	\ifFld@combo	2670, 2700
\Hy@temp	2493, 2508, 3046, 3047	\ifFld@hidden	2625, 2685
\Hy@unicodetrue	3086	\ifFld@multiline	2613
\Hy@VerboseAnchor	624	\ifFld@popdown	2671
\Hy@VerboseLinkStart	700, 716	\ifFld@radio	2667, 2690
\Hy@VerboseLinkStop		\ifHy@implicit	3488
..... 707, 730, 763, 823, 873, 911		\ifHy@pdfescapeform	3032, 3042
\Hy@VersionChecked	158	\ifHy@unicode	3085
\Hy@WrapperDef	620	\kv@set@family@handler	178
\Hy@xspace@end 704, 762, 822, 872, 910		\ltx@empty	2449,
\HyAnn@AbsPageLabel		2450, 2489, 2576, 2577, 2578, 2609	
..... 2577, 2583, 2629,		\ltx@gobble	2452
2726, 2770, 2798, 2837, 2884, 2922		\ltx@gobbletwo	2453
\HyAnn@Count 2581, 2582, 2585, 2586,		\ltx@ifUndefined	3535, 3545
2587, 2588, 2589, 2590, 2594, 2598		\ltx@one	2585
\HyField@AddToFields	2483, 2637,	\ltx@ReturnAfterFi	3065
2759, 2776, 2807, 2863, 2892, 2956		\ltx@zero	2582
\HyField@afields	2449	\m@ne	3544, 3559
\HyField@AfterAuxOpen 2454, 2480, 2513		\OBJ@OCG@view	164
\HyField@AuxAddToCoFields		\p@	2659
..... 2453, 2465, 2477, 2498		\pdf@ifdraftmode	2528
\HyField@AuxAddToFields		\PDF@SetupDoc	159
..... 2452, 2458, 2474, 2516		\PDFForm@Check	2923, 2955, 3157
\HyField@cofields	2450	\PDFForm@List	2771, 2774, 3244
\HyField@FlagsCheckBox	2914	\PDFForm@Name	
\HyField@FlagsChoice	2695 2623, 2684, 2787, 2913, 3071	
\HyField@FlagsPushButton		\PDFForm@Push	2799, 2805, 3210
..... 2790, 2830, 2881		\PDFForm@Radio	2727, 2747, 3279
\HyField@FlagsRadioButton	2691	\PDFForm@Reset	2885, 2891, 3403
\HyField@FlagsSubmit	2831	\PDFForm@Submit	2838, 2857, 3368
\HyField@FlagsText	2624	\PDFForm@Text	2630, 2635, 3318

<code>\protected@edef</code>	672	<code>\tl_if_exist:N</code> TF	3233, 3343, 3347
<code>\strip@pt</code> .	3192, 3273, 3304, 3355, 3411	<code>\tl_if_exist_p:N</code>	3224, 3332, 3334
<code>\toks@</code>	3515, 3521, 3586, 3592	<code>\tl_if_head_eq_charcode:n</code> NTF	926, 943
<code>\tw@</code>	3516, 3521, 3587, 3592	<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>	1450	 225, 226, 238, 269, 272, 298,
<code>\z@</code>	375, 2650, 3253, 3259,		301, 357, 360, 387, 535, 536, 537,
	3290, 3377, 3382, 3412, 3417, 3557		538, 539, 540, 652, 661, 702, 718,
<code>\zref@labelbylist</code>	2587		780, 1885, 1889, 1893, 1916, 2051,
<code>\zref@labelbyprops</code>	2586		2054, 2061, 2064, 2065, 2066, 2067,
<code>\zref@refused</code>	2588		2068, 2069, 2074, 2078, 2114, 2158
<code>\texorpdfstring</code>	30	<code>\tl_set_eq:NN</code>	679
text commands:		<code>\tl_to_str:N</code>	782, 789, 794
<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>	2845, 2849	<code>\token_to_str:N</code>	41
<code>\the</code>	616,	<code>\toks</code>	3516, 3521, 3587, 3592
	2586, 2587, 2588, 2589, 2590, 2594,		
	2598, 2698, 2702, 2705, 3521, 3592		
<code>\theH...</code>	12	U	
<code>\thepage</code>	480	unicode (hypersetup key)	1420
<code>\ThisShouldNotHappen</code>	3051	<code>\Url</code>	4, 5, 209
<code>\title</code>	2	url (hypersetup key)	10, 1479
tl commands:		<code>\url</code>	4–6, 22, 349, 375
<code>\c_space_tl</code>	165, 1163,	urlborder (hypersetup key)	13
	1194, 1199, 1204, 2752, 2753, 2760,	urlborderstyle (hypersetup key) .	14, 1108
	2975, 2977, 2987, 2989, 2991, 2994,	urlcolor (hypersetup key)	994
	2996, 3192, 3195, 3327, 3355, 3356	<code>\urldef</code>	5, 22, 375, 377
<code>\tl_const:Nn</code>	505	<code>\Urlfont</code>	6
<code>\tl_gput_right:Nn</code>	616	<code>\UrlSpecial</code>	4
<code>\tl_gset:Nn</code>	1900, 1920, 1924	use commands:	
<code>\tl_if_blank:n</code> TF		<code>\use:n</code>	879
	799, 2138, 2181, 2219, 2234, 2342, 2355		
<code>\tl_if_empty:N</code> TF		V	
	... 673, 1491, 3180, 3184, 3193, 3356	verbose (hypersetup key)	1430
<code>\tl_if_empty:n</code> TF			
	... 232, 1028, 1068, 1087,	W	
	1114, 1133, 1677, 1733, 1838, 2123	<code>\write</code>	2473, 2476, 2496, 2515
<code>\tl_if_empty_p:N</code>	1902, 1927		
<code>\tl_if_eq:Nn</code> TF	2049	X	
		<code>\x</code>	2662, 2682, 3520, 3522, 3591, 3593