

The `hyperref-generic` module

A generic driver for `hyperref`

The L^AT_EX Project*

Version 0.96q, released 2025-03-26

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”, information 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 data 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 split. The first value is used for the Info dictionary, the others are used in the XMP-metadata. Commas in a title must then be protected with braces:

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

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

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

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

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

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

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

4 Dates

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

5 PDF page size (mediabox)

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

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

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

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

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

6 Commands to create “external” references

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

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

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

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

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

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

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

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

6.1 Special problem: links to files

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

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

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

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

6.2 Splits

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

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

For this reason three new commands have been create:

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

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

6.3 Options

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

key	applicable commands	note
<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

```

\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

```

These boolean are used by the `colorlinks` and `ocgcolorlinks` and related keys. These keys insert 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` or `colorlinks=true` will as before disable the `pdfborder` (`colorlinks=false` will leave the `pdfborder` untouched), but it is possible to use the key in the document at any time, or to reenable the border if wanted. Internally `colorlinks` & friends will no longer define/undefine `\Hy@colorlink`, but instead use the hooks provided by the `l3pdfannot` package.

Color keys accept the following input syntax:

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

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

frenchlinks The option `frenchlinks` does nothing at all.

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

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

8 PDF strings

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

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

The new method is under heavy development!

Important differences here are

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

9 Package options from hyperref

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

Options that still must be set as package options are

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

Options that can be without problems set as package options are

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

Ignored options:

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

10 Disabling links

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

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

`nested-links` (*setup key*)

Links are sometimes nested. E.g. if a section heading contains a reference it can lead to nested links in the table of contents or if `\nameref` is used. That is not forbidden and normally work as expected: If the link area overlap normally the inner link is "on top" and chosen at a click. But it is not always actually wanted, so with the `nested-links` (a boolean key) it is possible to disable such nested links.

11 Draftmode

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

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

12 Dropped options

A number of options are ignored by this driver

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

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

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

unicode This is always true.

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

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

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

- The `/F` key is added to links and `Print` is activated, `Hidden`, `Invisible`, `NoView` are deactivated.
- `/NeedAppearances` is suppressed
- `PushButtons`, which use the action `/S/JavaScript` are suppressed.
- `ResetButtons`, which use the action `/S/ResetForm` are suppressed.
- In widget annotations, the `/AA` dictionary is suppressed.

13 Destinations

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

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

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

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

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

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

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

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

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

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

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

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

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

13.1 Names of destinations

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

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

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

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

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

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

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

14 Assorted key descriptions

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

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

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

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

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

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

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

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

will create

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

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.

`nesting` (*setup key*)

`pdfborder` (*setup key*) This key set accept as value three numbers or three numbers and an array describing

`linkborder` (*setup key*) a dash pattern, examples are `0 0 1` or `0 0 1 [3 2]`. The first two numbers should

`urlborder` (*setup key*) according to the reference set round corners, but PDF viewer seem to ignore it. The

`runborder` (*setup key*) third number is the line width of the border. Settings done with `pdfborderstyle` should

`menuborder` (*setup key*) take precedence.

`pdfborderstyle` (*setup key*) The value of this key is the content of the BS dictionary. As an example

`linkborderstyle` (*setup key*) `/Type/Border /W 1 /S/U /D[3 2]`

<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])

`pdfcreationdate` (*setup key*) Setting these keys is normally not needed. If they are used the values of the first

`pdfmoddate` (*setup key*) two keys are stored directly in the Info dictionary for `/Creationdate` and `/ModDate`.

`pdfmetadate` (*setup key*) 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. `D:20200202111111+01'00'`.

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

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

File I

hyperref-generic driver implementation

```

1 <*package>
2 <@=hyp>
3 \ProvidesFile{hgeneric-testphase.def}[2025-03-26 v0.96q %
4   generic Hyperref driver for the LaTeX PDF management testphase bundle]
5
6 \RequirePackage{etoolbox} %why?

```

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

```

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

```

1 messages

Redirect the message name:

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

At first a message for the testing of the resource management

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

The pdfversion should be set in \DocumentMetadata

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

A generic message for ignored keys.

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

```
57 }
```

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

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

pdf/A message for not allowed Named actions

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

A message if the destination name is empty.

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

A message if the destination check fails

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

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

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



```

103     Value-`#1`~is~unknown~or~deprecated~in\\
104     pdf-version-#2. Value-`#3`~is used instead.
105 }

```

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

```

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

```

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

```

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

```

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

```

149 \msg_new:nnn

```

```

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

```

2 Variants

```

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

```

3 Overwriting/providing commands from hyperref

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

```

159 \chardef\Hy@VersionChecked=1 %don't check the version!
160 %\cs_set_protected:Npn \PDF@SetupDoc{}
161 %\PDF@FinishDoc{}% dummy needed for hyperref ...

```

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

\hypercalcbp

```

162 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n

```

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

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

```

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

```

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

```

167 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
168 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:
169 \legacy_if:nT { Hy@setpdfversion }
170 {
171   \msg_warning:nn { hyp }{ pdfversion-disabled }
172 }
173 \Hy@DisableOption{pdfversion}

```

\Acrobatmenu should use the new internal link command

```

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

```

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

```

178 % TODO should go at some time ...
179 % \kv@set@family@handler{Hyp}
180 % { \msg_warning:nne {hyp}{unknown-key-to-Hyp}{#1} }
181 \cs_set_protected:Npn \hypersetup #1
182 {
183   %\kvsetkeys{Hyp} {#1}
184   \keys_set:nn { hyp }{ #1 }
185 }
186 % TODO for now unknown keys should only give warnings.
187 \keys_define:nn { hyp }
188 {
189   unknown .code:n =
190   {
191     \msg_warning:nneee { hyp } { unknown-key }
192     { hyp }{ \l_keys_key_str } { #1 }
193   }
194 }

```

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

```

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

```

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

```

203
204 \bool_new:N \l__hyp_href_url_encode_bool
205 \bool_new:N \l__hyp_href_url_ismap_bool
206 \tl_new:N \l__hyp_href_url_protocol_tl
207 \tl_new:N \l__hyp_href_pdf_destination_tl
208 \tl_new:N \l__hyp_href_pdf_page_tl
209 \tl_new:N \l__hyp_href_run_parameter_tl
210 \cs_new_protected:Npn \__hyp_href_url_format: {\begingroup\Url}
211
212
213 \keys_define:nn { hyp / href }
214 {
215   ,urlencode .bool_set:N = \l__hyp_href_url_encode_bool
216   ,format .code:n = { \cs_set:Nn \__hyp_href_url_format: {#1} },
217   ,protocol .tl_set:N = \l__hyp_href_url_protocol_tl
218   ,destination .tl_set:N = \l__hyp_href_pdf_destination_tl

```

```

219 ,pdfremotestartview .code:n =
220 {
221   \keys_set:nn { hyp }
222   { pdfremotestartview = #1 }
223 }
224 ,page .code:n =
225 {
226   \tl_set:Nn \l__hyp_href_pdf_page_tl {#1}
227   \tl_set:Nn \Hy@href@page {#1}
228 }
229 ,ismap .bool_set:N = \l__hyp_href_url_ismap_bool
230 ,run-parameter .tl_set:N = \l__hyp_href_run_parameter_tl
231 ,nextactionraw .code:n =
232 { %perhaps some safety match later, see hyperref code
233   \tl_if_empty:nTF {#1}
234   {
235     \pdfdict_remove:nn{l_hyp/annot/A}{Next}
236   }
237   {
238     \pdfdict_put:nnn{l_hyp/annot/A}{Next}{#1}
239     \tl_set:Nn \Hy@href@nextactionraw {/Next~#1}
240     \keys_set:nn {hyp }{ pdfnewwindow = true}
241   }
242 }
243 ,afrelationship .code:n =
244 {
245   \pdfdict_put:nne
246   { l_pdffile/Filespec}{AFRelationship}{ \pdf_name_from_unicode_e:n {#1}}
247 }
248
249 }
250
251 \keys_define:nn { hyp }
252 {
253   ,href / urlencode .bool_set:N = \l__hyp_href_url_encode_bool
254   ,href / urlencode .default:n = {true}
255   ,href / urlencode .initial:n = {false}
256   ,href / protocol .tl_set:N = \l__hyp_href_url_protocol_tl
257   ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
258   ,href / format .code:n = { \cs_set:Nn \__hyp_href_url_format:{#1} }
259 }
260
261 \hook_new_pair:nn{cmd/href/before}{cmd/href/after}
262
263 \DeclareRobustCommand*{\href}[1][ ]{%
264   \mode_leave_vertical:
265   \hook_use:n{cmd/href/before}
266   \group_begin:
267   \keys_set:nn { hyp / href } {#1}
268   \bool_if:NTF \l__hyp_href_url_encode_bool
269   {
270     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
271   }
272   {

```

```

273     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
274   }
275   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
276 }
277
278 \begingroup
279   \catcode'\$=6 %
280   \catcode'\#=12 %
281   \gdef\href@#1{\expandafter\href@split$1##\}%
282   \gdef\href@split$1#$2#$3\\$4{%
283     \hyper@@link{$1}{$2}{$4}%<---__hyp-docstrip doubling!
284     \endgroup
285     \hook_use:n{cmd/href/after}
286   }%
287 \endgroup
288
289 \hook_new_pair:nn{cmd/hrefurl/before}{cmd/hrefurl/after}
290
291 \DeclareRobustCommand*{\hrefurl}[1] []
292 {
293   \mode_leave_vertical:
294   \hook_use:n{cmd/href/before}
295   \group_begin:
296   \keys_set:nn { hyp / href } {#1}
297   \bool_if:NTF \l__hyp_href_url_encode_bool
298     {
299     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
300     }
301     {
302     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
303     }
304   \hyper@normalise\__hyp_href_url_aux:nn}
305
306 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
307 {
308   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
309   \group_end:
310   \hook_use:n{cmd/href/after}
311 }
312
313 \hook_new_pair:nn{cmd/hrefpdf/before}{cmd/hrefpdf/after}
314 \DeclareRobustCommand*{\hrefpdf}[1] []
315 {
316   \mode_leave_vertical:
317   \hook_use:n{cmd/hrefpdf/before}
318   \group_begin:
319   \keys_set:nn { hyp / href } {#1}
320   \hyper@normalise\__hyp_href_pdf_aux:nn
321 }
322
323 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
324 {
325   \exp_args:Nno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
326   \group_end:

```

```

327     \hook_use:n{cmd/hrefpdf/after}
328   }
329
330 \hook_new_pair:nn{cmd/hrefrun/before}{cmd/hrefrun/after}
331 \DeclareRobustCommand*{\hrefrun}[1] []
332 {
333   \mode_leave_vertical:
334   \hook_use:n{cmd/hrefrun/before}
335   \group_begin:
336   \keys_set:nn { hyp / href } {#1}
337   \hyper@normalise\__hyp_href_run_aux:nn
338 }
339
340 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
341 {
342   \exp_args:Nno\hyper@linklaunch{#1}{#2}{\l__hyp_href_run_parameter_tl}
343   \group_end:
344   \hook_use:n{cmd/hrefrun/after}
345 }
346
347
348 \hook_new_pair:nn{cmd/url/before}{cmd/url/after}
349
350 \DeclareRobustCommand*{\url}[1] []
351 {
352   \mode_leave_vertical:
353   \hook_use:n{cmd/url/before}
354   \group_begin:
355   \keys_set:nn { hyp / href } {#1}
356   \bool_if:NTF \l__hyp_href_url_encode_bool
357   {
358     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
359   }
360   {
361     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
362   }
363   \hyper@normalise\__hyp_href_url_aux:n
364 }
365
366 \cs_new_protected:Npn \__hyp_href_url_aux:n #1
367 {
368   \exp_args:Nno
369   \hyper@linkurl{\__hyp_href_url_format: {#1}}
370   {\l__hyp_href_url_protocol_tl#1}
371   \group_end:
372   \hook_use:n{cmd/url/after}
373 }
374

```

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

```

375
376 \def\urldef#1#2{\begingroup\def\url{\hyper@normalise\url@}\setbox\z@\hbox\bgroup
377   \def\url@HyperHook##1\endgroup{\url@def{#1}{#2}}%

```

```

378 % Because hyperref breaks \urldef and does not define its own (Grrrr!)...
379 \def\url@##1{\egroup\endgroup\DeclareRobustCommand#1{#2{##1}}}%
380 #2}
381

```

make the new commands compatible with `\pdfstringdef`:

```

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

```

4 Compatibility commands

4.1 Metadata

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

```

386 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
387 {
388   %\tl_set:cn {@#1}{#2}
389   \AddToDocumentProperties[hyperref]{#1}{#2}
390 }
391 \cs_generate_variant:Nn \__hyp_store_metadata:nn {en,ne,ee,no,eo}

```

4.2 citecolor

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

```

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

```

```

416     \keys_set:nn { hyp }
417     {
418         linkbordercolor      = hyp/color/citeborder
419     }
420 }
421 \cs_gset_eq:NN \__hyp_citebordercolor_hook_init: \prg_do_nothing:
422 }
423

```

5 Checks

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

```

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

```

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

```

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

```

empty hook chunk to ensure that the chunk exists.

```

449 {
450     \AddToHook{begindocument/before}[hyperref/bookmark]{}
451 }
452 \legacy_if:nT {Hy@draft}
453 {
454     \PassOptionsToPackage{draft}{bookmark}
455 }

```


6 Reference and label commands

This uses the in-built property module.

```
\__hyp_property_record:nn
```

```
456 %
```

A label command which adds the space commands from LaTeX:

```
457 \cs_new_protected:Npn \__hyp_property_record:nn #1 #2 %label/attributes
458 {
459   \@bsphack
460   \property_record:nn{#1}{#2}
461   \@esphack
462 }
```

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.

```
463 \cs_generate_variant:Nn \__hyp_property_record:nn {ee}
```

(End of definition for __hyp_property_record:nn.)

7 Variables

7.1 Private temporary variables

At first a few generic tmp variables

```
\l__hyp_tmpa_tl
\l__hyp_tmpb_tl 464 \box_new:N \l__hyp_tmpa_box
\l__hyp_tmpa_seq 465 \tl_new:N \l__hyp_tmpa_tl
\l__hyp_tmpa_int 466 \tl_new:N \l__hyp_tmpb_tl
\l__hyp_tmpa_box 467 \seq_new:N \l__hyp_tmpa_seq
\l__hyp_tmpa_str 468 \int_new:N \l__hyp_tmpa_int
469 \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  TODO: document and check use!
\l__hyp_uri_tmpa_tl 470 \tl_new:N \l__hyp_dest_name_tmpa_tl
\l__hyp_filename_tmpa_tl 471 \tl_new:N \l__hyp_uri_tmpa_tl
\__hyp_text_tmpa_str\__g__hyp_text_tmpa_str 472 \tl_new:N \l__hyp_filename_tmpa_tl
473 \tl_new:N \l__hyp_para_tmpa_tl
474 \str_new:N \l__hyp_text_tmpa_str
475 \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.

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

(End of definition for \c__hyp_dest_undefined_tl.)

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

`\c__hyp_map_annot_hyp_prop`

`\c__hyp_map_hyp_annot_prop`

```
477 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
478 {url,link,file,menu,run}
479 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
480 {
481   URI    = url,
482   GoTo   = link,
483   GoToR  = file,
484   Named  = menu,
485   Launch = run
486 }
487 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
488 {
489   url    = URI,
490   link   = GoTo,
491   file   = GoToR,
492   menu   = Named,
493   run    = Launch
494 }
495
```

(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!

`\g__hyp_dest_pdfstartview_tl`

`\l__hyp_dest_pdfremotestartview_tl`

```
496 \tl_new:N \g__hyp_dest_pdfstartpage_tl
497 \tl_new:N \g__hyp_dest_pdfstartview_tl
498 \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`

```
499 \tl_new:N \l__hyp_text_enc_uri_print_tl
500 \tl_new:N \l__hyp_text_enc_info_print_tl
501 \tl_new:N \l__hyp_text_enc_dest_tl
502 \tl_new:N \l__hyp_text_enc_dest_print_tl
503 \tl_new:N \l__hyp_text_enc_file_print_tl
504 \tl_new:N \l__hyp_text_enc_para_print_tl
```

```

505
506 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
507 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
508 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
509 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
510 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
511 \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.)

It is also unclear how the /Contents entry would look at best. So we use sockets. The first argument is the target (url or destination), For GoTo we also pass the text as argument. The sockets should put something into the relevant annotation dictionaries.

```

512 \tl_new:N\l__hyp_link_Contents_tl
513 \socket_new:nn {hyp/link/GoTo/Contents}{2}
514 \socket_new:nn {hyp/link/URI/Contents}{1}
515 \socket_new_plug:nmn {hyp/link/GoTo/Contents}{default}
516 {
517   \__hyp_text_pdfstring:eon
518   { Go-to-destination~#1 }
519   { \l__hyp_text_enc_info_print_tl }
520   \l__hyp_link_Contents_tl
521   \pdfannot_dict_put:nne {link/GoTo}{Contents}
522   {\l__hyp_link_Contents_tl}
523 }
524 \socket_new_plug:nmn {hyp/link/URI/Contents}{default}
525 {
526   \__hyp_text_pdfstring:eon
527   { #1 }
528   { \l__hyp_text_enc_info_print_tl }
529   \l__hyp_link_Contents_tl
530   \pdfannot_dict_put:nne {link/URI}{Contents}
531   {\l__hyp_link_Contents_tl}
532 }
533 \socket_assign_plug:nn{hyp/link/GoTo/Contents}{default}
534 \socket_assign_plug:nn{hyp/link/URI/Contents}{default}

```

\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.

```

535 \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)

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`

```

536 \str_new:N \g__hyp_bordercolormodel_str

```

(End of definition for \g__hyp_bordercolormodel_str.)

7.4 Booleans

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

```
537 \seq_map_inline:Nn \c__hyp_annot_types_seq
538 {
539   \bool_new:c {l_hyp_annot_color#1_bool}
540 }
```

(End of definition for `\l_hyp_annot_colorlink_bool` and others. These variables are documented on page 7.)

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

```
541 \seq_map_inline:Nn \c__hyp_annot_types_seq
542 {
543   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
544 }
```

(End of definition for `\l_hyp_annot_ocgcolorlink_bool` and others. These variables are documented on page 7.)

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

```
545 \seq_map_inline:Nn \c_pdfannot_link_types_seq
546 {
547   \bool_new:c {l__hyp_annot_#1_bool}
548   \bool_set_true:c {l__hyp_annot_#1_bool}
549 }
```

(End of definition for `\l__hyp_annot_GoTo_bool` `\l__hyp_annot_URI_bool` `\l__hyp_annot_GoToR_boo1` `\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.

```
550 \box_new:N \l__hyp_dest_box
```

(End of definition for `\l__hyp_dest_box`.)

7.6 Regex

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

```
551 \regex_const:Nn \c__hyp_dest_startview_regex
552 {
553   \A\ *
554   (?
555     (?:XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
556     |
557     (?:Fit\b|FitB\b)
558     |
559     (?:\b(?:FitH|FitV|FitBH|FitBV)\b(?:\ +(?:\d+|\d*\.\d+)|\ +null){1})
560     |
561     (?:FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
562   )
563 }
```

(End of definition for `\c__hyp_dest_startview_regex`.)

7.7 PDF dictionaries

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

```
564 \pdfdict_new:n {l__hyp_page/Trans}
565 \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`

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

(End of definition for `hyp/text/pdfstring`.)

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`

```
567 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
568 {
569   \str_set:Ne #2 {\text_purify:n { #1 } }
570 }
```

(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`

```
571 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
572 {
573
574 }
```

(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 `l3pdftools`.

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

`_hyp_text_string_from_unicode:nN`

```
575 \cs_new_protected:Npn \_hyp\_text\_string\_from\_unicode:nN #1 #2
576 {
577   \pdf\_string\_from\_unicode:nVN { #1 } #2 #2
578 }
```

(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`

```
579 \cs_new_protected:Npn \_hyp\_text\_pdfstring:nnN #1 #2 #3
580 {
581   \group\_begin:
582   \Hy@pdfstringtrue
583   \hook\_use:n {hyp/text/pdfstring}
584   \_hyp\_text\_purify:nN { #1 } \l\_hyp\_text\_tmpa\_str
585   \_hyp\_text\_cleanup:N \l\_hyp\_text\_tmpa\_str
586   \_hyp\_text\_string\_from\_unicode:nN { #2 } \l\_hyp\_text\_tmpa\_str
587   \str\_gset\_eq:NN \g\_hyp\_text\_tmpa\_str\l\_hyp\_text\_tmpa\_str
588   \group\_end:
589   \str\_set\_eq:NN #3 \g\_hyp\_text\_tmpa\_str
590 }
591 \cs\_generate\_variant:Nn \_hyp\_text\_pdfstring:nnN {enN,onN,eoN,ooN,noN}
```

(End of definition for _hyp_text_pdfstring:nnN.)

!!! temporary until all instances are gone

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

This is a special version for info keys:

`_hyp_text_pdfstring_info:nN`

```
594 \cs\_new\_protected:Npn \_hyp\_text\_pdfstring\_info:nN #1 #2
595 {
596   \_hyp\_text\_pdfstring:noN { #1 }{\l\_hyp\_text\_enc\_info\_print\_tl } #2
597 }
598 \cs\_generate\_variant:Nn \_hyp\_text\_pdfstring\_info:nN {eN,oN}
```

(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 <</Nums[0<</S/D>>]>>` (but it would be possible), we also don't test for empty `\thepage`, `hyperref` seems to handle this fine and the pdf is valid.

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

```
\__hyp_PageLabels_gpush:
  \Hy@PutCatalog
  \HyPL@StorePageLabel
599 \cs_new_protected:Npn\__hyp_PageLabels_gpush:
600   {
601     \pdfmanagement_add:nne {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
602   }
603
604 \def\Hy@PutCatalog #1 {}
605
606
607 \legacy_if:nT { Hy@pdfpagelabels }
608   {
609     \cs_set_protected:Npn \HyPL@StorePageLabel #1
610       {
611         \tl_gput_right:Ne \HyPL@Labels { \the\Hy@abspage<<#1>> }
612         \__hyp_PageLabels_gpush:
613       }
614   }
```

(End of definition for `__hyp_PageLabels_gpush:`, `\Hy@PutCatalog`, and `\HyPL@StorePageLabel`.)

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 Link level

Links can be nested. Inner links need perhaps special handling, e.g. to deactivate the link, or to change the border, or in the case of tagging to add some additional structure to handle the parent-child rules. We therefore add a global counter which is increased at the begin of link and decreased at the end.

`g__hyp_linknestlevel_int`

```

615 \int_new:N \g__hyp_linknestlevel_int
(End of definition for g__hyp_linknestlevel_int.)
616 \prg_new_conditional:Npnn \__hyp_if_outer_link: {TF}
617 {
618   \int_compare:nNnTF { \g__hyp_linknestlevel_int } > {1}
619     { \prg_return_false: }
620     { \prg_return_true: }
621 }
622 \cs_new:Npn \__hyp_check_link_nesting:TF #1 #2
623 {
624   \use_i:nn {#1}{#2}
625 }
626 \keys_define:nn { hyp }
627 {
628   nested-links .choice:,
629   nested-links / true .code:n =
630     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \use_i:nn },
631   nested-links / false .code:n =
632     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \__hyp_if_outer_link:TF },
633   nested-links .default:n = {true}
634 }

```

10.2 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 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`

```

635 \Hy@WrapperDef \__hyp_destination:nn #1 #2
636 {
637   \mode_if_horizontal:T { \@savsf\spacefactor }
638   \Hy@SaveLastskip      %defined in hyperref
639   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging

```



```

640   \_hyp_text_pdfstring:eON
641     { \HyperDestNameFilter{#1} }
642     { \l__hyp_text_enc_dest_tl }
643     \l__hyp_tmpa_tl
644   \str_if_eq:nnTF {#2} {fitrbox}
645     {
646       \exp_args:NV
647       \pdf_destination:nnnn \l__hyp_tmpa_tl
648       { \box_wd:N \l__hyp_dest_box }
649       { \box_ht:N \l__hyp_dest_box }
650       { \box_dp:N \l__hyp_dest_box }
651     }
652     {
653       \exp_args:NV
654       \pdf_destination:nf
655       { \l__hyp_tmpa_tl }
656       { #2 }
657     }
658   \Hy@RestoreLastskip   %defined in hyperref
659   \mode_if_horizontal:T { \spacefactor\@savsf }
660 }

```

(End of definition for _hyp_destination:nn.)

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

```

\hyper@anchor
\hyper@anchorstart 661 \tl_new:N\l_hyp_current_dest_name_tl
\hyper@anchorend 662 \hook_new:n{hyp/anchor}
hyp/anchor 663 \cs_new_protected:Npn \hyper@anchor #1
\l_hyp_current_dest_name_tl 664 {
665   \exp_args:NnV
666   \_hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
667   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
668   \hook_use:n{hyp/anchor}
669 }
670
671 \cs_new_protected:Npn \hyper@anchorstart #1
672 {
673   \Hy@activeanchortrue
674   \exp_args:NnV
675   \_hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
676   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
677   \hook_use:n{hyp/anchor}
678 }
679
680 \cs_new_protected:Npn \hyper@anchorend
681 {
682   \Hy@activeanchorfalse
683 }

```

(End of definition for `\hyper@anchor` and others.)

10.3 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:

```
684 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
685 {
686   \mode_leave_vertical:
687   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
688   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
689     {
690       \msg_warning:nne
691         { hyp }
692         { empty-destination-name }
693         { \c__hyp_dest_undefined_tl }
694       \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
695     }
696     {
697       \__hyp_text_pdfstring:eoN
698       { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
699       { \l__hyp_text_enc_dest_tl }
700       \l__hyp_dest_name_tmpa_tl
701     }
702   \exp_args:No
703     \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
704 }
705
706 \cs_new_protected:Npn \__hyp_link_goto_end:
707 {
708   \pdfannot_link_goto_end:
709 }
```

Now the three `hyperref` commands. The split 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

```

710
711 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
712 {
713   \bool_if:NTF \l__hyp_annot_GoTo_bool
714   {
715     \int_gincr:N\g__hyp_linknestlevel_int
716     \__hyp_check_link_nesting:TF
717     {
718       \Hy@VerboseLinkStart{#1}{#2}
719       \group_begin:
720       \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
721       \socket_use:nmn{hyp/link/GoTo/Contents}{#2}{#3}
722       \hook_use:n {hyp/link/#1}
723       \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
724       \__hyp_link_goto_end:
725       \group_end:
726       \Hy@VerboseLinkStop
727     }
728     {
729       \group_begin: #3\group_end:
730     }
731     \int_gdecr:N\g__hyp_linknestlevel_int
732   }
733   {\let\protect\relax#3}
734 }
735 \cs_new_protected:Npn \hyper@linkstart #1 #2 % #1 context, #2=destination name
736 {
737   \bool_if:NT \l__hyp_annot_GoTo_bool
738   {
739     \int_gincr:N\g__hyp_linknestlevel_int
740     \__hyp_check_link_nesting:TF
741     {
742       \Hy@VerboseLinkStart{#1}{#2}% only for debug
743       \group_begin:
744       \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
745       \socket_use:nmn{hyp/link/GoTo/Contents}{#2}{#3}
746       \hook_use:n {hyp/link/#1}
747       \__hyp_link_goto_begin:nw {#2}
748     }
749     {
750       \group_begin:
751     }

```

```

752     }
753   }
754
755   \cs_new_protected:Npn \hyper@linkend
756   {
757     \bool_if:NT \l__hyp_annot_GoTo_bool
758     {
759       \__hyp_check_link_nesting:TF
760       {
761         \__hyp_link_goto_end:
762         \group_end:
763         \Hy@VerboseLinkStop
764       }
765       {
766         \group_end:
767       }
768     }
769   }
770 }

```

10.4 URI links

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

```

771 \pdfdict_new:n {l_hyp/annot/A/URI}
772 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
773 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
774
775 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
776 {
777   \bool_if:NTF \l__hyp_annot_URI_bool
778   {
779     \int_gincr:N\g__hyp_linknestlevel_int
780     \__hyp_check_link_nesting:TF
781     {
782       \group_begin:
783       \__hyp_text_pdfstring:eoN
784       { #2}
785       { \l__hyp_text_enc_uri_print_tl }
786       \l__hyp_uri_tmpa_tl
787       \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
788       \bool_if:NT \l__hyp_href_url_ismap_bool
789       {
790         \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
791       }

```

This socket adds something to the /Contents key.

```

792     \socket_use:nn{hyp/link/URI/Contents}{#2}
793     \cs_set_eq:NN \# \c_hash_str
794     \cs_set_eq:NN \% \c_percent_str
795     \Hy@safe@activetrue
796     \mode_leave_vertical:
797     \pdfannot_dict_put:nne {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
798     \pdfannot_link:nen { URI }
799     {

```

```

800     }
801     {
802     \let\protect\relax
803     #1
804     \Hy@xspace@end
805     \Hy@VerboseLinkStop %where is the start??
806     }
807     \group_end:
808     }
809     {
810     \group_begin: #1 \group_end:
811     }
812     \int_gdecr:N\g__hyp_linknestlevel_int
813     }
814     {\let\protect\relax#1}}
815 }
816

```

10.5 GoToR Links files

```

817 \pdfdict_new:n {l_hyp/annot/A/GoToR}
818 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
819 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
820
821 \cs_generate_variant:Nn \pdffile_embed_file:nnn {noe}
822 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
823 {
824 \bool_if:NTF \l__hyp_annot_GoToR_bool
825 {
826 \int_gincr:N\g__hyp_linknestlevel_int
827 \__hyp_check_link_nesting:TF
828 {
829 \group_begin:
830 \tl_set:Ne \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
831 \exp_args:Ne
832 \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
833 {
834 \pdfdict_put:nne { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n}
835 \pdffile_embed_file:noe
836 {}
837 {\l__hyp_filename_tmpa_tl }
838 {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
839 }
840 \pdfdict_put:nne
841 {l_hyp/annot/A/GoToR}
842 {F}
843 {\pdf_object_ref:e {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
844 \__hyp_text_pdfstring:nnN
845 { #3 }
846 { \l__hyp_text_enc_dest_print_tl }
847 \l__hyp_dest_name_tmpa_tl
848 \tl_if_blank:eTF {#3}
849 {
850 \pdfdict_put:nne {l_hyp/annot/A/GoToR}{D}

```

```

851         {
852         [
853         \int_eval:n
854         { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
855         /\l__hyp_dest_pdfremotestartview_tl
856         ]
857         }
858     }
859     {
860     \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
861     }
862     \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>

```

863     \pdf_object_unnamed_write:ne{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
864     \pdfannot_dict_put:nne {link/GoToR}{A}{\pdf_object_ref_last:}
865     \pdfannot_link:nnn %expansion??
866     { GoToR }
867     {
868     }
869     {
870     \let\protect\relax
871     #1\Hy@xspace@end
872     \Hy@VerboseLinkStop %where is the start??
873     }
874     \group_end:
875     }
876     {
877     \group_begin: #1 \group_end:
878     }
879     \int_gdecr:N\g__hyp_linknestlevel_int
880     }
881     {{\let\protect\relax#1}}
882     }

```

10.6 Launch links

We define `\hyper@linklaunch` for naming consistency

```

883 \pdfdict_new:n {l_hyp/annot/A/Launch}
884 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
885 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
886
887 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
888 {
889     \bool_if:NTF \l__hyp_annot_Launch_bool
890     {
891         \int_gincr:N\g__hyp_linknestlevel_int
892         \__hyp_check_link_nesting:TF
893         {
894             \group_begin:
895             \__hyp_text_pdfstring:nnN
896             { #1 }
897             { \l__hyp_text_enc_file_print_tl }
898             \l__hyp_filename_tmpa_tl

```

```

899         \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
900         \__hyp_text_pdfstring:noN
901         { #3 }
902         { \l__hyp_text_enc_para_print_tl }
903         \l__hyp_para_tmpa_tl
904         \bool_if:nTF
905         {
906             \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}{}
907             ||
908             \pdf_version_compare_p:Nn > {1.9}
909         }
910         {
911             \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
912         }
913         {
914             \pdfdict_put:nne
915             {l_hyp/annot/A/Launch}
916             {Win}
917             {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
918         }
919         \mode_leave_vertical:
920         \pdfannot_dict_put:nne {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}
921         \pdfannot_link:nen
922         { Launch }
923         {
924             % /A
925             % <<
926             % \pdfdict_use:n {l_hyp/annot/A/Launch}
927             % >>
928         }
929         {
930             \let\protect\relax
931             #2\Hy@xspace@end
932             \Hy@VerboseLinkStop %where is the start??
933         }
934         \group_end:
935     }
936     { \group_begin: #2 \group_end: }
937     \int_gdecr:N\g__hyp_linknestlevel_int
938 }
939 {\let\protect\relax#2}}
940 }

```

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

```

941 \use:e
942 { % filename, anchor text, linkname
943     \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str #1 \exp_not:N \ \ #2 #3
944 }
945 {
946     \hyper@linklaunch {#1}{#2}{#3}
947 }

```

10.7 Named links (menu)

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

```
948 \pdfdict_new:n {l_hyp/annot/A/Named}
949 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
950 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
951
952 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
953 {
954   \bool_if:NTF \l__hyp_annot_Named_bool
955   {
956     \int_gincr:N\g__hyp_linknestlevel_int
957     \__hyp_check_link_nesting:TF
958     {
959       \group_begin:
960       \pdfmeta_standard_verify:nnTF {named_actions}{#1}
961       {
962         \mode_leave_vertical:
963         \pdfdict_put:nne {l_hyp/annot/A/Named}{N}
964           {\pdf_name_from_unicode_e:n{#1}}
965         \pdfannot_dict_put:nne {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}
966         \pdfannot_link:nnn { Named }
967         {
968           % /A
969           % <<
970           % \pdfdict_use:n { l_hyp/annot/A/Named }
971           % >>
972         }
973         {
974           #2
975           \Hy@xspace@end
976           \Hy@VerboseLinkStop
977         }
978       }
979       {
980         \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
981         #2
982       }
983       \group_end:
984     }
985     { \group_begin: #2 \group_end: }
986     \int_gdecr:N\g__hyp_linknestlevel_int
987   }
988   {\let\protect\relax#2}}
989 }
990
```

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.

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

```
\_hyp_color_select:n \_hyp_color_select:n {<color>}
```

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

```
\_hyp_color_select:n \_hyp_color_select_aux:wn
```

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

```
992 \cs_new_protected:Npn \_hyp_color_select:n #1
993 {
994   \tl_if_head_eq_charcode:nNTF {#1}[ %]
995   {
996     \_hyp_color_select_aux:wn #1
997   }
998   {
999     \color_select:n {#1}
1000   }
1001 }
1002
1003 \cs_new_protected:Npn \_hyp_color_select_aux:wn [#1] #2
1004 {
1005   \color_select:nn {#1}{#2}
1006 }
1007
1008 \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 \_hyp_color_set_aux:nwn
```

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

```
1009 \cs_new_protected:Npn \_hyp_color_set:nn #1 #2
1010 {
1011   \tl_if_head_eq_charcode:nNTF {#2}[ %]
1012   {
1013     \_hyp_color_set_aux:nwn { #1 } #2
1014   }
1015   {
1016     \color_set:nn {#1} {#2}
1017   }
1018 }
1019
1020 \cs_new_protected:Npn \_hyp_color_set_aux:nwn #1 [#2] #3
1021 {
1022   \color_set:nnn {#1}{#2}{#3}
1023 }
1024
1025 \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.

```
1026 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1027   {
1028     \hook_gput_code:nnn
1029       {pdfannot/link/#2/begin}
1030       {hyp/color}
1031       {
1032         \bool_if:cT { l_hyp_annot_color#1_bool }
1033         {
1034           \group_begin:
1035           \color_select:n { hyp/color/#1}
1036         }
1037       }
1038     \hook_gput_code:nnn
1039       {pdfannot/link/#2/end}
1040       {hyp/color}
1041       {
1042         \bool_if:cT { l_hyp_annot_color#1_bool }
1043         {
1044           \group_end:
1045         }
1046       }
1047   }
```

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

```
1048 \keys_define:nn { hyp }
1049   {
1050     ,colorlinks .choice:
1051     ,colorlinks / true .meta:n =
1052       {
1053         ,pdfborder={0~0~0}
1054         ,pdfborderstyle=
1055         ,colorurl =#1
1056         ,colorlink =#1
1057         ,colorryn =#1
1058         ,colormenu =#1
1059         ,colorfile =#1
1060       }
1061     ,colorlinks / false .meta:n =
1062       {
1063         ,colorurl =#1
1064         ,colorlink =#1
1065         ,colorryn =#1
1066         ,colormenu =#1
1067         ,colorfile =#1
1068       }
1069     ,colorlinks .default:n = {true}
1070   }
```

```

colorurl (setup key)
colorlink (setup key) 1071 \seq_map_inline:Nn \c__hyp_annot_types_seq
colorrun (setup key) 1072 {
colormenu (setup key) 1073   \keys_define:nn { hyp }
colorfile (setup key) 1074   {
  urlcolor (setup key) 1075     ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
linkcolor (setup key) 1076     ,#1color .code:n =   { \__hyp_color_set:ne {hyp/color/#1}{##1} }
  runcolor (setup key) 1077   }
menucolor (setup key) 1078 }
filecolor (setup key) 1079
allcolors (setup key) 1080 \keys_define:nn { hyp }
1081 {
1082   ,allcolors .meta:n =
1083   {
1084     ,urlcolor=#1
1085     ,linkcolor=#1
1086     ,runcolor=#1
1087     ,filecolor=#1
1088     ,menucolor=#1
1089   }
1090   ,allcolors .value_required:n = true
1091 }

```

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)

```

1092 \keys_define:nn { hyp }
1093 {
1094   ,bordercolormodel .choices:nn =
1095   {rgb,cmyk}
1096   { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
1097   ,bordercolormodel .initial:n ={rgb}
1098 }

1099 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1100 {
1101   \keys_define:nn { hyp }
1102   {
1103     #1bordercolor .code:n =
1104     {
1105       \tl_if_empty:nTF { ##1 }
1106       {
1107         \pdfannot_dict_remove:nn
1108         {link/#2}
1109         { C }
1110       }
1111       {
1112         \__hyp_color_set:ne {hyp/color/#1border}{##1}
1113         \color_export:nVN

```

```

1114         {hyp/color/#1border}
1115         \g__hyp_bordercolormodel_str
1116         \l__hyp_tmpa_tl
1117         \pdfannot_dict_put:nne
1118         {link/#2}
1119         { C }
1120         { [\l__hyp_tmpa_tl] }
1121     }
1122 }
1123 }
1124 }
1125
1126 \keys_define:nn { hyp }
1127 {
1128     ,allbordercolors .meta:n =
1129     {
1130         ,linkbordercolor=#1
1131         ,urlbordercolor =#1
1132         ,filebordercolor=#1
1133         ,menubordercolor=#1
1134         ,runbordercolor =#1
1135     }
1136     ,allbordercolors .value_required:n = true
1137 }
1138

```

11.3.2 Borderwidth and -arc

```

1139 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1140 {
1141     \keys_define:nn { hyp }
1142     {
1143         #1border .code:n =
1144         {
1145             \tl_if_empty:nTF { ##1 }
1146             {
1147                 \pdfannot_dict_remove:nn
1148                 {link/#2}
1149                 { Border }
1150             }
1151             {
1152                 \pdfannot_dict_put:nnn
1153                 {link/#2}
1154                 { Border }
1155                 { [##1] }
1156             }
1157         }
1158     }
1159 }
1160 \keys_define:nn { hyp }
1161 {
1162     ,pdfborder .code:n =
1163     {
1164         \tl_if_empty:nTF { #1 }

```

```

1165     {
1166       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1167       {
1168         \pdfannot_dict_remove:nn
1169         {link/##2}
1170         { Border }
1171       }
1172     }
1173     {
1174       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1175       {
1176         \pdfannot_dict_put:nnn
1177         {link/##2}
1178         { Border }
1179         { [#1] }
1180       }
1181     }
1182   }
1183   ,pdfborder .initial:n = {0~0~1},
1184 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1185 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1186 {
runborderstyle (setup key) 1187   \keys_define:nn { hyp }
fileborderstyle (setup key) 1188   {
menuborderstyle (setup key) 1189     #1borderstyle .code:n =
1190     {
1191       \tl_if_empty:nTF { ##1 }
1192       {
1193         \pdfannot_dict_remove:nn
1194         {link/#2}
1195         { BS }
1196       }
1197       {
1198         \pdfannot_dict_put:nnn
1199         {link/#2}
1200         { BS }
1201         { <<##1>> }
1202       }
1203     }
1204   }
1205 }
1206 \keys_define:nn { hyp }
1207 {
1208   ,pdfborderstyle .code:n =
1209   {
1210     \tl_if_empty:nTF { #1 }
1211     {
1212       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1213       {

```

```

1214         \pdfannot_dict_remove:nn
1215         {link/##2}
1216         { BS }
1217     }
1218 }
1219 {
1220     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1221     {
1222         \pdfannot_dict_put:nnn
1223         {link/##2}
1224         { BS }
1225         { <<#1>> }
1226     }
1227 }
1228 }
1229 ,pdfborderstyle .initial:n = {},
1230 }

```

11.4 ocolorlinks

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

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

```

1231 \cs_new_protected:Npn \__hyp_ocg_init:
1232 {
1233     \pdf_object_new:n { __hyp/OCG/View }
1234     \pdf_object_new:n { __hyp/OCG/Print }
1235     \pdf_object_new:n { __hyp/OCG/config }
1236     \pdf_object_new:n { __hyp/OCG/refarray }
1237     \pdf_object_write:mne { __hyp/OCG/refarray } { array }
1238     {
1239         \pdf_object_ref:n { __hyp/OCG/View }
1240         \c_space_tl
1241         \pdf_object_ref:n { __hyp/OCG/Print }
1242     }
1243     \pdf_object_write:nnn { __hyp/OCG/View } { dict }
1244     {
1245         /Type/OCG
1246         /Name(View)
1247         /Usage
1248         <<
1249         /Print <</PrintState/OFF>>~
1250         /View <</ViewState/ON >>~
1251         >>
1252     }
1253     \pdf_object_write:nnn { __hyp/OCG/Print } { dict }
1254     {
1255         /Type/OCG
1256         /Name(Print)
1257         /Usage
1258         <<
1259         /Print <</PrintState/ON>>~

```

```

1260         /View <</ViewState/OFF>>~
1261     >>
1262     }
1263     \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG} }
1264     \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG} }
1265     \pdf_object_write:nne { __hyp/OCG/config } { dict }
1266     {
1267         /OFF[\pdf_object_ref:n { __hyp/OCG/Print }]
1268         /AS[
1269             <<
1270                 /Event/View
1271                 /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1272                 /Category[/View]
1273             >>
1274             <<
1275                 /Event/Print
1276                 /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1277                 /Category[/Print]
1278             >>
1279             <<
1280                 /Event/Export
1281                 /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1282                 /Category[/Print]
1283             >>
1284             ]
1285         }
1286     \pdfmanagement_add:nne { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG} }
1287     \cs_gset:Npn \__hyp_ocg_init: {}
1288 }

```

(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.

```

1289 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1290 {
1291     \hook_gput_code:nnn
1292     {pdfannot/link/#2/begin}
1293     {hyp/ocg}
1294     {
1295         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1296         {
1297             \__hyp_ocg_init:
1298             \group_begin:
1299             \hbox_set:Nw \l__hyp_tmpa_box
1300         }
1301     }
1302     \hook_gput_code:nnn
1303     {pdfannot/link/#2/end}
1304     {hyp/ocg}
1305     {
1306         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1307         {
1308             \hbox_set_end:
1309             \mbox

```

```

1310         {
1311         \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1312         \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1313         \pdf_emc:
1314         \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1315         \group_begin:
1316         \color_select:n { hyp/color/#1 }
1317         \box_use_drop:N \l__hyp_tmpa_box
1318         \group_end:
1319         \pdf_emc:
1320         }
1321     \group_end:
1322 }
1323 }
1324 }

```

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

```

ocgcolorlink (setup key) 1325 \bool_lazy_or:nnTF
  ocgcolorurl (setup key) 1326 { \pdf_version_compare_p:Nn > {1.4} }
ocgcolorfile (setup key) 1327 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
ocgcolormenu (setup key) 1328 {
  ocgcolorrund (setup key) 1329 \keys_define:nn { hyp }
1330     {
1331     ,_ocgcolorlinks .meta:n =
1332     {
1333     ocgcolorlink=#1,
1334     ocgcolorurl=#1,
1335     ocgcolorfile=#1,
1336     ocgcolorrund=#1,
1337     ocgcolormenu=#1
1338     }
1339     ,_ocgcolorlinks .default:n = true
1340     }
1341 }
1342 {
1343 \keys_define:nn { hyp }
1344 {
1345 ,_ocgcolorlinks .code:n =
1346 {
1347 \msg_warning:nnee
1348 { hyp }
1349 { ignore-deprecated-or-unknown-option-in-pdf-version }
1350 { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1351 }
1352 }
1353 }
1354
1355 \keys_define:nn { hyp }
1356 {
1357 ,ocgcolorlinks .choice:
1358 ,ocgcolorlinks / true .meta:n =
1359 {
1360 pdfborder      ={{0~0~0}},
1361 pdfborderstyle ={},

```



```

1362     colorlinks      = false,
1363     _ocgcolorlinks = true
1364   }
1365   ,ocgcolorlinks / false .meta:n =
1366   {
1367     _ocgcolorlinks = false
1368   }
1369   ,ocgcolorlinks .default:n = {true}
1370 }
1371
1372 \seq_map_inline:Nn \c__hyp_annot_types_seq
1373 {
1374   \bool_lazy_or:nnTF
1375   { \pdf_version_compare_p:Nn > {1.4} }
1376   { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1377   {
1378     \keys_define:nn { hyp }
1379     {
1380       ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1381     }
1382   }
1383   {
1384     \keys_define:nn { hyp }
1385     {
1386       ,ocgcolor#1 .code:n=
1387       {
1388         \msg_warning:nnee
1389         { hyp }
1390         { ignore-deprecated-or-unknown-option-in-pdf-version }
1391         { ocgcolor#1 }
1392         { \pdf_version_major:.\pdf_version_minor: }
1393       }
1394     }
1395   }
1396 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1397 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1398 {
1399   \keys_define:nn { hyp }
1400   {
1401     ,#1highlight .choices:nn =
1402     { /I, /N, /O, /P}
1403     {
1404       \pdfannot_dict_put:nnn
1405       {link/#2}
1406       { H }
1407       { ##1 }
1408     }
1409     ,#1highlight / .code:n =
1410     {

```

```

1412         \pdfannot_dict_remove:nn
1413         {link/#2}
1414         { H }
1415
1416     }
1417     ,#1highlight / unknown .code:n =
1418     {
1419         \msg_warning:nnee { hyp } { unknown-choice+empty }
1420         { #1highlight }
1421         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1422         { \exp_not:n {##1} }
1423     }
1424 }
1425 }
1426
1427
1428 \keys_define:nn { hyp }
1429 {
1430     ,pdfhighlight .choices:nn =
1431     { /I, /N, /O, /P}
1432     {
1433         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1434         {
1435             \pdfannot_dict_put:nnn
1436             {link/###2}
1437             { H }
1438             { #1 }
1439         }
1440     }
1441     ,pdfhighlight / .code:n =
1442     {
1443         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1444         {
1445             \pdfannot_dict_remove:nn
1446             {link/##2}
1447             { H }
1448         }
1449     }
1450     ,pdfhighlight .initial:n = {/I},
1451     ,pdfhighlight / unknown .code:n =
1452     {
1453         \msg_warning:nnee { hyp } { unknown-choice+empty }
1454         { pdfhighlight }
1455         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1456         { \exp_not:n {##1} }
1457     }
1458 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1459 \keys_define:nn { hyp }
hideurl (setup key)
hidefile (setup key)
hiderun (setup key)
hidemenu (setup key)

```

```

1460 {
1461   hidelinks .meta:n =
1462   {
1463     ,colorlinks      = false
1464     ,ocgcolorlinks  = false
1465     ,pdfborder      = { 0~0~0 }
1466     ,pdfborderstyle=
1467   }
1468 }
1469
1470 \seq_map_inline:Nn \c__hyp_annot_types_seq
1471 {
1472   \keys_define:nn { hyp }
1473   {
1474     hide#1 .meta:n =
1475     {
1476       ,color#1      = false
1477       ,ocgcolor#1  = false
1478       ,#1border    = { 0~0~0 }
1479       ,#1borderstyle =
1480     }
1481   }
1482 }

```

11.7 color schemes and settings

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

`colorscheme` (*setup key*)

```

1483 \keys_define:nn { hyp }
1484 {
1485   colorscheme .code:n =
1486   {
1487     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1488     {
1489       \keys_set:nn { hyp }
1490       {
1491         ##1 = ##2
1492       }
1493     }
1494   }
1495 }
1496 \keys_set:nn { hyp } {colorscheme=phelype}

```

12 Keys

12.1 Ignored keys

The following are ignored (with or without warnings)

`unicode` (*setup key*)

`pdfencoding` (*setup key*)

`pdfversion` (*setup key*)

```

1497 \keys_define:nn { hyp }
1498 {

```

```

1499     ,unicode      .code:n = {}
1500     ,pdfencoding .code:n = {}
1501     ,pdfversion  .code:n =
1502     {
1503         \msg_warning:nn { hyp }{ pdfversion-disabled }
1504     }
1505 }
1506 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
debug (setup key) 1507 \keys_define:nn { hyp }
draft (setup key) 1508 {
final (setup key) 1509     ,verbose .legacy_if_set:n = {Hy@verbose}
1510     ,debug .legacy_if_set:n = {Hy@verbose}
1511 }
1512 \keys_define:nn { hyp }
1513 {
1514     ,draft .code:n =
1515     {
1516         \Hy@drafttrue
1517         \PassOptionsToPackage{draft}{bookmark}
1518     }
1519     ,final .code:n =
1520     {
1521         \Hy@finaltrue
1522         \PassOptionsToPackage{final}{bookmark}
1523     }
1524 }

extension (setup key)
hypertextnames (setup key) 1525 \keys_define:nn { hyp }
naturalnames (setup key) 1526 {
pageanchor (setup key) 1527     ,extension .tl_set:N = \XR@ext
linktoc (setup key) 1528     ,extension .initial:n= pdf
linktocpage (setup key) 1529     ,hypertextnames .legacy_if_set:n = {Hy@hypertextnames}
plainpages (setup key) 1530     ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
localanchorname (setup key) 1531     ,localanchorname .legacy_if_set:n = {Hy@localanchorname}
linkfileprefix (setup key) 1532     ,naturalnames .legacy_if_set:n = {Hy@naturalnames}
1533     ,pageanchor .legacy_if_set:n = {Hy@pageanchor}
1534     ,plainpages .legacy_if_set:n = {Hy@plainpages}
1535 }
1536
1537 \keys_define:nn { hyp }
1538 {
1539     ,linktoc .choices:nn = { none, section, all, page }
1540     {
1541         \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1542     }
1543     ,linktoc / unknown .code:n =
1544     {

```

```

1545     \msg_warning:nneee { hyp } { unknown-choice }
1546     { linktoc }
1547     { none, section, all, page }
1548     { \exp_not:n {#1} }
1549   }
1550   ,linktocpage .choice:
1551   ,linktocpage / true .meta:n = {linktoc=page}
1552   ,linktocpage / false .meta:n = {linktoc=section}
1553   ,linktocpage .default:n = true
1554 }
1555

```

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

```

url (setup key) 1556 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1557 {
menu (setup key) 1558   \keys_define:nn { hyp }
run (setup key) 1559   {
1560     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1561   }
1562 }

```

```

1563 \keys_define:nn { hyp }
1564 {
1565   ,baseurl .code:n =
1566   {
1567     \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1568     \tl_if_empty:NTF \l__hyp_tmpa_tl
1569     {
1570       \pdfmanagement_remove:nn {Catalog} { URI }
1571     }
1572     {
1573       \pdfmanagement_add:nne {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1574     }
1575     \__hyp_store_metadata:nn {baseurl}{#1}
1576   }
1577   %only false does something ...
1578   ,bookmarks .choice:
1579   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1580   ,bookmarks / true .code:n = {}
1581   ,bookmarks .default:n = {true}
1582   ,bookmarksnumbered .legacy_if_set:n = {Hy@bookmarksnumbered}
1583   ,bookmarksopen .legacy_if_set:n = {Hy@bookmarksopen}
1584   ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1585   ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1586   ,pdfcenterwindow .choice:
1587   ,pdfcenterwindow / false .code:n =
1588   {
1589     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1590   }
1591   ,pdfcenterwindow / true .code:n =
1592   {
1593     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1594   }
1595   ,pdfcenterwindow / .code:n =

```

```

1596     {
1597     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1598     }
1599 ,pdfcenterwindow / unknown .code:n =
1600     {
1601     \msg_warning:nnee { hyp } { no-bool }
1602     { pdfcenterwindow }
1603     { \exp_not:n {#1} }
1604     }
1605 ,pdfcenterwindow .default:n = true
1606 ,pdfdirection .choice:
1607 ,pdfdirection / L2R .code:n =
1608     {
1609     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1610     }
1611 ,pdfdirection / R2L .code:n =
1612     {
1613     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1614     }
1615 ,pdfdirection / .code:n =
1616     {
1617     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1618     }
1619 ,pdfdirection / unknown .code:n =
1620     {
1621     \msg_warning:nneee { hyp } { unknown-choice+empty }
1622     { pdfdirection }
1623     { L2R , R2L }
1624     { \exp_not:n {#1} }
1625     }
1626 ,pdfdisplaydoctitle .choice:
1627 ,pdfdisplaydoctitle / false .code:n =
1628     {
1629     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1630     }
1631 ,pdfdisplaydoctitle / true .code:n =
1632     {
1633     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1634     }
1635 ,pdfdisplaydoctitle .default:n = true
1636 ,pdfduplex .choices:nn =
1637 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1638     {
1639     \pdf_version_compare:NnTF > {1.6}
1640     {
1641     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1642     { PrintDuplex } { /#1 }
1643     }
1644     {
1645     \msg_warning:nnee
1646     {hyp}
1647     {ignore-deprecated-or-unknown-option-in-pdf-version}
1648     {pdfduplex}
1649     {\pdf_version:}

```

```

1650     }
1651   }%
1652 ,pdfduplex / .code:n =
1653   {
1654     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1655   }
1656 ,pdfduplex / unknown .code:n =
1657   {
1658     \msg_warning:nnee { hyp } { unknown-choice+empty }
1659     { pdfduplex }
1660     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1661     { \exp_not:n {#1} }
1662   }
1663 ,pdffitwindow .choice:
1664 ,pdffitwindow / false .code:n =
1665   {
1666     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1667   }
1668 ,pdffitwindow / true .code:n =
1669   {
1670     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1671   }
1672 ,pdffitwindow / .code:n =
1673   {
1674     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1675   }
1676 ,pdffitwindow .default:n = true
1677 ,pdffitwindow / unknown .code:n =
1678   {
1679     \msg_warning:nnee { hyp } { no-bool }
1680     { pdffitwindow }
1681     { \exp_not:n {#1} }
1682   }
1683 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1684 ,pdflinkmargin .initial:n = {1pt}
1685 ,pdfmenubar .choice:
1686 ,pdfmenubar / true .code:n =
1687   {
1688     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1689   }
1690 ,pdfmenubar / false .code:n =
1691   {
1692     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1693     { HideMenubar } { true }
1694   }
1695 ,pdfmenubar / .code:n =
1696   {
1697     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1698   }
1699 ,pdfmenubar .default:n = true
1700 ,pdfmenubar / unknown .code:n =
1701   {
1702     \msg_warning:nnee { hyp } { no-bool }
1703     { pdfmenubar }

```

```

1704         { \exp_not:n {#1} }
1705     }
1706 ,pdfnewwindow .choice:
1707 ,pdfnewwindow / true .code:n =
1708     {
1709         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1710         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1711     }
1712 ,pdfnewwindow / false .code:n =
1713     {
1714         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1715         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1716     }
1717 ,pdfnewwindow / .code:n =
1718     {
1719         \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1720         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1721     }
1722 ,pdfnonfullscreenpagemode .choices:nn =
1723 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1724 {
1725     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1726     { NonFullScreenPageMode } {/#1}
1727 }
1728 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1729 {
1730     \pdf_version_compare:NnTF < {1.6}
1731     {
1732         %message
1733     }
1734     {
1735         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1736         {NonFullScreenPageMode}{/UseAttachments}
1737     }
1738 }
1739 ,pdfnonfullscreenpagemode / .code:n =
1740 {
1741     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1742 }
1743 ,pdfnonfullscreenpagemode / unknown .code:n =
1744 {
1745     \msg_warning:nnee { hyp } { unknown-choice+empty }
1746     { pdfnonfullscreenpagemode }
1747     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1748     { \exp_not:n {#1} }
1749 }
1750 ,pdfnumcopies .code:n =
1751 {
1752     \pdf_version_compare:NnTF > {1.6}
1753     {
1754         \tl_if_empty:nTF {#1}
1755         {
1756             \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1757         }

```



```

1758         {
1759             \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1760             {NumCopies}{#1}
1761         }
1762     }
1763     {
1764         \msg_warning:nnee
1765         {hyp}
1766         {ignore-deprecated-or-unknown-option-in-pdf-version}
1767         {pdfnumcopies}
1768         {\pdf_version:}
1769     }
1770 }
1771 ,pdfpagelayout .choices:nn =
1772 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1773 { \pdfmanagement_add:nne {Catalog} { PageLayout }{ /#1 } }
1774 ,pdfpagelayout / .code:n =
1775 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1776 ,pdfpagelayout / unknown .code:n =
1777 {
1778     \msg_warning:nneee { hyp } { unknown-choice+empty }
1779     { pdfpagelayout }
1780     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1781     { \exp_not:n {#1} }
1782 }
1783 ,pdfpagemode .choices:nn =
1784 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1785 { \pdfmanagement_add:nne {Catalog} { PageMode }{ /#1 } }
1786 ,pdfpagemode / UseAttachments .code:n =
1787 {
1788     \pdf_version_compare:NnTF > {1.5}
1789     {
1790         \pdfmanagement_add:nne {Catalog} { PageMode }{ /UseAttachments }
1791     }
1792     {
1793         \msg_warning:nnee
1794         {hyp}
1795         {ignore-deprecated-or-unknown-value-in-pdf-version}
1796         {UseAttachments}
1797         {\pdf_version:}
1798     }
1799 }
1800 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1801 ,pdfpagemode / unknown .code:n =
1802 {
1803     \msg_warning:nneee { hyp } { unknown-choice+empty }
1804     { pdfpagemode }
1805     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1806     { \exp_not:n {#1} }
1807 }
1808 ,pdfpagescrop .code:n =
1809 {
1810     \tl_if_empty:nTF {#1} %or blank?
1811     {

```

```

1812         \pdfmanagement_remove:nn {Pages} { CropBox }
1813     }
1814     {
1815         \pdfmanagement_add:nne {Pages} { CropBox } { [#1] }
1816     }
1817 }
1818 ,pdfpicktraybypdfsize .choice:
1819 ,pdfpicktraybypdfsize / true .code:n =
1820 {
1821     \pdf_version_compare:NnTF > {1.6}
1822     {
1823         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1824         { PickTrayByPDFSize } { true }
1825     }
1826     {
1827         \msg_warning:nnee
1828         {hyp}
1829         {ignore-deprecated-or-unknown-option-in-pdf-version}
1830         {pdfpicktraybypdfsize}
1831         {\pdf_version:}
1832     }
1833 }
1834 ,pdfpicktraybypdfsize / false .code:n =
1835 {
1836     \pdf_version_compare:NnTF > {1.6}
1837     {
1838         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1839         { PickTrayByPDFSize } { false }
1840     }
1841     {
1842         \msg_warning:nnee
1843         {hyp}
1844         {ignore-deprecated-or-unknown-option-in-pdf-version}
1845         {pdfpicktraybypdfsize}
1846         {\pdf_version:}
1847     }
1848 }
1849 ,pdfpicktraybypdfsize / .code:n =
1850 {
1851     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1852 }
1853 ,pdfpicktraybypdfsize / unknown .code:n =
1854 {
1855     \msg_warning:nnee { hyp } { no-bool }
1856     { picktraybypdfsize }
1857     { \exp_not:n {#1} }
1858 }
1859 ,pdfprintarea .choices:nn =
1860 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1861 {
1862     \pdf_version_compare:NnTF < {2.0}
1863     {
1864         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1865         { PrintArea } { /#1 }

```

```

1866     }
1867     {
1868         \msg_warning:nnee
1869         {hyp}
1870         {ignore-deprecated-or-unknown-option-in-pdf-version}
1871         {pdfprintarea}
1872         {\pdf_version:}
1873     }
1874 }%
1875 ,pdfprintarea / .code:n =
1876 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1877 ,pdfprintarea / unknown .code:n =
1878 {
1879     \msg_warning:nneee { hyp } { unknown-choice+empty }
1880     { pdfprintarea }
1881     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1882     { \exp_not:n {#1} }
1883 }
1884 ,pdfprintclip .choices:nn =
1885 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1886 {
1887     \pdf_version_compare:NnTF < {2.0}
1888     {
1889         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1890         { PrintClip } { /#1 }
1891     }
1892     {
1893         \msg_warning:nnee
1894         {hyp}
1895         {ignore-deprecated-or-unknown-option-in-pdf-version}
1896         {pdfprintclip}
1897         {\pdf_version:}
1898     }
1899 }%
1900 ,pdfprintclip / .code:n =
1901 {
1902     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1903 }
1904 ,pdfprintclip / unknown .code:n =
1905 {
1906     \msg_warning:nneee
1907     { hyp }
1908     { unknown-choice+empty }
1909     { pdfprintclip }
1910     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1911     { \exp_not:n {#1} }
1912 }
1913 ,pdfprintpagerange .code:n =
1914 {
1915     \pdf_version_compare:NnTF > {1.6}
1916     {
1917         \tl_if_empty:nTF { #1}
1918         {
1919             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }

```

```

1920         { PrintPageRange }
1921     }
1922     {
1923         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1924         {PrintPageRange}{[#1]}
1925     }
1926 }
1927 {
1928     \msg_warning:nnee
1929     {hyp}
1930     {ignore-deprecated-or-unknown-option-in-pdf-version}
1931     {pdfprintpagerange}
1932     {\pdf_version:}
1933 }
1934 }
1935 ,pdfprintscaling .choices:nn =
1936 { None, AppDefault }
1937 {
1938     \pdf_version_compare:NnTF > {1.5}
1939     {
1940         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1941         { PrintScaling } { /#1 }
1942     }
1943     {
1944         \msg_warning:nnee
1945         {hyp}
1946         {ignore-deprecated-or-unknown-option-in-pdf-version}
1947         {pdfprintscaling}
1948         {\pdf_version:}
1949     }
1950 }%
1951 ,pdfprintscaling / .code:n =
1952 {
1953     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1954 }
1955 ,pdfprintscaling / unknown .code:n =
1956 {
1957     \msg_warning:nnee { hyp } { unknown-choice+empty }
1958     { pdfprintarea }
1959     { None, AppDefault }
1960     { \exp_not:n {#1} }
1961 }
1962 ,pdfremotestartview .code:n =
1963 {
1964     \tl_set:Ne \l__hyp_tmpa_tl {#1-null-null-null~}
1965     \exp_args:NNV
1966     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1967     {
1968         \tl_set:Ne \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1969     }
1970     {
1971         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1972         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1973     }

```

```

1974     }
1975 ,pdfremotestartview .initial:n = {Fit}
1976 % pdfstartpage is special as it shares code with pdfstartview
1977 ,pdfstartpage .code:n =
1978 {
1979     \tl_gset:Ne \g__hyp_dest_pdfstartpage_tl { #1 }
1980     \bool_if:nTF
1981     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1982     {
1983         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1984     }
1985     {
1986         \pdfmanagement_add:nne {Catalog} { OpenAction }
1987         {
1988             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1989         }
1990     }
1991 }
1992 ,pdfstartpage .initial:n =1
1993 ,pdfstartview .code:n =
1994 {
1995     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
1996     \exp_args:NNV
1997     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1998     {
1999         \tl_gset:Ne \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
2000     }
2001     {
2002         \msg_warning:nmmm {hyp}{invalid-destination-value}{#1}{pdfstartview}
2003         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
2004     }
2005     \bool_if:nTF
2006     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
2007     {
2008         \pdfmanagement_remove:nn {Catalog} { OpenAction }
2009     }
2010     {
2011         \pdfmanagement_add:nne {Catalog} { OpenAction }
2012         {
2013             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
2014         }
2015     }
2016 }
2017 ,pdfstartview .initial:n = Fit
2018 ,pdftoolbar .choice:
2019 ,pdftoolbar / true .code:n =
2020 {
2021     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
2022 }
2023 ,pdftoolbar / false .code:n =
2024 {
2025     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
2026     { HideToolbar } { true }
2027 }

```

```

2028 ,pdftoolbar / true .code:n =
2029 {
2030     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
2031 }
2032 ,pdftoolbar .default:n = true
2033 ,pdftoolbar / unknown .code:n =
2034 {
2035     \msg_warning:nnee { hyp } { no-bool }
2036     { pdftoolbar }
2037     { \exp_not:n {#1} }
2038 }
2039 % pdfview see below.
2040 ,pdfviewarea .choices:nn =
2041 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2042 {
2043     \pdf_version_compare:NnTF < {2.0}
2044     {
2045         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2046         { ViewArea } { /#1 }
2047     }
2048     {
2049         \msg_warning:nnee
2050         {hyp}
2051         {ignore-deprecated-or-unknown-option-in-pdf-version}
2052         {pdfviewarea}
2053         {\pdf_version:}
2054     }
2055 }%
2056 ,pdfviewarea / .code:n =
2057 {
2058     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
2059 }
2060 ,pdfviewarea / unknown .code:n =
2061 {
2062     \msg_warning:nneee { hyp } { unknown-choice+empty }
2063     { pdfviewarea }
2064     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2065     { \exp_not:n {#1} }
2066 }
2067 ,pdfviewclip .choices:nn =
2068 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2069 {
2070     \pdf_version_compare:NnTF < {2.0}
2071     {
2072         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2073         { ViewClip } { /#1 }
2074     }
2075     {
2076         \msg_warning:nnee
2077         {hyp}
2078         {ignore-deprecated-or-unknown-option-in-pdf-version}
2079         {pdfviewclip}
2080         {\pdf_version:}
2081     }

```

```

2082     }%
2083 ,pdfviewclip / .code:n =
2084     {
2085         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
2086     }
2087 ,pdfviewclip / unknown .code:n =
2088     {
2089         \msg_warning:nneee { hyp } { unknown-choice+empty }
2090         { pdfviewclip }
2091         { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2092         { \exp_not:n {#1} }
2093     }
2094 ,pdfwindowui .choice:
2095 ,pdfwindowui / true .code:n =
2096     {
2097         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2098     }
2099 ,pdfwindowui / false .code:n =
2100     {
2101         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2102         { HideWindowUI } { true }
2103     }
2104 ,pdfwindowui / .code:n =
2105     {
2106         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2107     }
2108 ,pdfwindowui / unknown .code:n =
2109     {
2110         \msg_warning:nnee { hyp } { no-bool }
2111         { pdfwindowui }
2112         { \exp_not:n {#1} }
2113     }
2114 ,pdfwindowui .default:n = true
2115 }

```

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

```

2116 \keys_define:nn { hyp }
2117     {
2118         ,pdfview .code:n =
2119             {
2120                 \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
2121                 \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } } {
2122                     { xyz }
2123                     {
2124                         \int_compare:nNnTF { \seq_count:N \l__hyp_tmpa_seq } > { 1 }
2125                         {
2126                             \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2127                             \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2128                             {
2129                                 \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2130                             }
2131                             {
2132                                 \tl_set:Ne \l__hyp_dest_pdfview_tl
2133

```

```

2134         {
2135             \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2136         }
2137     }
2138 }
2139 {
2140     \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2141 }
2142 }
2143 { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2144 { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2145 { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2146 { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2147 { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2148 { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2149 { fitr }
2150 {
2151     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2152     {
2153         \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2154     }
2155     {
2156         %ensure 4 values ...
2157         \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2158         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2159         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2160         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2161         \hbox_set_to_wd:Nnn \l__hyp_dest_box
2162         {
2163             \fp_eval:n
2164             {
2165                 round
2166                 (
2167                     abs
2168                     (
2169                         \seq_item:Nn\l__hyp_tmpa_seq{4}
2170                         -
2171                         (\seq_item:Nn\l__hyp_tmpa_seq{2})
2172                     ),
2173                     3
2174                 )
2175             }bp
2176         }{}
2177         \box_set_dp:Nn \l__hyp_dest_box
2178         {
2179             \fp_eval:n
2180             {
2181                 round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2182             }bp
2183         }
2184         \box_set_ht:Nn \l__hyp_dest_box
2185         {
2186             \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2187         }

```



```

2188         }
2189     }
2190 }
2191 {
2192     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2193     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2194 }
2195 }
2196 ,pdfview .initial:n = {xyz}
2197 }

```

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.

```

2198 \keys_define:nn { hyp }
2199 {
2200     ,pdflang .code:n =
2201     {
2202         \tl_if_empty:nF { #1 }
2203         {
2204             \pdfmanagement_add:nne {Catalog} { Lang } { (#1) }
2205             \AddToDocumentProperties[document]{lang}{#1}
2206         }
2207     }
2208 }

```

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. The values are expanded with `\text_expand:n`
`pdfproducer` (*setup key*)
`pdfkeywords` (*setup key*)

```

2209 \regex_new:N\l__hyp_optlang_regex
2210 \regex_set:Nn\l__hyp_optlang_regex {\A\[[A-Za-z\-\+]\](.*)}
2211 \cs_generate_variant:Nn\clist_item:nn{on}
2212 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2213 {
2214     \keys_define:nn { hyp }
2215     {
2216         pdf#1 .code:n =
2217         {
2218             \tl_set:Ne\l__hyp_tmpa_tl {\text_expand:n{##1}}
2219             \__hyp_store_metadata:no {pdf#1}{\l__hyp_tmpa_tl}
2220             \tl_if_empty:NTF \l__hyp_tmpa_tl
2221             {
2222                 \str_case:nnF { #1 }
2223                 {
2224                     {creator}
2225                     {
2226                         \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }

```

```

2227         \pdfmanagement_add:nne {Info}{Creator}{()}
2228     }
2229     {producer}
2230     {
2231         \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2232         \pdfmanagement_add:nne {Info}{Producer}{()}
2233     }
2234 }
2235 {
2236     \pdfmanagement_remove:nn {Info}{#2}
2237 }
2238 }
2239 {
2240     \tl_set:Nel__hyp_tmpb_tl {\clist_item:on{\__hyp_tmpa_tl}{1}}
2241     \exp_args:NNV
2242     \regex_extract_once:NnN \l__hyp_optlang_regex \l__hyp_tmpb_tl\l__hyp_tmpa_str
2243     \seq_if_empty:NTF\l__hyp_tmpa_seq
2244     {
2245         \__hyp_text_pdfstring_info:oN {\__hyp_tmpa_tl}\l__hyp_tmpa_str
2246     }
2247     {
2248         \__hyp_text_pdfstring_info:eN {\seq_item:Nn \l__hyp_tmpa_seq{3}}\l__hyp_tmpa_str
2249     }
2250     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2251     {
2252         \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2253     }
2254 }
2255 }
2256 }
2257 \keys_define:nn { hyp / info }
2258 {
2259     #2 .code:n =
2260     {
2261         \tl_set:Nel__hyp_tmpa_tl {\text_expand:n{##1}}
2262         \__hyp_store_metadata:eo {pdf\str_lowercase:n{##1}}{\l__hyp_tmpa_tl}
2263         \tl_if_blank:nTF {##1}
2264         {
2265             \pdfmanagement_remove:nn {Info}{#2}
2266         }
2267         {
2268             \__hyp_text_pdfstring_info:oN {\l__hyp_tmpa_tl}\l__hyp_tmpa_str
2269             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2270             {
2271                 \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2272             }
2273         }
2274     }
2275     ,unknown .code:n =
2276     {
2277         \__hyp_text_pdfstring_info:eN {##1}\l__hyp_tmpa_str
2278         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2279         {
2280             \exp_args:Nno

```

```

2281         \pdfmanagement_add:nne {Info}
2282         { \l_keys_key_str } {\l__hyp_tmpa_str}
2283     }
2284 }
2285 }
2286 }
2287 \__hyp_setup_info_key:nn {author} {Author}
2288 \__hyp_setup_info_key:nn {title} {Title}
2289 \__hyp_setup_info_key:nn {producer} {Producer}
2290 \__hyp_setup_info_key:nn {creator} {Creator}
2291 % ignored key: addtopdfcreator
2292 \__hyp_setup_info_key:nn {subject} {Subject}
2293 \__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) 2294 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2295 {
2296     \keys_define:nn { hyp }
2297     {
2298         pdf#1 .code:n =
2299         {
2300             \tl_if_blank:nTF {##1}
2301             {
2302                 \pdfmanagement_remove:nn {Info}{#2}
2303             }
2304             {
2305                 \pdfmanagement_add:nne {Info}{#2}{(##1)}
2306             }
2307             \__hyp_store_metadata:nn {pdf#1}{##1}
2308             \AddToDocumentProperties[document]{#1}{##1}
2309         }
2310     }
2311     \keys_define:nn { hyp / info }
2312     {
2313         #2 .code:n =
2314         {
2315             \tl_if_blank:nTF {##1}
2316             {
2317                 \pdfmanagement_remove:nn {Info}{#2}
2318             }
2319             {
2320                 \pdfmanagement_add:nne {Info}{#2}{(##1)}
2321             }
2322             \exp_args:Ne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2323         }
2324     }
2325 }
2326
2327 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2328 \__hyp_setup_info_date_key:nn {moddate} {ModDate}
2329 \keys_define:nn { hyp }
2330 {
2331     pdfmetadate .code:n = { \__hyp_store_metadata:nn {pdfmetadate}{#1} }

```

```
2332     }
```

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

```
2333 \keys_define:nn { hyp }
2334 {
2335   ,pdftrapped .code:n =
2336   {
2337     \exp_args:Nne
2338     \keys_set:nn { hyp } { _pdftrapped = \str_uppercase:n { #1 } }
2339   }
2340   ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2341   {
2342     \pdfmanagement_add:nne {Info}{Trapped}
2343     {/
2344       \str_uppercase:f { \str_head:n { #1 } }
2345       \str_lowercase:f { \str_tail:n { #1 } }
2346     }
2347     \__hyp_store_metadata:ne {pdftrapped}
2348     {
2349       \str_uppercase:f { \str_head:n { #1 } }
2350       \str_lowercase:f { \str_tail:n { #1 } }
2351     }
2352   }
2353   ,_pdftrapped / unknown .code:n =
2354   {
2355     \msg_warning:nneee { hyp } { unknown-choice }
2356     { pdftrapped }
2357     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2358     { \exp_not:n {#1} }
2359   }
2360 }
```

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

```
2361 \keys_define:nn { hyp }
2362 {
2363   pdfinfo .code:n =
2364   {
2365     \keys_set:nn { hyp / info } { #1 }
2366   }
2367 }
```

Now we set some default values

```
2368 \keys_set:nn { hyp} {pdfcreator = LaTeX-with-hyperref}
2369 \keys_set:nn { hyp} {pdfauthor = }
2370 \keys_set:nn { hyp} {pdftitle = }
2371 \keys_set:nn { hyp} {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:

```
2372 \clist_map_inline:nn
2373 {
```

```

2374 ,pdfcopyright
2375 ,pdftype
2376 ,pdflicenseurl
2377 ,pdfauthortitle
2378 ,pdfcaptionwriter
2379 ,pdfmetalang
2380 ,pdfsource
2381 ,pdfdocumentid
2382 ,pdfinstanceid
2383 ,pdfversionid
2384 ,pdfrendition
2385 ,pdfpublication
2386 ,pdfpubtype
2387 ,pdfbytes
2388 ,pdfnumpages
2389 ,pdfissn
2390 ,pdfeissn
2391 ,pdfisbn
2392 ,pdfbookedition
2393 ,pdfpublisher
2394 ,pdfvolumenum
2395 ,pdfissuenum
2396 ,pdfpagerange
2397 ,pdfdoi
2398 ,pdfurl
2399 ,pdfidentifier
2400 ,pdfsubtitle
2401 ,pdfpubstatus
2402 ,pdfcontactaddress
2403 ,pdfcontactcity
2404 ,pdfcontactregion
2405 ,pdfcontactpostcode
2406 ,pdfcontactcountry
2407 ,pdfcontactphone
2408 ,pdfcontactemail
2409 ,pdfcontacturl
2410 ,pdfdate
2411 }
2412 {
2413   \keys_define:nn { hyp }
2414   {
2415     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2416   }
2417 }
2418

```

12.5 Transitions

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

```

2419 \keys_define:nn { hyp }
2420 {
2421   pdfpageduration .code:n =
2422   {
2423     \tl_if_blank:nTF { #1 }

```

```

2424     {
2425     \pdfmanagement_remove:nn {Page}{Dur}
2426     }
2427     {
2428     \pdfmanagement_add:nnn {Page}{Dur}{#1}
2429     }
2430 }
2431 }

```

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

```

2432 \keys_define:nn { hyp }
2433 {
2434   pdfpagetransition .code:n =
2435   {
2436     \tl_if_blank:nTF {#1}
2437     {
2438       \pdfmanagement_remove:nn {Page}{Trans}
2439     }
2440     {
2441       \group_begin:
2442       \keys_set:nn { hyp / trans }{style=R,#1}
2443       \pdf_object_unnamed_write:ne { dict }
2444       {
2445         \pdfdict_use:n {l__hyp_page/Trans}
2446       }
2447       \pdfmanagement_add:nne {Page}{Trans}{\pdf_object_ref_last:}

```

```

2448         \group_end:
2449     }
2450 }
2451 }
2452 \keys_define:nn { hyp / trans }
2453 {
2454     ,style .choices:nn =
2455     { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2456     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2457     ,style / unknown .code:n =
2458     {
2459         \msg_warning:nnee { hyp } { unknown-choice }
2460         { trans / style }
2461         { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2462         { \exp_not:n {#1} }
2463     }
2464     ,duration .code:n =
2465     {
2466         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{#1}
2467     }
2468     ,direction .choices:nn =
2469     { H,V }
2470     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2471     ,direction .choices:nn =
2472     { 0,90,180,270,315 }
2473     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2474     ,direction / None .code:n =
2475     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2476     ,direction / unknown .code:n =
2477     {
2478         \msg_warning:nnee { hyp } { unknown-choice }
2479         { trans / direction }
2480         {
2481             H~(horizontal,~only~Split,~Blinds),
2482             V~(vertical,~only~Split,~Blinds),
2483             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2484             90~(bottom~to~top,~only~Wipe),
2485             180~(right~to~left,~only~Wipe),
2486             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2487             315~(top~left~to~bottom,~only~Glitter),
2488             None~(only~Fly)
2489         }
2490         { \exp_not:n {#1} }
2491     }
2492     ,motion .choices:nn =
2493     { I,0 }
2494     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2495     ,motion / unknown .code:n =
2496     {
2497         \msg_warning:nnee { hyp } { unknown-choice }
2498         { trans / motion }
2499         { I~(inwards) , 0~(outwards) }
2500         { \exp_not:n {#1} }
2501     }

```

```

2502 ,scale .code:n =
2503   { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2504 ,opaque .choices:nn = {true,false}
2505   { \pdfdict_put:nnn { l__hyp_page/Trans }{ B } { #1} }
2506 ,opaque / unknown .code:n =
2507   {
2508     \msg_warning:nneee { hyp } { unknown-choice }
2509     { trans / B }
2510     { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2511     { \exp_not:n {#1} }
2512   }
2513 % try to set unknown keys as style
2514 ,unknown .code:n =
2515   {
2516     % warning ...
2517     \exp_args:Nne\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2518   }
2519 }

```

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

```

2520 \keys_set_known:nv{ hyp }{opt@hyperref.sty}
  Unfinished Form field code
2521 \NewDocumentCommand \MakeFieldObject { m m }
2522   {
2523     \pdfxform_new:nnn { #2 }{ } { #1 }
2524   }
2525
2526
2527 \prop_new:N \g__hyp_AcroForm_CoFields_prop
2528 \prop_new:N \g__hyp_AcroForm_Fields_prop
2529
2530 \let\HyField@afields\@empty
2531 \let\HyField@cofields\@empty
2532 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2533
2534 % the value doesn't matter, but with a prop we avoid duplicates and it is
2535 % clearly faster than removing them from a sequence
2536 \def\HyField@AuxAddToFields#1
2537   {
2538     \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2539   }%
2540
2541 %fields with empty key get a value too -- lets hope that
2542 %this give the expected behaviour
2543 \def\HyField@AuxAddToCoFields #1 #2
2544   {
2545     \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2546   }
2547
2548 \Hy@AtBeginDocument
2549   {
2550     \if@filesw
2551       \immediate\write\@mainaux{%
2552         \string\providecommand\string\HyField@AuxAddToFields[1]{}%

```



```

2553     }%
2554     \immediate\write\@mainaux{%
2555         \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2556     }%
2557 \fi
2558 \let\HyField@AfterAuxOpen\@firstofone
2559 }%
2560
2561 \def\HyField@AddToFields
2562 {
2563     \exp_args:Ne\HyField__hypAddToFields
2564     {
2565         \pdfannot_box_ref_last:
2566     }
2567 \ifx\Fld@calculate@code\@empty
2568 \else
2569     \begingroup
2570     \Hy@safe@activestrue
2571     \edef\Hy@temp{%
2572         \endgroup
2573         \if@filesw
2574             \write\@mainaux
2575             {
2576                 \string\HyField@AuxAddToCoFields
2577                 {
2578                     \Fld@calculate@sortkey
2579                 }
2580                 {
2581                     \pdfannot_box_ref_last:
2582                 }
2583             }
2584         \fi
2585     }%
2586 \Hy@temp
2587 \fi
2588 }%
2589
2590 \def\HyField__hypAddToFields#1{
2591     \HyField@AfterAuxOpen{%
2592         \if@filesw
2593             \write\@mainaux{%
2594                 \string\HyField@AuxAddToFields{#1}%
2595             }%
2596         \fi
2597     }%
2598 }%
2599
2600 \ExplSyntaxOff
2601 \ExplSyntaxOn
2602
2603 \def\@Form[#1]
2604 {
2605     \kvsetkeys{Form}{#1}
2606     \pdf@ifdraftmode{}

```

```

2607 {
2608   \Hy@FormObjects
2609   \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2610   {
2611     \pdfmanagement_add:nne { Catalog / AcroForm } { Fields }{##1}
2612     %\pdfmanagement_show:n { Catalog / AcroForm }
2613   }
2614   \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2615   {
2616     \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2617     {
2618       \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2619     }
2620     \seq_sort:Nn \l__hyp_tmpa_seq
2621     {
2622       \str_compare:nNnTF {##1} > {##2}
2623       { \sort_return_swapped: }
2624       { \sort_return_same: }
2625     }
2626     \seq_map_inline:Nn \l__hyp_tmpa_seq
2627     {
2628       \pdfmanagement_add:nne { Catalog / AcroForm }
2629       { CO }
2630       {
2631         \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2632       }
2633     }
2634   }
2635   \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2636   {ZaDb} {\pdf_object_ref:n {__hyp/Font/ZaDb} }
2637   \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2638   {Helv} {\pdf_object_ref:n {__hyp/Font/Helv} }
2639   \pdfmanagement_add:nne {Catalog /AcroForm}
2640   {DA}{(/Helv~10~Tf~0~g)}
2641   \pdfmeta_standard_verify:nTF {form_no_NeedAppearances}
2642   {
2643     \legacy_if:nT { HyField@NeedAppearances }
2644     {
2645       \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2646     }
2647   }
2648   {
2649     \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2650   }
2651 }
2652 }
2653 \ExplSyntaxOff
2654 \let\@endForm\@empty
2655 \let\HyAnn@AbsPageLabel\@empty
2656 \let\Fld@pageobjref\@empty
2657
2658 \ExplSyntaxOn
2659 \newcount\HyAnn@Count
2660 \HyAnn@Count=\z@

```

```

2661 \def\HyAnn@AbsPageLabel
2662 {
2663   \global\advance\HyAnn@Count by\@ne
2664   %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2665   %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2666   %\zref@refused{HyAnn@\the\HyAnn@Count}%
2667   \__hyp_property_record:ee {HyAnn@\the\HyAnn@Count}{abspage}
2668   \property_ref_undefined_warn:ee {HyAnn@\the\HyAnn@Count}{abspage}
2669 }%
2670 \prg_generate_conditional_variant:Nnn \property_if_recorded:mn {ee} {T}
2671 \def\Fld@pageobjref
2672 {
2673   \property_if_recorded:eeT {HyAnn@\the\HyAnn@Count}{abspage}
2674   {
2675     /P~\pdf_pageobject_ref:e
2676     {
2677       \property_ref:ee{HyAnn@\the\HyAnn@Count}{abspage}
2678     }
2679   }
2680 }
2681 \ExplSyntaxOff
2682 \ExplSyntaxOn
2683 %% check if the attr should be set through
2684 %% hooks.
2685 %% check if options are missing.
2686 \def\@TextField[#1]#2{% parameters, label
2687   \def\Fld@name{#2}%
2688   \let\Fld@default\@empty
2689   \let\Fld@value\@empty
2690   \def\Fld@width{\DefaultWidthofText}%
2691   \def\Fld@height{%
2692     \ifFld@multiline
2693       \DefaultHeightofTextMultiline
2694     \else
2695       \DefaultHeightofText
2696     \fi
2697 }%
2698 \begingroup
2699   \expandafter\HyField@SetKeys\expandafter{%
2700     \DefaultOptionsofText,#1%
2701   }%
2702   \PDFForm@Name
2703   \HyField@FlagsText
2704   \ifFld@hidden\def\Fld@width{1sp}\fi
2705   \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2706   \LayoutTextField{#2}{%
2707     \leavevmode
2708     \HyAnn@AbsPageLabel
2709     \Hy@escapeform\PDFForm@Text
2710     \pdfannot_box:nmmn
2711       {\Fld@width}
2712       {\Fld@height}
2713       {Opt} %is this correct?
2714       {\PDFForm@Text}

```

```

2715     \MakeTextField{\Fld@width}{\Fld@height}
2716     \HyField@AddToFields
2717   }%
2718 \endgroup
2719 }
2720 \providecommand\@curropt{}
2721 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2722   \def\Fld@name{#2}
2723   \let\Fld@default\relax
2724   \let\Fld@value\relax
2725   \def\Fld@width{\DefaultWidthofChoiceMenu}
2726   \def\Fld@height{\DefaultHeightofChoiceMenu}
2727   \begingroup
2728     \Fld@menulength=0 %
2729     \@tempdima\z@
2730     \clist_map_variable:nNn { #3 } \@curropt
2731     %\@for\@curropt:=#3\do
2732     {%
2733       \expandafter\Fld@checkequals\@curropt==\%
2734       \Hy@StepCount\Fld@menulength
2735       \settowidth{\@tempdimb}{\@currDisplay}%
2736       \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2737     }%
2738     \advance\@tempdima by~15\p@
2739     \begingroup
2740       \HyField@SetKeys{#1}
2741     \edef\x{\endgroup
2742       \noexpand\expandafter
2743       \noexpand\HyField@SetKeys
2744       \noexpand\expandafter{%
2745         \expandafter\noexpand\csgname DefaultOptionsof%
2746         \ifFld@radio
2747           Radio%
2748         \else
2749           \ifFld@combo
2750             \ifFld@popdown
2751               PopdownBox%
2752             \else
2753               ComboBox%
2754             \fi
2755           \else
2756             ListBox%
2757           \fi
2758         \fi
2759       \endcsname
2760     }%
2761   }\x
2762   \HyField@SetKeys{#1}%
2763   \PDFForm@Name
2764   \ifFld@hidden\def\Fld@width{1sp}\fi
2765   \ifx\Fld@value\relax
2766     \let\Fld@value\Fld@default
2767   \fi
2768   \LayoutChoiceField{#2}{%

```

```

2769     \ifFld@radio
2770         \HyField@FlagsRadioButton
2771         \__hypRadio{#3}%
2772     \else
2773         \begingroup
2774             \HyField@FlagsChoice
2775             \ifdim\Fld@width<\@tempdima
2776                 \ifdim\@tempdima<1cm\@tempdima1cm\fi
2777                 \edef\Fld@width{\the\@tempdima}%
2778             \fi
2779             \ifFld@combo
2780             \else
2781                 \@tempdima=\the\Fld@menulength\Fld@charsize
2782                 \advance\@tempdima by-\Fld@borderwidth bp %
2783                 \advance\@tempdima by-\Fld@borderwidth bp %
2784                 \edef\Fld@height{\the\@tempdima}%
2785             \fi
2786             \__hypListBox{#3}%
2787         \endgroup
2788     \fi
2789 }%
2790 \endgroup
2791 }
2792
2793 \def\__hypRadio#1{%
2794     \Fld@listcount=0-%
2795     %\show\Fld@default
2796     \EdefEscapeName\Fld@default{\Fld@default}%
2797     \clist_map_variable:nNn { #1 } \@curropt
2798     %\@for\@curropt:=#1\do
2799     {%
2800         \expandafter\Fld@checkequals\@curropt==\%
2801         \EdefEscapeName\@currValue{\@currValue}%
2802         \Hy@StepCount\Fld@listcount
2803         \@currDisplay\space
2804         \leavevmode
2805         \HyAnn@AbsPageLabel
2806         \Hy@escapeform\PDFForm@Radio
2807         \pdfxform_if_exist:nF { __hyp_xform_Ding }
2808         {
2809             \pdfxform_new:nnn { __hyp_xform_Ding } {}
2810             {
2811                 \group_begin:
2812                 \fontfamily{pzd}
2813                 \fontencoding{U}
2814                 \fontseries{m}
2815                 \fontshape{n}
2816                 \selectfont
2817                 \char123
2818                 \group_end:
2819             }
2820         }
2821         \pdfannot_box:nnne
2822         {\Fld@width}

```

```

2823     {\Fld@height}
2824     {Opt} %is this correct?
2825     {
2826         \PDFForm@Radio
2827         /AP
2828         <<
2829         /N
2830         <<
2831         /@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2832         %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2833         >>
2834         >>
2835     }
2836     {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2837     \int_compare:nNnT { \Fld@listcount} = { 1 }
2838     { \HyField@AddToFields }
2839     \c_space_tl % deliberate space between radio buttons
2840             % to do: --> should be configurable
2841 }%
2842 }
2843
2844 \newcount\Fld@listcount
2845 \def\__hypListbox#1
2846 {
2847     \HyField@PDFChoices{#1}
2848     \mode_leave_vertical:
2849     \HyAnn@AbsPageLabel
2850     \Hy@escapeform\PDFForm@List
2851     \pdf_link_user:nnn
2852         {widget} %perhaps we need more types??
2853         {\PDFForm@List}
2854         {\MakeChoiceField{\Fld@width}{\Fld@height}}
2855     \HyField@AddToFields
2856 }
2857
2858
2859 \def\@PushButton[#1]#2{% parameters, label
2860     \def\Fld@name{#2}%
2861     \group_begin:
2862         \exp_args:No\HyField@SetKeys
2863         {
2864             \DefaultOptionsofPushButton,#1
2865         }
2866         \PDFForm@Name
2867         \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2868         {
2869             \HyField@FlagsPushButton
2870             \legacy_if:nT {Fld@hidden}
2871             {
2872                 \def\Fld@width{1sp}
2873             }
2874             \LayoutPushButtonField
2875             {
2876                 \mode_leave_vertical:

```

```

2877         \HyAnn@AbsPageLabel
2878         \Hy@escapeform\PDFForm@Push
2879         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2880         \pdfannot_box:nnnn
2881         { \box_wd:N\l_tmpa_box}
2882         { \box_ht:N\l_tmpa_box}
2883         { \box_dp:N\l_tmpa_box} %is this correct?
2884         { \PDFForm@Push}
2885         { \box_use:N\l_tmpa_box}
2886         \HyField@AddToFields
2887     }
2888 }
2889 {
2890     \msg_error:nn { hyp }{ pdfa-no-push-button }
2891     \LayoutPushButtonField
2892     {
2893         \mode_leave_vertical:
2894         \MakeButtonField{#2}
2895     }
2896 }
2897 \group_end:
2898 }
2899
2900 \def\@Submit[#1]#2
2901 {
2902     \def\Fld@width {\DefaultWidthofSubmit}
2903     \def\Fld@height{\DefaultHeightofSubmit}
2904     \group_begin:
2905         \exp_args:No\HyField@SetKeys
2906         {
2907             \DefaultOptionsofSubmit,#1
2908         }
2909     \HyField@FlagsPushButton
2910     \HyField@FlagsSubmit
2911     \legacy_if:nT { Fld@hidden }
2912     {
2913         \def\Fld@width{1sp}
2914     }
2915     \mode_leave_vertical:
2916     \HyAnn@AbsPageLabel
2917     \Hy@escapeform\PDFForm@Submit
2918     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2919     \pdfxform_if_exist:nF
2920     { __hyp_xform_Submit }
2921     {
2922         \pdfxform_new:nnn { __hyp_xform_Submit }{}
2923         {
2924             \fbox{\color_select:n{yellow}\textsf{Submit}}
2925         }
2926         \pdfxform_new:nnn { __hyp_xform_SubmitP }{}
2927         {
2928             \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2929         }
2930     }

```

```

2931 \pdfannot_box:nnnn
2932   {\box_wd:N\l_tmpa_box}
2933   {\box_ht:N\l_tmpa_box}
2934   {\box_dp:N\l_tmpa_box} %is this correct?
2935   {
2936     \PDFForm@Submit
2937     /AP<<
2938       /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2939       /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2940     >>
2941   }
2942   \HyField@AddToFields
2943   \box_use:N\l_tmpa_box
2944
2945 \group_end:
2946 }
2947
2948 \def\@Reset[#1]#2
2949 {
2950   \def\Fld@width {\DefaultWidthofReset}
2951   \def\Fld@height{\DefaultHeightofReset}
2952   \group_begin:
2953     \exp_args:No\HyField@SetKeys
2954     {
2955       \DefaultOptionsofReset,#1
2956     }
2957     \mode_leave_vertical:
2958     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2959     {
2960       \HyField@FlagsPushButton
2961       \legacy_if:nT { Fld@hidden }
2962         { \def\Fld@width{1sp} }
2963       \HyAnn@AbsPageLabel
2964       \Hy@escapeform\PDFForm@Reset
2965       \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2966       \pdfannot_box:nnnn
2967         {\box_wd:N\l_tmpa_box}
2968         {\box_ht:N\l_tmpa_box}
2969         {\box_dp:N\l_tmpa_box} %is this correct?
2970         { \PDFForm@Reset }
2971       \HyField@AddToFields
2972       \box_use:N \l_tmpa_box
2973     }
2974     {
2975       \msg_error:nn { hyp }{ pdfa-no-reset-button }
2976       \MakeButtonField{#2}
2977     }
2978   \group_end:
2979 }
2980
2981 \def\@CheckBox[#1]#2
2982   {% parameters, label
2983   \def\Fld@name{#2}
2984   \def\Fld@default{0}

```



```

2985 \group_begin:
2986 \def\Fld@width {\DefaultWidthofCheckBox}
2987 \def\Fld@height{\DefaultHeightofCheckBox}
2988 \exp_args:No\HyField@SetKeys
2989 {
2990 \DefaultOptionsofCheckBox,#1
2991 }
2992 \PDFForm@Name
2993 \HyField@FlagsCheckBox
2994 \legacy_if:nT { Fld@hidden }
2995 {
2996 \def\Fld@width{1sp}
2997 }
2998 \LayoutCheckField{#2}
2999 {
3000 \mode_leave_vertical:
3001 \HyAnn@AbsPageLabel
3002 \Hy@escapeform\PDFForm@Check
3003 \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
3004 {
3005 \pdfxform_new:nnn
3006 {__hyp_xform_CheckMarkYes}{}
3007 {
3008 \group_begin:
3009 \fontfamily{pzd}
3010 \fontencoding{U}
3011 \fontseries{m}
3012 \fontshape{n}
3013 \selectfont
3014 \char51
3015 \group_end:
3016 }
3017 \pdfxform_new:nnn
3018 {__hyp_xform_CheckMarkOff}{}
3019 {
3020 \group_begin:
3021 \fontfamily{pzd}
3022 \fontencoding{U}
3023 \fontseries{m}
3024 \fontshape{n}
3025 \selectfont
3026 \phantom{\char51} %perhaps xetex needs some small glyph ..
3027 \group_end:
3028 }
3029 }
3030 \pdfannot_box:nmm
3031 {\Fld@width}
3032 {\Fld@height}
3033 {Opt} %is this correct?
3034 {\PDFForm@Check}
3035 \HyField@AddToFields %check if this works with xelatex ...
3036 }
3037 \group_end:
3038 }

```

```

3039 \ExplSyntaxOff
3040
3041 %hm. Should a luatex driver use type1 fonts in fields????
3042 \ExplSyntaxOn
3043 \def\Hy@FormObjects
3044 {
3045   \pdf_object_new:n   {__hyp/Encoding/pdfdoc }
3046   \pdf_object_new:n   {__hyp/Font/ZaDb }
3047   \pdf_object_new:n   {__hyp/Font/Helv }
3048   \pdf_object_write:mne {__hyp/Encoding/pdfdoc } { dict }
3049   {
3050     /Type/Encoding
3051     /Differences[
3052       24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
3053       /ring/tilde
3054       \c_space_tl
3055       39/quotesingle
3056       \c_space_tl
3057       96/grave %
3058       \iow_newline:
3059       128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
3060       /fraction/guilsinglleft/guilsinglright/minus/perthousand
3061       /quotedblbase/quotedblleft/quotedblright/quoteleft
3062       /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
3063       /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
3064       \iow_newline:
3065       164/currency
3066       \c_space_tl
3067       166/brokenbar
3068       \c_space_tl
3069       168/dieresis/copyright/ordfeminine
3070       \c_space_tl
3071       172/logicalnot/.notdef/registered/macron/degree/plusminus
3072       /twosuperior/threesuperior/acute/mu
3073       \c_space_tl
3074       183/periodcentered/cedilla/onesuperior/ordmasculine
3075       \c_space_tl
3076       188/onequarter/onehalf/threequarters
3077       \iow_newline:
3078       192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
3079       /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
3080       /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
3081       /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
3082       /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
3083       /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
3084       /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
3085       /iacute/icircumflex/idieresis/eth/ntilde/ograve/oacute
3086       /ocircumflex/otilde/odieresis/divide/oslash/ugrave
3087       /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
3088     ]
3089   }
3090   \pdf_object_write:nnn {__hyp/Font/ZaDb } { dict }
3091   {
3092     /Type/Font

```

```

3093         /Subtype/Type1
3094         /Name/ZaDb
3095         /BaseFont/ZapfDingbats
3096     }
3097     \pdf_object_write:nne {__hyp/Font/Helv } { dict }
3098     {
3099         /Type/Font
3100         /Subtype/Type1
3101         /Name/Helv
3102         /BaseFont/Helvetica
3103         /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
3104     }
3105     \global\let\Hy@FormObjects\relax
3106 }
3107 \ExplSyntaxOff
3108 \providecommand*{\Fld@pageobjref}{}
3109 \ifcsname pdf@escapestring\endcsname
3110     \def\Hy@escapeform#1{%
3111         \ifHy@pdfescapeform
3112             \let\Hy@escapestring\pdfescapestring
3113         \else
3114             \let\Hy@escapestring\@firstofone
3115         \fi
3116     }%
3117     \Hy@escapeform{}%
3118 \else
3119     \let\Hy@escapestring\@firstofone
3120     \def\Hy@escapeform#1{%
3121         \ifHy@pdfescapeform
3122             \def\Hy@escapestring##1{%
3123                 \noexpand\Hy@escapestring{\noexpand##1}%
3124             }%
3125             \edef\Hy@temp{#1}%
3126             \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring}\@nil
3127         \def\Hy@escapestring##1{%
3128             \@ifundefined{Hy@esc@\string##1}{%
3129                 ##1%
3130                 \ThisShouldNotHappen
3131             }{%
3132                 \csname Hy@esc@\string##1\endcsname
3133             }%
3134         }%
3135     \else
3136         \let\Hy@escapestring\@firstofone
3137     \fi
3138 }%
3139 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3140     \ifx\#3\%
3141     \else
3142         \expandafter
3143         \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3144         \Hy@ReturnAfterFi{%
3145             \Hy__hypescapeform#3\@nil
3146         }%

```

```

3147     \fi
3148   }%
3149   \fi
3150   \def\PDFForm@Name{%
3151     \PDFForm__hypName\Fld@name
3152     \ifx\Fld@altname\relax
3153     \else
3154       \PDFForm__hypName\Fld@altname
3155     \fi
3156     \ifx\Fld@mappingname\relax
3157     \else
3158       \PDFForm__hypName\Fld@mappingname
3159     \fi
3160   }
3161   \def\PDFForm__hypName#1{%
3162     \begingroup
3163       \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3164       \ifHy@unicode
3165         \Hy@unicodedefalse
3166       \fi
3167       \fi
3168       \pdfstringdef\Hy@gtemp#1%
3169     \endgroup
3170     \let#1\Hy@gtemp
3171   }
3172   \def\Fld@X@additionalactions{%
3173     \ifx\Fld@keystroke@code\@empty
3174     \else
3175       /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3176     \fi
3177     \ifx\Fld@format@code\@empty
3178     \else
3179       /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3180     \fi
3181     \ifx\Fld@validate@code\@empty
3182     \else
3183       /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3184     \fi
3185     \ifx\Fld@calculate@code\@empty
3186     \else
3187       /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3188     \fi
3189     \ifx\Fld@onfocus@code\@empty
3190     \else
3191       /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3192     \fi
3193     \ifx\Fld@onblur@code\@empty
3194     \else
3195       /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3196     \fi
3197     \ifx\Fld@onmousedown@code\@empty
3198     \else
3199       /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3200     \fi

```

```

3201 \ifx\Fld@onmouseup@code\@empty
3202 \else
3203 /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3204 \fi
3205 \ifx\Fld@onenter@code\@empty
3206 \else
3207 /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3208 \fi
3209 \ifx\Fld@onexit@code\@empty
3210 \else
3211 /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3212 \fi
3213 }
3214 \ExplSyntaxOn
3215 \def\Fld@additionalactions
3216 {%
3217 \exp_args:Ne\str_if_eq:nF {\Fld@X@additionalactions}{}
3218 {
3219 \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3220 {/AA<<\Fld@X@additionalactions>>}
3221 }
3222 }
3223 \ExplSyntaxOff
3224 \def\Fld@annotnames{%
3225 /T(\Fld@name)%
3226 \ifx\Fld@altname\relax
3227 \else
3228 /TU(\Fld@altname)%
3229 \fi
3230 \ifx\Fld@mappingname\relax
3231 \else
3232 /TM(\Fld@mappingname)%
3233 \fi
3234 }
3235 \ExplSyntaxOn
3236 \def\PDFForm@Check
3237 {
3238 /Subtype/Widget
3239 ~\Fld@annotflags
3240 ~\Fld@pageobjref
3241 ~\Fld@annotnames
3242 /FT/Btn
3243 \Fld@flags
3244 /Q~\Fld@align
3245 /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3246 /AP
3247 <<
3248 /N
3249 <<
3250 /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3251 /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3252 >>
3253 >>
3254 /MK<<

```

```

3255 \int_compare:nNnF {\Fld@rotation}={0}
3256 {
3257   /R~\Fld@rotation
3258 }
3259 \tl_if_empty:NF\Fld@bordercolor
3260 {
3261   /BC[\Fld@bordercolor]
3262 }
3263 \tl_if_empty:NF\Fld@bcolor
3264 {
3265   /BG[\Fld@bcolor]
3266 }
3267 /CA(\Hy@escapestring{\Fld@cbsymbol})%
3268 >>
3269 /DA
3270 (
3271   /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3272   \tl_if_empty:NF \Fld@color
3273   {
3274     \c_space_tl \Fld@color
3275   }
3276 )
3277 /H/P
3278 \legacy_if:nTF {Fld@checked}
3279 {
3280   /V/Yes /AS/Yes
3281 }
3282 {
3283   /V/Off /AS/Off
3284 }
3285 \Fld@additionalactions
3286 }
3287 \ExplSyntaxOff
3288 \ExplSyntaxOn
3289 \def\PDFForm@Push
3290 {
3291   /Subtype/Widget
3292   ~\Fld@annotflags
3293   ~\Fld@pageobjref
3294   ~\Fld@annotnames
3295   /FT/Btn
3296   ~\Fld@flags
3297   /H/P
3298   /BS<<W~\Fld@borderwidth/S/\Fld@borderstyle>>
3299   \bool_if:nT
3300   {
3301     !\int_compare_p:nNn {\Fld@rotation} = {0}
3302     ||
3303     \tl_if_exist_p:N \Fld@bordercolor
3304   }
3305   {
3306     /MK
3307     <<
3308     \int_compare:nNnF {\Fld@rotation} = {0}

```

```

3309         {
3310             /R~\Fld@rotation
3311         }
3312         \tl_if_exist:NT \Fld@bordercolor
3313         {
3314             /BC[\Fld@bordercolor]
3315         }
3316         >>
3317     }
3318     /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3319     \Fld@additionalactions
3320 }
3321
3322 \ExplSyntaxOff
3323 \def\PDFForm@List{%
3324     /Subtype/Widget%
3325     \Fld@annotflags
3326     \Fld@pageobjref
3327     \Fld@annotnames
3328     /FT/Ch%
3329     \Fld@flags
3330     /Q \Fld@align
3331     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3332     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3333     \ifx\Fld@bordercolor\relax\else 1\fi
3334     \ifx\Fld@bcolor\relax \else 1\fi
3335     \space
3336 \else
3337     /MK<<%
3338     \ifnum\Fld@rotation=\z@
3339     \else
3340         /R \Fld@rotation
3341     \fi
3342     \ifx\Fld@bordercolor\relax
3343     \else
3344         /BC[\Fld@bordercolor]%
3345     \fi
3346     \ifx\Fld@bcolor\relax
3347     \else
3348         /BG[\Fld@bcolor]%
3349     \fi
3350     >>%
3351 \fi
3352 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3353     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3354 \Fld@choices
3355 \Fld@additionalactions
3356 }
3357 \ExplSyntaxOn
3358 \def\PDFForm@Radio
3359 {
3360     /Subtype/Widget
3361     ~\Fld@annotflags
3362     ~\Fld@pageobjref

```

```

3363 ~\Fld@annotnames
3364 /FT/Btn
3365 \Fld@flags
3366 /H/P
3367 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3368 /MK<<
3369 \ifnum\Fld@rotation=\z@
3370 \else
3371 /R~\Fld@rotation
3372 \fi
3373 \ifx\Fld@bordercolor\relax
3374 \else
3375 /BC[\Fld@bordercolor]%
3376 \fi
3377 \ifx\Fld@bcolor\relax
3378 \else
3379 /BG[\Fld@bcolor]%
3380 \fi
3381 /CA(\Hy\escapestring{\Fld@radiosymbol})%
3382 >>
3383 /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3384 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3385 \ifx\Fld@default\@empty
3386 /V/Off%
3387 /DV/Off%
3388 \else
3389 /V/\Fld@default
3390 /DV/\Fld@default
3391 \fi
3392 \Fld@additionalactions
3393 }
3394 \ExplSyntaxOff
3395 \ExplSyntaxOn
3396 % Does an appearance dict make sense here?
3397 \def\PDFForm@Text
3398 {
3399 /Subtype/Widget
3400 ~\Fld@annotflags
3401 ~\Fld@pageobjref
3402 ~\Fld@annotnames
3403 /FT/Tx
3404 ~\Fld@flags
3405 /Q~\Fld@align
3406 /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3407 \bool_if:nT
3408 {
3409 !\int_compare_p:nNn {\Fld@rotation} = {0}
3410 ||
3411 \tl_if_exist_p:N \Fld@bordercolor
3412 ||
3413 \tl_if_exist_p:N \Fld@bcolor
3414 }
3415 {
3416 /MK

```



```

3417     <<
3418         \int_compare:nNnF {\Fld@rotation} = {0}
3419         {
3420             /R~\Fld@rotation
3421         }
3422     \tl_if_exist:NT \Fld@bordercolor
3423     {
3424         /BC[\Fld@bordercolor]
3425     }
3426     \tl_if_exist:NT \Fld@bcolor
3427     {
3428         /BG[\Fld@bcolor]
3429     }
3430     >>
3431 }
3432 /DA
3433 (
3434     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3435     \tl_if_empty:NF {\c_space_tl\Fld@color}
3436 )
3437 /DV(\Hy@escapestring{\Fld@default})
3438 /V(\Hy@escapestring{\Fld@value})
3439 ~\Fld@additionalactions
3440 \int_compare:nNnT { \Fld@maxlen}>{0}
3441 {
3442     /MaxLen~\Fld@maxlen
3443 }
3444 }
3445 \ExplSyntaxOff
3446
3447 \def\PDFForm@Submit{%
3448     /Subtype/Widget%
3449     \Fld@annotflags
3450     \Fld@pageobjref
3451     \Fld@annotnames
3452     /FT/Btn%
3453     \Fld@flags
3454     /H/P%
3455     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3456     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3457         \ifx\Fld@bordercolor\relax\else 1\fi
3458         \space
3459     \else
3460         /MK<<%
3461         \ifnum\Fld@rotation=\z@
3462         \else
3463             /R \Fld@rotation
3464         \fi
3465         \ifx\Fld@bordercolor\relax
3466         \else
3467             /BC[\Fld@bordercolor]%
3468         \fi
3469     >>%
3470 \fi

```

```

3471 /A<<%
3472 /S/SubmitForm%
3473 /F<<%
3474 /FS/URL%
3475 /F(\Hy@escapestring{\Form@action})%
3476 >>%
3477 \Fld@submitflags
3478 >>%
3479 \Fld@additionalactions
3480 }
3481 \ExplSyntaxOn
3482 \def\PDFForm@Reset{%
3483 /Subtype/Widget%
3484 \Fld@annotflags
3485 \Fld@pageobjref
3486 \Fld@annotnames
3487 /FT/Btn%
3488 \Fld@flags
3489 /H/P%
3490 /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3491 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3492 \ifx\Fld@bordercolor\relax\else 1\fi
3493 \space
3494 \else
3495 /MK<<%
3496 \ifnum\Fld@rotation=\z@
3497 \else
3498 /R~\Fld@rotation
3499 \fi
3500 \ifx\Fld@bordercolor\relax
3501 \else
3502 /BC[\Fld@bordercolor]%
3503 \fi
3504 >>%
3505 \fi
3506 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3507 /A<</S/ResetForm>>%
3508 \Fld@additionalactions
3509 }%
3510
3511
3512 %these patterns are used in hyperref checks.
3513 %it is unclear if they are really useful and if a backend support is
3514 %needed.
3515 \str_case:VnF \c_sys_backend_str
3516 {
3517 { pdfmode }
3518 {
3519 \def\HyPat@ObjRef
3520 {
3521 [0-9]*[1-9][0-9]*~0~R
3522 }
3523 }
3524 { dvipdfmx }

```

```

3525 {
3526   \def\HyPat@ObjRef
3527   {
3528     @[\~]+
3529   }
3530 }
3531 { xdvipdfmx }
3532 {
3533   \def\HyPat@ObjRef
3534   {
3535     @[\~]+
3536   }
3537 }
3538 }
3539 { %also set in hyperref sty, so probably not needed.
3540   \def\HyPat@ObjRef/{.+}
3541 }
3542
3543
3544 \ExplSyntaxOff
3545 % UF: removed Hy@writebookmark
3546 %   \Hy@currentbookmarklevel{0}
3547 %   \Hy@numberline
3548 %   \__hypwritetorep
3549 %   counter{bookmark@seq@number}
3550 % removed \HyPsd@SanitizeForOutFile, not needed
3551 % removed \currentpdfbookmark, defined by bookmark,
3552 % should use \newcommand there
3553 % removed \subpdfbookmark, defined by bookmark,
3554 % should use \newcommand there
3555 % removed \belowpdfbookmark, defined by bookmark,
3556 % should use \newcommand there
3557 % removed \pdfbookmark, defined by bookmark,
3558 % \BOOKMARK
3559 % \@BOOKMARK
3560 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3561 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3562 %% removed \ReadBookmarks / unneeded with bookmark.
3563 %% removed \Hy@OutlineName
3564 %% removed \check@bm@number
3565 %% removed \calc@bm@number
3566
3567 \ifHy@implicit
3568 \else
3569   \expandafter\endinput
3570 \fi
3571 \newlength\Hy@SectionHShift
3572 \def\Hy@SectionAnchorHref#1{%
3573   \ifx\protect\@typeset@protect
3574     \Hy__hypSectionAnchor{#1}%
3575   \fi
3576 }
3577 \DeclareRobustCommand*{\Hy__hypSectionAnchor}[1]{%
3578   \leavevmode

```

```

3579 \hbox to Opt{%
3580   \kern-\Hy@SectionHShift
3581   \Hy@raisedlink{%
3582     \hyper@anchorstart{#1}\hyper@anchorend
3583   }%
3584   \hss
3585 }%
3586 }
3587 \@ifundefined{hyper@nopatch@sectioning}
3588 {
3589   \let\H@old@ssect\@ssect
3590   \def\@ssect#1#2#3#4#5{%
3591     \Hy@MakeCurrentHrefAuto{section*}%
3592     \setlength{\Hy@SectionHShift}{#1}%
3593     \begingroup
3594       \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3595       \toks\tw@\expandafter{%
3596         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3597         #5%
3598       }%
3599       \edef\x{\endgroup
3600         \the\toks@\the\toks\tw@}%
3601       }\x
3602   }
3603   \let\H@old@schapter\@schapter
3604   \def\@schapter#1{%
3605     \begingroup
3606       \let\@mkboth\@gobbletwo
3607       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3608       \Hy@raisedlink{%
3609         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3610       }%
3611     \endgroup
3612     \H@old@schapter{#1}%
3613   }
3614   \@ifundefined{@chapter}{-}{%
3615     \let\Hy@org@chapter\@chapter
3616     \def\@chapter{%
3617       \def\Hy@next{%
3618         \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3619         \Hy@raisedlink{%
3620           \hyper@anchorstart{\@currentHref}\hyper@anchorend
3621         }%
3622       }%
3623       \ifnum\c@secnumdepth>\m@ne
3624         \@ifundefined{if@mainmatter}%
3625         \iftrue{\csname if@mainmatter\endcsname}%
3626         \let\Hy@next\relax
3627       \fi
3628     \fi
3629     \Hy@next
3630     \Hy@org@chapter
3631   }%
3632 }

```

```

3633 \let\H@old@part\@part
3634 \begingroup\expandafter\expandafter\expandafter\endgroup
3635 \expandafter\ifx\csname chapter\endcsname\relax
3636   \let\Hy@secnum@part\z@
3637 \else
3638   \let\Hy@secnum@part\m@ne
3639 \fi
3640 \def\@part{%
3641   \ifnum\Hy@secnum@part>\c@secnumdepth
3642     \phantomsection
3643   \fi
3644   \H@old@part
3645 }
3646 \let\H@old@spart\@spart
3647 \def\@spart#1{%
3648   \Hy@MakeCurrentHrefAuto{part*}%
3649   \Hy@raisedlink{%
3650     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3651   }%
3652   \H@old@spart{#1}%
3653 }
3654 \let\H@old@sect\@sect
3655 \def\@sect#1#2#3#4#5#6[#7]#8{%
3656   \ifnum #2>\c@secnumdepth
3657     \expandafter\@firstoftwo
3658   \else
3659     \expandafter\@secondoftwo
3660   \fi
3661   {%
3662     \Hy@MakeCurrentHrefAuto{section*}%
3663     \setlength{\Hy@SectionHShift}{#3}%
3664     \begingroup
3665     \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6} [#7]}%
3666     \toks\tw@\expandafter{%
3667       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3668       #8%
3669     }%
3670     \edef\x{\endgroup
3671       \the\toks@\the\toks\tw@}%
3672     }\x
3673   }{%
3674     \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6} [#7]{#8}%
3675   }%
3676 }
3677 }{}
3678 \expandafter\def\csname Parent-4\endcsname{}
3679 \expandafter\def\csname Parent-3\endcsname{}
3680 \expandafter\def\csname Parent-2\endcsname{}
3681 \expandafter\def\csname Parent-1\endcsname{}
3682 \expandafter\def\csname Parent0\endcsname{}
3683 \expandafter\def\csname Parent1\endcsname{}
3684 \expandafter\def\csname Parent2\endcsname{}
3685 \expandafter\def\csname Parent3\endcsname{}
3686 \expandafter\def\csname Parent4\endcsname{}

```

```

3687 %%
3688 %% End of file 'hgeneric-testphase.def'.
3689 </package>
3690 <*colorscheme>
3691 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3692 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3693 % linkcolor=
3694 %,filecolor=
3695 %,urlcolor=
3696 %,menucolor=
3697 %,runcolor=
3698 %,linkbordercolor=
3699 %,filebordercolor=
3700 %,urlbordercolor=
3701 %,menubordercolor=
3702 %,runbordercolor=
3703
3704 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }
3705 {
3706   linkcolor      = [rgb]{1,0,0}, %red
3707   filecolor      = [rgb]{0,1,1}, %cyan
3708   urlcolor       = [rgb]{1,0,1}, %magenta
3709   menucolor      = [rgb]{1, 0, 0}, %red
3710   runcolor       = [rgb]{0,1,1}, %cyan
3711   %-----
3712   linkbordercolor = [rgb]{1, 0 ,0 },
3713   filebordercolor = [rgb]{0, .5, .5},
3714   urlbordercolor  = [rgb]{0, 1, 1},
3715   menubordercolor = [rgb]{1, 0, 0},
3716   runbordercolor  = [rgb]{0, .7, .7}
3717 }
3718
3719 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3720 {
3721   linkcolor      = [rgb]{0,0.2,0.6},
3722   filecolor      = [rgb]{0.8,0,0.8},
3723   urlcolor       = [rgb]{0.8,0,0.8},
3724   menucolor      = [rgb]{0,0.2,0.6},
3725   runcolor       = [rgb]{0.8,0,0.8},
3726   %----- %-----
3727   linkbordercolor = [rgb]{0,0.2,0.6},
3728   filebordercolor = [rgb]{0.8,0,0.8},
3729   urlbordercolor  = [rgb]{0.8,0,0.8},
3730   menubordercolor = [rgb]{0,0.2,0.6},
3731   runbordercolor  = [rgb]{0.8,0,0.8}
3732 }
3733
3734 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3735 { %two colors: intern/extern
3736   linkcolor      = [rgb]{0.79216, 0, 0.12549},
3737   filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3738   urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3739   menucolor      = [rgb]{0.79216, 0, 0.12549 },
3740   runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },

```

```

3741 %----- %-----
3742 linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3743 filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3744 urlbordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3745 menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3746 runbordercolor = [rgb]{0.01961, 0.44314, 0.6902 }
3747 }
3748
3749 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3750 { %all darkgray
3751 linkcolor = [rgb]{0.4 ,0.4 ,0.4 },
3752 filecolor = [rgb]{0.4 ,0.4 ,0.4 },
3753 urlcolor = [rgb]{0.4 ,0.4 ,0.4 },
3754 menucolor = [rgb]{0.4 ,0.4 ,0.4 },
3755 runcolor = [rgb]{0.4 ,0.4 ,0.4 },
3756 %----- %-----
3757 linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3758 filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3759 urlbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3760 menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3761 runbordercolor = [rgb]{0.4 ,0.4 ,0.4 }
3762 }
3763
3764 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3765 { %dvipsnam.def
3766 linkcolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3767 filecolor = [rgb]{1, 0, 0}, %Red
3768 urlcolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3769 menucolor = [rgb]{1, 0, 0}, %Red
3770 runcolor = [rgb]{1, 0, 0}, %Red
3771 %----- %-----
3772 linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3773 filebordercolor = [rgb]{1, 0, 0}, %Red
3774 urlbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3775 menubordercolor = [rgb]{1, 0, 0}, %Red
3776 runbordercolor = [rgb]{1, 0, 0} %Red
3777 }
3778
3779 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3780 { %dvipsnam.def
3781 linkcolor = [rgb]{0.72, 0, 0}, %BrickRed
3782 filecolor = [rgb]{0, 1, 0}, %Green
3783 urlcolor = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3784 menucolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3785 runcolor = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3786 %----- %-----
3787 linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3788 filebordercolor = [rgb]{0, 1, 0}, %Green
3789 urlbordercolor = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3790 menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3791 runbordercolor = [rgb]{0.64, 0.08, 0.98} %Mulberry
3792 }
3793
3794

```

```

3795 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3796 {
3797   linkcolor      = [rgb]{0.50196, 0, 0.02353},
3798   filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3799   urlcolor       = [rgb]{0.54118, 0, 0.52941},
3800   menucolor      = [rgb]{0.44706, 0.45882, 0},
3801   runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3802 %----- %-----
3803   linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3804   filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3805   urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3806   menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3807   runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3808 }
3809
3810 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3811 {
3812   linkcolor      = [rgb]{0,0,0},
3813   filecolor      = [rgb]{0,0,0},
3814   urlcolor       = [rgb]{0,0,0},
3815   menucolor      = [rgb]{0,0,0},
3816   runcolor       = [rgb]{0,0,0},
3817 %----- %-----
3818   linkbordercolor = [rgb]{0,0,0},
3819   filebordercolor = [rgb]{0,0,0},
3820   urlbordercolor  = [rgb]{0,0,0},
3821   menubordercolor = [rgb]{0,0,0},
3822   runbordercolor  = [rgb]{0,0,0}
3823 }
3824 </colorscheme>

```


Index

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

Symbols	
<code>\#</code>	280, 793
<code>\\$</code>	279
<code>\%</code>	794
<code>\-</code>	2210
<code>\.</code>	555, 559, 561
@curropt commands:	
<code>\@curropt:</code>	2731, 2798
<code>\[</code>	2210
<code>\]</code> ...	20, 21, 31, 32, 33, 41, 45, 55, 75, 82, 89, 96, 103, 118, 123, 124, 132, 133, 142, 143, 144, 145, 146, 153, 161, 281, 282, 943, 2733, 2800, 3140
<code>_</code>	553, 555, 559, 561
<code>\]</code>	2210
A	
<code>\A</code>	553, 2210
<code>\Acrobatmenu</code>	18, 174
<code>\addcontentsline</code>	13
<code>\AddToDocumentProperties</code> ..	389, 2205, 2308
<code>\AddToHook</code>	427, 444, 450
<code>\AddToHookNext</code>	199
<code>\advance</code>	2663, 2738, 2782, 2783
<code>allcolors</code> (hypersetup key) ..	<u>1071</u>
<code>\author</code>	2
B	
<code>\b</code>	557
<code>\begingroup</code>	210, 278, 376, 2569, 2698, 2727, 2739, 2773, 3162, 3593, 3605, 3634, 3664
<code>\belowpdfbookmark</code>	3555
<code>\bgroup</code>	275, 376
<code>\BOOKMARK</code>	3558
<code>bookmarkstype</code> (hypersetup key) ..	13
bool commands:	
<code>\bool_if:NTF</code>	268, 297, 356, 713, 737, 757, 777, 788, 824, 889, 954, 1032, 1042, 1295, 1306
<code>\bool_if:nTF</code>	904, 1980, 2005, 3299, 3407
<code>\bool_lazy_and:nnTF</code>	424
<code>\bool_lazy_or:nnTF</code>	1325, 1374
<code>\bool_new:N</code> ...	204, 205, 539, 543, 547
<code>\bool_set_true:N</code>	548
bordercolormodel (hypersetup key) <i>13</i> , <u>1092</u>	
box commands:	
<code>\box_dp:N</code>	650, 2883, 2934, 2969
<code>\box_ht:N</code>	649, 2882, 2933, 2968
<code>\box_new:N</code>	464, 550
<code>\box_set_dp:Nn</code>	2177
<code>\box_set_ht:Nn</code>	2184
<code>\box_use:N</code> ...	1312, 2885, 2943, 2972
<code>\box_use_drop:N</code>	1317
<code>\box_wd:N</code>	648, 2881, 2932, 2967
<code>\l_tmpa_box</code>	2879, 2881, 2882, 2883, 2885, 2918, 2932, 2933, 2934, 2943, 2965, 2967, 2968, 2969, 2972
C	
<code>\catcode</code>	279, 280
<code>\char</code>	2817, 3014, 3026
<code>\chardef</code>	159
<code>\cite</code>	34
clist commands:	
<code>\clist_item:nn</code>	2211, 2240
<code>\clist_map_function:nN</code>	125, 134
<code>\clist_map_inline:nn</code>	2372
<code>\clist_map_variable:nNn</code> ..	2730, 2797
color commands:	
<code>\color_export:nnN</code> ..	41, 435, 991, 1113
<code>\color_select:n</code>	999, 1035, 1316, 2924, 2928
<code>\color_select:nn</code>	1005
<code>\color_set:nn</code>	2, 394, 1016
<code>\color_set:nnn</code>	2, 393, 1022
color names:	
<code>hyp/annot/file</code>	<u>536</u>
<code>hyp/annot/link</code>	<u>536</u>
<code>hyp/annot/menu</code>	<u>536</u>
<code>hyp/annot/run</code>	<u>536</u>
<code>hyp/annot/url</code>	<u>536</u>
<code>colorfile</code> (hypersetup key)	<u>1071</u>
<code>colorlink</code> (hypersetup key)	<u>1071</u>
<code>colorlinks</code> (hypersetup key)	<u>1048</u>
<code>colormenu</code> (hypersetup key)	<u>1071</u>
<code>colorrun</code> (hypersetup key)	<u>1071</u>
<code>colorscheme</code> (hypersetup key)	<u>1</u> , <u>1483</u>
<code>colorurl</code> (hypersetup key)	<u>1071</u>
cs commands:	
<code>\cs_generate_variant:Nn</code>	156, 157, 158, 391, 463, 591, 598, 821, 991, 1008, 1025, 2211
<code>\cs_gset:Npn</code>	1287
<code>\cs_gset_eq:NN</code>	410, 421
<code>\cs_if_exist:NTF</code>	13

<code>\l__hyp_annot_Named_bool</code>	954	<code>\l__hyp_filename_tmpa_tl</code> . . .	470,
<code>\c__hyp_annot_types_seq</code>			830, 832, 837, 838, 843, 898, 899, 917
.	477, 537, 541, 1071, 1372, 1470	<code>__hyp_href_pdf_aux:nn</code>	320, 323
<code>\l__hyp_annot_URI_bool</code>	777	<code>\l__hyp_href_pdf_destination_tl</code> .	
<code>\g__hyp_bordercolormodel_str</code>	207, 218, 257, 325
.	437, 536, 1096, 1115	<code>\l__hyp_href_pdf_page_tl</code> 208, 226, 854	
<code>__hyp_check_link_nesting:TF</code> . . .		<code>__hyp_href_run_aux:nn</code>	337, 340
.	622, 630,	<code>\l__hyp_href_run_parameter_tl</code> . . .	
	632, 716, 740, 759, 780, 827, 892, 957	209, 230, 342
<code>__hyp_citebordercolor_hook_-</code>		<code>__hyp_href_url_aux:n</code>	363, 366
<code>init:</code>	399, 412, 421	<code>__hyp_href_url_aux:nn</code>	304, 306
<code>__hyp_citecolor_hook_init:</code>		<code>\l__hyp_href_url_encode_bool</code> . . .	
.	397, 401, 410	204, 215, 253, 268, 297, 356
<code>__hyp_clist_display:n</code> . 118, 125, 134		<code>__hyp_href_url_format:</code>	
<code>__hyp_color_select:n</code>	210, 216, 258, 369
.	41, 992, 992, 1008	<code>\l__hyp_href_url_ismap_bool</code>	
<code>__hyp_color_select_aux:wn</code>	205, 229, 788
.	992, 996, 1003	<code>\l__hyp_href_url_protocol_tl</code> . . .	
<code>__hyp_color_set:nn</code> . . 40, 41, 397,		206, 217, 256, 308, 370
399, 434, 1009, 1009, 1025, 1076, 1112		<code>__hyp_if_outer_link:</code>	616
<code>__hyp_color_set_aux:nwn</code>		<code>__hyp_if_outer_link:TF</code>	632
.	1009, 1013, 1020	<code>\l__hyp_link_Contents_tl</code>	
<code>\c__hyp_colorscheme_daleif_prop</code> 3719		512, 520, 522, 529, 531
<code>\c__hyp_colorscheme_henryford_-</code>		<code>__hyp_link_goto_begin:nw</code>	
<code>prop</code>	3810	684, 723, 747
<code>\c__hyp_colorscheme_julian_prop</code> 3734		<code>__hyp_link_goto_end:</code> . 706, 724, 761	
<code>\c__hyp_colorscheme_phelype_prop</code>		<code>\g__hyp_linknestlevel_int</code>	
.	3795	615, 618, 715, 731, 739, 768,
<code>\c__hyp_colorscheme_szabolcsA_-</code>			779, 812, 826, 879, 891, 937, 956, 986
<code>prop</code>	3764	<code>g__hyp_linknestlevel_int</code>	615
<code>\c__hyp_colorscheme_szabolcsB_-</code>		<code>\c__hyp_map_annot_hyp_prop</code>	477
<code>prop</code>	3779	<code>\c__hyp_map_hyp_annot_prop</code>	
<code>\c__hyp_colorscheme_tivv_prop</code> . 3749		477, 1026,
<code>\l__hyp_dest_box</code>	32,		1099, 1139, 1166, 1174, 1185, 1212,
	550, 648, 649, 650, 2161, 2177, 2184		1220, 1289, 1397, 1433, 1443, 1556
<code>\l__hyp_dest_name_tmpa_tl</code> . . 470,		<code>__hyp_ocg_init:</code> <u>1231</u> , 1231, 1287, 1297	
	687, 688, 694, 698, 700, 703, 847, 860	<code>\l__hyp_optlang_regex</code> 2209, 2210, 2242	
<code>\l__hyp_dest_pdfremotestartview_-</code>		<code>l__hyp_page/Trans</code>	564
<code>tl</code>	496, 855, 1968, 1972	<code>__hyp_PageLabels_gp</code> <u>599</u> , 599, 612	
<code>\g__hyp_dest_pdfstartpage_tl</code> . . .		<code>\l__hyp_para_tmpa_tl</code> 473, 903, 906, 917	
. . .	496, 1979, 1981, 1988, 2006, 2013	<code>\l__hyp_para_tmpa_tl_{uuu}\l__hyp_-</code>	
<code>\g__hyp_dest_pdfstartview_tl</code> . . .		<code>text_tmpa_str_{uuu}\g__hyp_text_-</code>	
	496, 1981, 1988, 1999, 2003, 2006, 2013	<code>tmpa_str</code>	470
<code>\l__hyp_dest_pdfview_tl</code>		<code>__hyp_property_record:nn</code>	
.	535, 666, 675,	456, 457, 463, 2667
	2130, 2133, 2140, 2143, 2144, 2145,	<code>__hyp_secondoftwewithopt:w</code> <u>nn</u> . . .	
	2146, 2147, 2148, 2153, 2157, 2193	382, 383, 384, 385
<code>\c__hyp_dest_startview_regex</code> . . .		<code>__hyp_setup_info_date_key:nn</code> . . .	
.	551, 1966, 1997	2294, 2327, 2328
<code>\c__hyp_dest_undefined_tl</code>		<code>__hyp_setup_info_key:nn</code> . . 2212,	
.	476, 693, 694	2287, 2288, 2289, 2290, 2292, 2293	
<code>__hyp_destination:nn</code>		<code>__hyp_store_metadata:nn</code>	
.	32, 635, 635, 666, 675	386, 391, 1575, 2219,
			2262, 2307, 2322, 2331, 2347, 2415

_hyp_text_cleanup:N	571, 571, 585	bookmarkstype	13
\l_hyp_text_enc_dest_print_tl	499, 846	bordercolormodel	13, 1092
\l_hyp_text_enc_dest_tl	32, 499, 642, 699	colorfile	1071
\l_hyp_text_enc_file_print_tl	499, 897	colorlink	1071
\l_hyp_text_enc_info_print_tl	499, 519, 528, 596	colorlinks	1048
\l_hyp_text_enc_para_print_tl	499, 902	colormenu	1071
\l_hyp_text_enc_uri_print_tl	270, 273, 299, 302, 358, 361, 499, 785, 1567	colorrund	1071
_hyp_text_pdfstring:nnN	517, 526, 579, 579, 591, 593, 596, 640, 697, 783, 844, 895, 900, 1567	colorscheme	1, 1483
_hyp_text_pdfstring_info:nN	594, 594, 598, 2245, 2248, 2268, 2277	colorurl	1071
_hyp_text_purify:nN	567, 567, 584	debug	1507
_hyp_text_string_from_unicode:nN	575, 575, 586	destlabel	13
\g_hyp_text_tmpa_str	475, 587, 589	draft	1507
\l_hyp_text_tmpa_str	474, 584, 585, 586, 587	extension	13, 1525
\l_hyp_tmpa_box	464, 1299, 1312, 1317	file	10, 1556
\l_hyp_tmpa_int	464	fileborderstyle	14, 1185
\l_hyp_tmpa_seq	464, 1966, 1968, 1997, 1999, 2120, 2121, 2125, 2127, 2151, 2158, 2159, 2160, 2169, 2171, 2181, 2186, 2242, 2243, 2248, 2618, 2620, 2626	filecolor	1071
\l_hyp_tmpa_str	464, 2245, 2248, 2250, 2252, 2268, 2269, 2271, 2277, 2278, 2282	final	1507
\l_hyp_tmpa_tl	464, 643, 647, 655, 1116, 1120, 1567, 1568, 1573, 1964, 1966, 1995, 1997, 2127, 2128, 2135, 2218, 2219, 2220, 2240, 2245, 2261, 2262, 2268	hidefile	1459
\l_hyp_tmpb_tl	464, 2240, 2242	hidelink	1459
\l_hyp_uri_tmpa_tl	470, 786, 787	hidelinks	1459
hyp/anchor	661	hidemenu	1459
hyp/annot/file (color name)	536	hiderun	1459
hyp/annot/link (color name)	536	hideurl	1459
hyp/annot/menu (color name)	536	hypertextnames	1525
hyp/annot/run (color name)	536	link	10, 1556
hyp/annot/url (color name)	536	linkborder	14
hyp/text/pdfstring	566	linkborderstyle	14, 1185
\hypercalcbp	12, 18, 162	linkcolor	1071
\HyperDestNameFilter	13, 641, 698	linkfileprefix	1525
\hypersetup	1, 2, 5, 9, 10, 13, 19, 68, 101, 181	linktoc	1525
\hypersetup keys:		linktocpage	1525
allcolors	1071	localanchorname	1525
		menu	10, 1556
		menuborder	14
		menuborderstyle	14, 1185
		menucolor	1071
		naturalnames	1525
		nested-links	10
		nesting	14
		ocgcolorfile	1325
		ocgcolorlink	1325
		ocgcolorlinks	1325
		ocgcolormenu	1325
		ocgcolorrund	1325
		ocgcolorurl	1325
		pageanchor	1525
		pdfauthor	2209
		pdfborder	14
		pdfborderstyle	14, 1185
		pdfcreationdate	14, 2294
		pdfcreator	2209
		pdfencoding	1497
		pdfinfo	2361
		pdfkeywords	2209

pdflang	14, 2198	iow commands:	
pdflinkmargin	14	\iow_newline:	3058, 3064, 3077
pdfmetadate	14, 2294		
pdfmoddate	14, 2294	K	
pdfproducer	2209	\kern	3580
pdfremotestartview	12	keys commands:	
pdfstartview	12	\keys_define:nn	187, 195, 213, 251, 395, 626, 1048, 1073, 1080, 1092, 1101, 1126, 1141, 1160, 1187, 1206, 1329, 1343, 1355, 1378, 1384, 1399, 1428, 1459, 1472, 1483, 1497, 1507, 1512, 1525, 1537, 1558, 1563, 2116, 2198, 2214, 2257, 2296, 2311, 2329, 2333, 2361, 2413, 2419, 2432, 2452
pdfsubject	2209	\l_keys_key_str	192, 2282, 2517
pdftitle	2209	\keys_set:nn	184, 221, 240, 267, 296, 319, 336, 355, 405, 416, 1489, 1496, 2338, 2365, 2368, 2369, 2370, 2371, 2442, 2517
pdftrapped	2333	\keys_set_known:nn	2520
pdfversion	1497	\kvsetkeys	183, 2605
pdfview	12, 2116		
plainpages	1525	L	
run	10, 1556	\label	9, 13
runborder	14	\LayoutCheckField	2998
runborderstyle	14, 1185	\LayoutChoiceField	2768
runcolor	1071	\LayoutPushButtonField	2874, 2891
unicode	1497	\LayoutTextField	2706
url	10, 1556	\leavevmode	2707, 2804, 3578
urlborder	14	legacy commands:	
urlborderstyle	14, 1185	\legacy_if:nTF	169, 442, 452, 607, 2643, 2870, 2911, 2961, 2994, 3278
urlcolor	1071	\let	383, 384, 385, 733, 802, 814, 870, 881, 930, 939, 988, 2530, 2531, 2558, 2654, 2655, 2656, 2688, 2689, 2723, 2724, 2766, 3105, 3112, 3114, 3119, 3136, 3170, 3589, 3603, 3606, 3615, 3626, 3633, 3636, 3638, 3646, 3654
verbose	1507	link (hypersetup key)	10, 1556
hypertextnames (hypersetup key)	1525	linkborder (hypersetup key)	14
hypListbox internal commands:		linkborderstyle (hypersetup key)	14, 1185
_hypListbox	2786, 2845	linkcolor (hypersetup key)	1071
hypRadio internal commands:		linkfileprefix (hypersetup key)	1525
_hypRadio	2771, 2793	linktoc (hypersetup key)	1525
hypwritetorep internal commands:		linktocpage (hypersetup key)	1525
_hypwritetorep	3548	localanchorname (hypersetup key)	1525
		\long	7
I			
\ifcase	3332, 3456, 3491	M	
\ifcsname	3109	\MakeButtonField	2879, 2894, 2918, 2965, 2976
\ifdim	2736, 2775, 2776	\MakeChoiceField	2854
\ifnum	3163, 3332, 3338, 3369, 3456, 3461, 3491, 3496, 3623, 3641, 3656	\MakeFieldObject	2521
\iftrue	3625		
\ifx	2567, 2705, 2765, 3140, 3152, 3156, 3173, 3177, 3181, 3185, 3189, 3193, 3197, 3201, 3205, 3209, 3226, 3230, 3333, 3334, 3342, 3346, 3353, 3373, 3377, 3384, 3385, 3457, 3465, 3492, 3500, 3573, 3635		
\immediate	2551, 2554		
int commands:			
\int_compare:nNnTF	618, 2125, 2151, 2837, 3255, 3308, 3418, 3440		
\int_compare_p:nNn	3301, 3409		
\int_eval:n	853		
\int_gdecr:N	731, 768, 812, 879, 937, 986		
\int_gincr:N	715, 739, 779, 826, 891, 956		
\int_max:nn	854		
\int_new:N	468, 615		

<code>\MakeRadioField</code>	2836	<code>\paperwidth</code>	3
<code>\MakeTextField</code>	2715	<code>\PassOptionsToPackage</code> ...	454, 1517, 1522
<code>\mbox</code>	1309	pdf commands:	
menu (hypersetup key)	10, 1556	<code>\pdf_bdcobject:nn</code>	1311, 1314
menuborder (hypersetup key)	14	<code>\pdf_destination:nn</code>	27, 156, 654
menuborderstyle (hypersetup key)	14, 1185	<code>\pdf_destination:nnnn</code>	647
menucolor (hypersetup key)	1071	<code>\pdf_emc:</code>	1313, 1319
mode commands:		<code>\pdf_link_user:nnn</code>	2851
<code>\mode_if_horizontal:TF</code>	637, 659	<code>\pdf_name_from_unicode_e:n</code>	246, 834, 964
<code>\mode_leave_vertical:</code> ..	264, 293,	<code>\pdf_object_if_exist:nTF</code>	832
	316, 333, 352, 686, 796, 862, 919,	<code>\pdf_object_new:n</code>	1233,
	962, 2848, 2876, 2893, 2915, 2957, 3000	1234, 1235, 1236, 3045, 3046, 3047	
msg commands:		<code>\pdf_object_ref:n</code>	157, 843,
<code>\msg_error:nn</code>	426, 2890, 2975	1239, 1241, 1263, 1264, 1267, 1271,	
<code>\msg_info:nnn</code>	2226, 2231	1276, 1281, 1286, 2636, 2638, 3103	
<code>\msg_line_context:</code>	83	<code>\pdf_object_ref_last:</code>	864, 2447
<code>\g_msg_module_name_prop</code>	12	<code>\pdf_object_unnamed_write:nn</code>	863, 2443
<code>\msg_new:nnn</code> 51, 58, 63, 67, 71, 78,		<code>\pdf_object_write:nnn</code>	1237,
85, 92, 99, 106, 112, 119, 128, 138, 149		1243, 1253, 1265, 3048, 3090, 3097	
<code>\msg_new:nnnn</code>	15, 26, 37	<code>\pdf_pageobject_ref:n</code>	158, 1988, 2013, 2675
<code>\msg_warning:nn</code>	171, 1503	<code>\pdf_string_from_unicode:nnN</code> ..	577
<code>\msg_warning:nnn</code>	180, 690, 980	<code>\pdf_version:</code>	1649, 1768, 1797, 1831, 1846,
<code>\msg_warning:nnnn</code>		1872, 1897, 1932, 1948, 2053, 2080	
..... 1347, 1388, 1601, 1645,		<code>\pdf_version_compare:NnTF</code>	
1679, 1702, 1764, 1793, 1827, 1842,		1639, 1730, 1752, 1788, 1821, 1836,	
1855, 1868, 1893, 1928, 1944, 1971,		1862, 1887, 1915, 1938, 2043, 2070	
2002, 2035, 2049, 2076, 2110, 2192		<code>\pdf_version_compare_p:Nn</code>	908, 1326, 1375
<code>\msg_warning:nnnnn</code>	191,	<code>\pdf_version_major:</code>	
1419, 1453, 1545, 1621, 1658, 1745,	 168, 1327, 1350, 1376, 1392	
1778, 1803, 1879, 1906, 1957, 2062,		<code>\pdf_version_minor:</code> ..	167, 1350, 1392
2089, 2355, 2459, 2478, 2497, 2508		pdfannot commands:	
		<code>\pdfannot_box:nnnn</code>	
		.. 2710, 2821, 2880, 2931, 2966, 3030	
		<code>\pdfannot_box_ref_last:</code> ..	2565, 2581
		<code>\pdfannot_dict_put:nnn</code>	
		521, 530, 797, 864, 920, 965, 1117,	
		1152, 1176, 1198, 1222, 1404, 1435	
		<code>\pdfannot_dict_remove:nn</code> ..	1107,
		1147, 1168, 1193, 1214, 1412, 1445	
		<code>\pdfannot_link:nnn</code> .	798, 865, 921, 966
		<code>\pdfannot_link_goto_begin:nw</code> ..	703
		<code>\pdfannot_link_goto_end:</code>	708
		<code>\pdfannot_link_margin:n</code>	8, 1683
		<code>\c_pdfannot_link_types_seq</code>	545
		pdfauthor (hypersetup key)	2209
		<code>\pdfbookmark</code>	3557
		pdfborder (hypersetup key)	14
		pdfborderstyle (hypersetup key) .	14, 1185
		pdfcreationdate (hypersetup key)	14, 2294
N			
naturalnames (hypersetup key)	1525		
nested-links (hypersetup key)	10		
nesting (hypersetup key)	14		
<code>\newcommand</code>	165, 3552, 3554, 3556		
<code>\newcount</code>	2659, 2844		
<code>\NewDocumentCommand</code>	2521		
<code>\NewExpandableDocumentCommand</code>	382		
<code>\newlength</code>	3571		
<code>\noexpand</code> ...	2742, 2743, 2744, 2745, 3123		
<code>\nolinkurl</code>	4		
O			
ocgcolorfile (hypersetup key)	1325		
ocgcolorlink (hypersetup key)	1325		
ocgcolorlinks (hypersetup key)	1325		
ocgcolormenu (hypersetup key)	1325		
ocgcolorrun (hypersetup key)	1325		
ocgcolorurl (hypersetup key)	1325		
P			
pageanchor (hypersetup key)	1525		

\@ifundefined . 3128, 3587, 3614, 3624
\@mainaux 2551, 2554, 2574, 2593
\@mkboth 3606
\@ne 2663
\@nil 3126, 3139, 3145
\@part 3633, 3640
\@pdfauthor 23
\@pdfborder 163
\@pdfborderstyle 164
\@savsf 637, 659
\@schapter 3603, 3604
\@secondoftwo 3659
\@sect 3654, 3655
\@spart 3646, 3647
\@ssect 3589, 3590
\@tempdima . 2729, 2736, 2738, 2775,
2776, 2777, 2781, 2782, 2783, 2784
\@tempdimb 2735, 2736
\@typeset@protect 3573
\BKM@color 432, 438
\c@secnumdepth 3623, 3641, 3656
\calc@bm@number 3565
\check@bm@number 3564
\define@key 429
\Fld@additionalactions 3215, 3285,
3319, 3355, 3392, 3439, 3479, 3508
\Fld@align 3244, 3330, 3405
\Fld@altname . . 3152, 3154, 3226, 3228
\Fld@annotflags 3239,
3292, 3325, 3361, 3400, 3449, 3484
\Fld@annotnames 3224, 3241,
3294, 3327, 3363, 3402, 3451, 3486
\Fld@bcolor 3263, 3265, 3346,
3348, 3377, 3379, 3413, 3426, 3428
\fld@bcolor 3334
\Fld@bordercolor 3259,
3261, 3303, 3312, 3314, 3333, 3342,
3344, 3373, 3375, 3411, 3422, 3424,
3457, 3465, 3467, 3492, 3500, 3502
\Fld@borderstyle 3245,
3298, 3331, 3367, 3406, 3455, 3506
\Fld@borderwidth . 2782, 2783, 3245,
3298, 3331, 3367, 3406, 3455, 3506
\Fld@calculate@code . 2567, 3185, 3187
\Fld@calculate@sortkey 2578
\Fld@cbsymbol 3267
\Fld@charsize
. . . 2781, 3271, 3352, 3383, 3434, 3490
\Fld@checkequals 2733, 2800
\Fld@choices 3354
\Fld@color 3272, 3274, 3353, 3384, 3435
\Fld@default
. . . 2688, 2705, 2723, 2766, 2795,
2796, 2984, 3385, 3389, 3390, 3437
\Fld@flags 3243,
3296, 3329, 3365, 3404, 3453, 3488
\Fld@format@code 3177, 3179
\Fld@height
2691, 2712, 2715, 2726, 2784, 2823,
2836, 2854, 2903, 2951, 2987, 3032
\Fld@keystroke@code 3173, 3175
\Fld@listcount 2794, 2802, 2837, 2844
\Fld@mappingname 3156, 3158, 3230, 3232
\Fld@maxlen 3440, 3442
\Fld@menulength 2728, 2734, 2781
\Fld@name
. . . 2687, 2722, 2860, 2983, 3151, 3225
\Fld@onblur@code 3193, 3195
\Fld@onclick@code 3318
\Fld@onenter@code 3205, 3207
\Fld@onexit@code 3209, 3211
\Fld@onfocus@code 3189, 3191
\Fld@onmousedown@code . . . 3197, 3199
\Fld@onmouseup@code 3201, 3203
\Fld@pageobjref
. 2656, 2671, 3108, 3240,
3293, 3326, 3362, 3401, 3450, 3485
\Fld@radiosymbol 3381
\Fld@rotation 3255,
3257, 3301, 3308, 3310, 3332, 3338,
3340, 3369, 3371, 3409, 3418, 3420,
3456, 3461, 3463, 3491, 3496, 3498
\Fld@submitflags 3477
\Fld@validate@code 3181, 3183
\Fld@value
. . . 2689, 2705, 2724, 2765, 2766, 3438
\Fld@width 2690,
2704, 2711, 2715, 2725, 2764, 2775,
2777, 2822, 2836, 2854, 2872, 2902,
2913, 2950, 2962, 2986, 2996, 3031
\Fld@X@additionalactions
. 3172, 3217, 3220
\Form@action 3475
\H@old@part 3633, 3644
\H@old@schapter 3603, 3612
\H@old@sect 3654, 3665, 3674
\H@old@spart 3646, 3652
\H@old@ssect 3589, 3594
\href@ 275, 281
\href@split 281, 282
\Hy@abspage 611
\Hy@activeanchorfalse 682
\Hy@activeanchortrue 673
\Hy@AtBeginDocument 2532, 2548
\Hy@bookmarkstype 1585
\Hy@chapapp 3607, 3618
\Hy@colorlink 35
\Hy@currentbookmarklevel 3546

<code>\Hy@DisableOption</code>	173	<code>\HyAnn@Count</code>	2659, 2660, 2663, 2664, 2665, 2666, 2667, 2668, 2673, 2677
<code>\Hy@drafttrue</code>	1516	<code>\HyField@AddToFields</code>	2561, 2716, 2838, 2855, 2886, 2942, 2971, 3035
<code>\Hy@escapeform</code>	2709, 2806, 2850, 2878, 2917, 2964, 3002, 3110, 3117, 3120	<code>\HyField@afields</code>	2530
<code>\Hy@escapestring</code>	3112, 3114, 3119, 3122, 3123, 3126, 3127, 3136, 3139, 3175, 3179, 3183, 3187, 3191, 3195, 3199, 3203, 3207, 3211, 3267, 3318, 3381, 3437, 3438, 3475	<code>\HyField@AfterAuxOpen</code>	2532, 2558, 2591
<code>\Hy@finaltrue</code>	1521	<code>\HyField@AuxAddToCoFields</code>	2543, 2555, 2576
<code>\Hy@FormObjects</code>	2608, 3043, 3105	<code>\HyField@AuxAddToFields</code>	2536, 2552, 2594
<code>\Hy@gtemp</code>	3168, 3170	<code>\HyField@cofields</code>	2531
<code>\Hy@href</code>	275	<code>\HyField@FlagsCheckBox</code>	2993
<code>\Hy@href@nextactionraw</code>	239	<code>\HyField@FlagsChoice</code>	2774
<code>\Hy@href@page</code>	227	<code>\HyField@FlagsPushButton</code>	2869, 2909, 2960
<code>\Hy@linkfileprefix</code>	1530	<code>\HyField@FlagsRadioButton</code>	2770
<code>\Hy@linktoc</code>	1541	<code>\HyField@FlagsSubmit</code>	2910
<code>\Hy@MakeCurrentHref</code>	200	<code>\HyField@FlagsText</code>	2703
<code>\Hy@MakeCurrentHrefAuto</code>	3591, 3607, 3618, 3648, 3662	<code>\HyField@PDFChoices</code>	2847
<code>\Hy@next</code>	3617, 3626, 3629	<code>\HyField@SetKeys</code>	2699, 2740, 2743, 2762, 2862, 2905, 2953, 2988
<code>\Hy@numberline</code>	166, 3547	<code>\HyPat@ObjRef</code>	3519, 3526, 3533, 3540
<code>\Hy@org@chapter</code>	3615, 3630	<code>\hyper@@link</code>	283
<code>\Hy@OutlineName</code>	3563	<code>\hyper@anchor</code>	661
<code>\Hy@OutlineRerunCheck</code>	3561	<code>\hyper@anchorend</code>	661, 3582, 3609, 3620, 3650
<code>\Hy@pdfmajorversion</code>	168	<code>\hyper@anchorstart</code>	661, 3582, 3609, 3620, 3650
<code>\Hy@pdfminorversion</code>	167	<code>\hyper@link</code>	34, 35, 711
<code>\Hy@pdfstringtrue</code>	30, 582	<code>\hyper@linkend</code>	34, 35, 755
<code>\Hy@pdfversion</code>	3163	<code>\hyper@linkfile</code>	325, 822
<code>\Hy@pstringdef</code>	592, 3143	<code>\hyper@linklaunch</code>	38, 342, 887, 946
<code>\Hy@PutCatalog</code>	599	<code>\hyper@linknamed</code>	40, 176, 952
<code>\Hy@raisedlink</code>	3581, 3608, 3619, 3649	<code>\hyper@linkstart</code>	34, 35, 735
<code>\Hy@RestoreLastskip</code>	658	<code>\hyper@linkurl</code>	308, 369, 775
<code>\Hy@ReturnAfterFi</code>	7, 3144	<code>\hyper@normalise</code>	275, 304, 320, 337, 363, 376
<code>\Hy@safe@activestru</code>	795, 2570	<code>\HyPL@Labels</code>	601, 611
<code>\Hy@SaveLastskip</code>	638	<code>\HyPL@storePageLabel</code>	599
<code>\Hy@secnum@part</code>	3636, 3638, 3641	<code>\HyPsd@SanitizeForOutFile</code>	3550
<code>\Hy@SectionAnchorHref</code>	3572, 3596, 3667	<code>\if@filesw</code>	2550, 2573, 2592
<code>\Hy@SectionHShift</code>	3571, 3580, 3592, 3663	<code>\ifFld@combo</code>	2749, 2779
<code>\Hy@StepCount</code>	2734, 2802	<code>\ifFld@hidden</code>	2704, 2764
<code>\Hy@temp</code>	2571, 2586, 3125, 3126	<code>\ifFld@multiline</code>	2692
<code>\Hy@unicodefalse</code>	3165	<code>\ifFld@popdown</code>	2750
<code>\Hy@VerboseAnchor</code>	639	<code>\ifFld@radio</code>	2746, 2769
<code>\Hy@VerboseLinkStart</code>	718, 742	<code>\ifHy@implicit</code>	3567
<code>\Hy@VerboseLinkStop</code>	726, 763, 805, 872, 932, 976	<code>\ifHy@pdfescapeform</code>	3111, 3121
<code>\Hy@VersionChecked</code>	159	<code>\ifHy@unicode</code>	3164
<code>\Hy@WrapperDef</code>	635	<code>\kv@set@family@handler</code>	179
<code>\Hy@xspace@end</code>	723, 804, 871, 931, 975	<code>\m@ne</code>	3623, 3638
<code>\HyAnn@AbsPageLabel</code>	2655, 2661, 2708, 2805, 2849, 2877, 2916, 2963, 3001	<code>\OBJ@OCG@view</code>	165
		<code>\p@</code>	2738
		<code>\pdf@ifdraftmode</code>	2606

