

The `hyperref-generic` module

A generic driver for `hyperref`

The L^AT_EX Project*

Version 0.96l, released 2024-09-13

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 `pdfflatex`, `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:

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

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

4 Dates

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

5 PDF page size (mediabox)

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

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

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

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

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

6 Commands to create “external” references

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

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

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

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

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

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

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

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

6.1 Special problem: links to files

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

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

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

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

6.2 Splits

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

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

For this reason three new commands have been create:

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

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

6.3 Options

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

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

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

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

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

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

Some main changes are

- The default colors have been changed.

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

7.1 Background information

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

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

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

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

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

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

7.2 New Keys

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

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

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

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

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

The default is `phelype`.

7.3 Public interfaces

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

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

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

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

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

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

7.4 Changed behaviour

colorlinks `colorlinks` 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 new driver offers also boolean keys `link`, `url`, `file`, `run` and `menu` which allow to locally disable a link type. So e.g. `\hypersetup{link=false}\ref{abc}` will give a reference without link (this is naturally also possible with `\ref*{abc}`). This disables also all hooks of the link type, so the link is for example no longer colored. It also removes the implicit grouping of the content.

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

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

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

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

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

13.1 Names of destinations

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

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

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

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

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

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

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

14 Assorted key descriptions

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

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

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

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

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

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

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

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

will create

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

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 dvi_{pdf}mx 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}[2024-09-13 v0.961 %
4 generic Hyperref driver for the LaTeX PDF management testphase bundle]
5
6 \RequirePackage{etoolbox} %why?

```

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

```

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

```

1 messages

Redirect the message name:

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

At first a message for the testing of the resource management

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

The pdfversion should be set in \DocumentMetadata

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

A generic message for ignored keys.

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

```
57 }
```

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

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

pdf/A message for not allowed Named actions

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

A message if the destination name is empty.

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

A message if the destination check fails

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

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

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



```

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

```

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

```

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

```

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

```

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

```

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

```

149 \msg_new:nnn

```

```

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

```

2 Variants

```

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

```

3 Overwriting/providing commands from hyperref

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

```

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

```

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

\hypercalcbp

```

162 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n

```

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

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

```

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

```

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

```

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

```

\Acrobatmenu should use the new internal link command

```

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

```

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

```

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

```

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_tmpa_seq 464 \box_new:N \l__hyp_tmpa_box
\l__hyp_tmpa_int 465 \tl_new:N \l__hyp_tmpa_tl
\l__hyp_tmpa_box 466 \seq_new:N \l__hyp_tmpa_seq
\l__hyp_tmpa_str 467 \int_new:N \l__hyp_tmpa_int
468 \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 469 \tl_new:N \l__hyp_dest_name_tmpa_tl
\l__hyp_filename_tmpa_tl 470 \tl_new:N \l__hyp_uri_tmpa_tl
\__hyp_text_tmpa_str_{uu}\g__hyp_text_tmpa_str 471 \tl_new:N \l__hyp_filename_tmpa_tl
472 \tl_new:N \l__hyp_para_tmpa_tl
473 \str_new:N \l__hyp_text_tmpa_str
474 \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.

```
475 \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
476 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
477 {url,link,file,menu,run}
478 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
479 {
480   URI    = url,
481   GoTo   = link,
482   GoToR  = file,
483   Named  = menu,
484   Launch = run
485 }
486 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
487 {
488   url    = URI,
489   link   = GoTo,
490   file   = GoToR,
491   menu   = Named,
492   run    = Launch
493 }
494
```

(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
495 \tl_new:N \g__hyp_dest_pdfstartpage_tl
496 \tl_new:N \g__hyp_dest_pdfstartview_tl
497 \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
498 \tl_new:N \l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_dest_tl
499 \tl_new:N \l__hyp_text_enc_info_print_tl
\l__hyp_text_enc_dest_print_tl
500 \tl_new:N \l__hyp_text_enc_dest_tl
\l__hyp_text_enc_file_print_tl
501 \tl_new:N \l__hyp_text_enc_dest_print_tl
\l__hyp_text_enc_para_print_tl
502 \tl_new:N \l__hyp_text_enc_file_print_tl
503 \tl_new:N \l__hyp_text_enc_para_print_tl
```

```

504
505 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
506 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
507 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
508 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
509 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
510 \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 define \l__hyp_link_Contents_tl

```

511 \tl_new:N\l__hyp_link_Contents_tl
512 \socket_new:nn {hyp/link/GoTo/Contents}{2}
513 \socket_new:nn {hyp/link/URI/Contents}{1}
514 \socket_new_plug:nnn {hyp/link/GoTo/Contents}{default}
515 {
516   \__hyp_text_pdfstring:eoN
517   { Go-to-destination~#1 }
518   { \l__hyp_text_enc_info_print_tl }
519   \l__hyp_link_Contents_tl
520 }
521 \socket_new_plug:nnn {hyp/link/URI/Contents}{default}
522 {
523   \__hyp_text_pdfstring:eoN
524   { #1 }
525   { \l__hyp_text_enc_info_print_tl }
526   \l__hyp_link_Contents_tl
527 }
528 \socket_assign_plug:nn{hyp/link/GoTo/Contents}{default}
529 \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.

```

530 \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**

```

531 \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.

```

532 \seq_map_inline:Nn \c__hyp_annot_types_seq
533 {
534   \bool_new:c {l_hyp_annot_color#1_bool}
535 }
```

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

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

```

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

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

`\not_Named_bool_####\l_hyp_annot_Launch_bool`

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

```

540 \seq_map_inline:Nn \c_pdfannot_link_types_seq
541 {
542   \bool_new:c {l__hyp_annot_#1_bool}
543   \bool_set_true:c {l__hyp_annot_#1_bool}
544 }
```

(End of definition for \l__hyp_annot_GoTo_bool \l__hyp_annot_URI_bool \l__hyp_annot_GoToR_ - bool \l__hyp_annot_Named_bool \l__hyp_annot_Launch_bool.)

7.5 Boxes

`\l__hyp_dest_box`

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

```

545 \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.

```

546 \regex_const:Nn \c__hyp_dest_startview_regex
547 {
548   \A\ *
549   (?
550     (?:XYZ (?:\ +(?:?:\d+|\d*\.\d+)|null)){3}\ )
551   |
552   (?:Fit\b|FitB\b)
553   |
554   (?:?:FitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1}
555   |
556   (?:FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
557   )
558 }
```

(End of definition for `\c__hyp_dest_startview_regex`.)

7.7 PDF dictionaries

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

```
559 \pdfdict_new:n {l__hyp_page/Trans}
560 \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`

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

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

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

`__hyp_text_purify:nN`

```
562 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
563 {
564   \str_set:Ne #2 {\text_purify:n { #1 } }
565 }
```

(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`

```
566 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
567 {
568
569 }
```

(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`

```
570 \cs_new_protected:Npn \_hyp\_text\_string\_from\_unicode:nN #1 #2
571 {
572   \pdf\_string\_from\_unicode:nVN { #1 } #2 #2
573 }
```

(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`

```
574 \cs_new_protected:Npn \_hyp\_text\_pdfstring:nnN #1 #2 #3
575 {
576   \group\_begin:
577   \Hy@pdfstringtrue
578   \hook\_use:n {hyp/text/pdfstring}
579   \_hyp\_text\_purify:nN { #1 } \l\_hyp\_text\_tmpa\_str
580   \_hyp\_text\_cleanup:N      \l\_hyp\_text\_tmpa\_str
581   \_hyp\_text\_string\_from\_unicode:nN { #2 } \l\_hyp\_text\_tmpa\_str
582   \str\_gset\_eq:NN \g\_hyp\_text\_tmpa\_str\l\_hyp\_text\_tmpa\_str
583   \group\_end:
584   \str\_set\_eq:NN #3 \g\_hyp\_text\_tmpa\_str
585 }
586 \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

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

This is a special version for info keys:

`_hyp_text_pdfstring_info:nN`

```
589 \cs\_new\_protected:Npn \_hyp\_text\_pdfstring\_info:nN #1 #2
590 {
591   \_hyp\_text\_pdfstring:noN { #1 }{\l\_hyp\_text\_enc\_info\_print\_tl } #2
592 }
```

(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
593 \cs_new_protected:Npn \__hyp_PageLabels_gpush:
594   {
595   \pdfmanagement_add:nne {Catalog} {PageLabels}{<</Nums [\HyPL@Labels]>>}
596   }
597
598 \def \Hy@PutCatalog #1 {}
599
600
601 \legacy_if:nT { Hy@pdfpagelabels }
602   {
603   \cs_set_protected:Npn \HyPL@StorePageLabel #1
604     {
605     \tl_gput_right:Nc \HyPL@Labels { \the\Hy@abspage<<#1>> }
606     \__hyp_PageLabels_gpush:
607     }
608   }
```

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

10 Core Hyperref Commands

Every `hyperref` has to define eight core command:

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

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

10.1 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`

```

609 \int_new:N \g__hyp_linknestlevel_int
(End of definition for g__hyp_linknestlevel_int.)
610 \prg_new_conditional:Npnn \__hyp_if_outer_link: {TF}
611 {
612   \int_compare:nNnTF { \g__hyp_linknestlevel_int } > {1}
613     { \prg_return_false: }
614     { \prg_return_true: }
615 }
616 \cs_new:Npn \__hyp_check_link_nesting:TF #1 #2
617 {
618   \use_i:nn {#1}{#2}
619 }
620 \keys_define:nn { hyp }
621 {
622   nested-links .choice:,
623   nested-links / true .code:n =
624     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \use_i:nn },
625   nested-links / false .code:n =
626     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \__hyp_if_outer_link:TF },
627   nested-links .default:n = {true}
628 }

```

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`

```

629 \Hy@WrapperDef \__hyp_destination:nn #1 #2
630 {
631   \mode_if_horizontal:T { \@savsf\spacefactor }
632   \Hy@SaveLastskip      %defined in hyperref
633   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging

```



```

634 \__hyp_text_pdfstring:eoN
635   { \HyperDestNameFilter{#1} }
636   { \l__hyp_text_enc_dest_tl }
637   \l__hyp_tmpa_tl
638 \str_if_eq:nnTF {#2} {fitrbox}
639   {
640     \exp_args:NV
641     \pdf_destination:nnnn \l__hyp_tmpa_tl
642     { \box_wd:N \l__hyp_dest_box }
643     { \box_ht:N \l__hyp_dest_box }
644     { \box_dp:N \l__hyp_dest_box }
645   }
646   {
647     \exp_args:NV
648     \pdf_destination:nf
649     { \l__hyp_tmpa_tl }
650     { #2 }
651   }
652 \Hy@RestoreLastskip %defined in hyperref
653 \mode_if_horizontal:T { \spacefactor\@savsf }
654 }

```

(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 655 \tl_new:N\l_hyp_current_dest_name_tl
\hyper@anchorend 656 \hook_new:n{hyp/anchor}
hyp/anchor 657 \cs_new_protected:Npn \hyper@anchor #1
\l_hyp_current_dest_name_tl 658 {
659   \exp_args:NnV
660   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
661   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
662   \hook_use:n{hyp/anchor}
663 }
664
665 \cs_new_protected:Npn \hyper@anchorstart #1
666 {
667   \Hy@activeanchortrue
668   \exp_args:NnV
669   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
670   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
671   \hook_use:n{hyp/anchor}
672 }
673
674 \cs_new_protected:Npn \hyper@anchorend
675 {
676   \Hy@activeanchorfalse
677 }

```

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

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:

```
678 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
679 {
680   \mode_leave_vertical:
681   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
682   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
683     {
684       \msg_warning:nne
685         { hyp }
686         { empty-destination-name }
687         { \c__hyp_dest_undefined_tl }
688       \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
689     }
690     {
691       \__hyp_text_pdfstring:eoN
692       { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
693       { \l__hyp_text_enc_dest_tl }
694       \l__hyp_dest_name_tmpa_tl
695     }
696   \exp_args:No
697   \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
698 }
699
700 \cs_new_protected:Npn \__hyp_link_goto_end:
701 {
702   \pdfannot_link_goto_end:
703 }
```

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

```

704
705 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
706 {
707   \bool_if:NTF \l__hyp_annot_GoTo_bool
708   {
709     \int_gincr:N\g__hyp_linknestlevel_int
710     \__hyp_check_link_nesting:TF
711     {
712       \Hy@VerboseLinkStart{#1}{#2}
713       \group_begin:
714       \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
715       \socket_use:nmn{hyp/link/GoTo/Contents}{#2}{#3}
716       \pdfannot_dict_put:nne {link/GoTo}{Contents}
717       {\l_hyp_link_Contents_tl}
718       \hook_use:n {hyp/link/#1}
719       \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
720       \__hyp_link_goto_end:
721       \group_end:
722       \Hy@VerboseLinkStop
723     }
724     {
725       \group_begin: #3\group_end:
726     }
727     \int_gdecr:N\g__hyp_linknestlevel_int
728   }
729   {{\let\protect\relax#3}}
730 }
731 \cs_new_protected:Npn \hyper@linkstart #1 #2 % #1 context, #2=destination name
732 {
733   \bool_if:NT \l__hyp_annot_GoTo_bool
734   {
735     \int_gincr:N\g__hyp_linknestlevel_int
736     \__hyp_check_link_nesting:TF
737     {
738       \Hy@VerboseLinkStart{#1}{#2}% only for debug
739       \group_begin:
740       \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
741       \hook_use:n {hyp/link/#1}
742       \__hyp_link_goto_begin:nw {#2}
743     }
744     {
745       \group_begin:

```

```

746     }
747   }
748 }
749
750 \cs_new_protected:Npn \hyper@linkend
751 {
752   \bool_if:NT \l__hyp_annot_GoTo_bool
753   {
754     \__hyp_check_link_nesting:TF
755     {
756       \__hyp_link_goto_end:
757       \group_end:
758       \Hy@VerboseLinkStop
759     }
760     {
761       \group_end:
762     }
763     \int_gdecr:N\g__hyp_linknestlevel_int
764   }
765 }

```

10.4 URI links

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

```

766 \pdfdict_new:n {l_hyp/annot/A/URI}
767 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
768 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
769
770 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
771 {
772   \bool_if:NTF \l__hyp_annot_URI_bool
773   {
774     \int_gincr:N\g__hyp_linknestlevel_int
775     \__hyp_check_link_nesting:TF
776     {
777       \group_begin:
778       \__hyp_text_pdfstring:eoN
779       { #2}
780       { \l__hyp_text_enc_uri_print_tl }
781       \l__hyp_uri_tmpa_tl
782       \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
783       \bool_if:NT \l__hyp_href_url_ismap_bool
784       {
785         \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
786       }

```

This socket defines \l__hyp_link_Contents_tl

```

787     \socket_use:nn{hyp/link/URI/Contents}{#2}
788     \pdfannot_dict_put:nne {link/URI}{Contents}{\l__hyp_link_Contents_tl}
789     \cs_set_eq:NN \# \c_hash_str
790     \cs_set_eq:NN \% \c_percent_str
791     \Hy@safe@activetrue
792     \mode_leave_vertical:
793     \pdfannot_dict_put:nne {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}

```

```

794     \pdfannot_link:nen { URI }
795     {
796     }
797     {
798     \let\protect\relax
799     #1
800     \Hy@xspace@end
801     \Hy@VerboseLinkStop %where is the start??
802     }
803     \group_end:
804     }
805     {
806     \group_begin: #1 \group_end:
807     }
808     \int_gdecr:N\g__hyp_linknestlevel_int
809     }
810     {\let\protect\relax#1}}
811 }
812

```

10.5 GoToR Links files

```

813 \pdfdict_new:n {l_hyp/annot/A/GoToR}
814 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
815 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
816
817 \cs_generate_variant:Nn \pdffile_embed_file:nnn {noe}
818 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
819 {
820   \bool_if:NTF \l__hyp_annot_GoToR_bool
821   {
822     \int_gincr:N\g__hyp_linknestlevel_int
823     \__hyp_check_link_nesting:TF
824     {
825       \group_begin:
826       \tl_set:Ne \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
827       \exp_args:Ne
828       \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
829       {
830         \pdfdict_put:nne { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n
831         \pdffile_embed_file:noe
832         {}
833         {\l__hyp_filename_tmpa_tl }
834         {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
835       }
836       \pdfdict_put:nne
837       {l_hyp/annot/A/GoToR}
838       {F}
839       {\pdf_object_ref:e {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
840       \__hyp_text_pdfstring:nnN
841       { #3 }
842       { \l__hyp_text_enc_dest_print_tl }
843       \l__hyp_dest_name_tmpa_tl
844       \tl_if_blank:eTF {#3}

```

```

845     {
846       \pdfdict_put:nne {l_hyp/annot/A/GoToR}{D}
847       {
848         [
849           \int_eval:n
850           { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
851           /\l__hyp_dest_pdfremotestartview_tl
852         ]
853       }
854     }
855     {
856       \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
857     }
858     \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>

```

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

```

10.6 Launch links

We define `\hyper@linklaunch` for naming consistency

```

879 \pdfdict_new:n {l_hyp/annot/A/Launch}
880 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
881 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
882
883 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
884 {
885   \bool_if:NTF \l__hyp_annot_Launch_bool
886   {
887     \int_gincr:N\g__hyp_linknestlevel_int
888     \__hyp_check_link_nesting:TF
889     {
890       \group_begin:
891       \__hyp_text_pdfstring:nnN
892       { #1 }

```

```

893         { \l__hyp_text_enc_file_print_tl }
894         \l__hyp_filename_tmpa_tl
895     \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
896     \__hyp_text_pdfstring:noN
897     { #3 }
898     { \l__hyp_text_enc_para_print_tl }
899     \l__hyp_para_tmpa_tl
900     \bool_if:nTF
901     {
902         \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}{}
903         ||
904         \pdf_version_compare_p:Nn > {1.9}
905     }
906     {
907         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
908     }
909     {
910         \pdfdict_put:nne
911         {l_hyp/annot/A/Launch}
912         {Win}
913         {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
914     }
915     \mode_leave_vertical:
916     \pdfannot_dict_put:nne {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}
917     \pdfannot_link:nen
918     { Launch }
919     {
920         % /A
921         % <<
922         % \pdfdict_use:n {l_hyp/annot/A/Launch}
923         % >>
924     }
925     {
926         \let\protect\relax
927         #2\Hy@xspace@end
928         \Hy@VerboseLinkStop %where is the start??
929     }
930     \group_end:
931     }
932     { \group_begin: #2 \group_end: }
933     \int_gdecr:N\g__hyp_linknestlevel_int
934     }
935     {{\let\protect\relax#2}}
936     }

```

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

```

937 \use:e
938 { % filename, anchor text, linkname
939   \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str #1 \exp_not:N \ \ #2 #3
940 }
941 {
942   \hyper@linklaunch {#1}{#2}{#3}
943 }

```

10.7 Named links (menu)

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

```
944 \pdfdict_new:n {l_hyp/annot/A/Named}
945 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
946 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
947
948 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
949 {
950   \bool_if:NTF \l__hyp_annot_Named_bool
951   {
952     \int_gincr:N\g__hyp_linknestlevel_int
953     \__hyp_check_link_nesting:TF
954     {
955       \group_begin:
956       \pdfmeta_standard_verify:nnTF {named_actions}{#1}
957       {
958         \mode_leave_vertical:
959         \pdfdict_put:nne {l_hyp/annot/A/Named}{N}
960         {\pdf_name_from_unicode_e:n{#1}}
961         \pdfannot_dict_put:nne {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}
962         \pdfannot_link:nnn { Named }
963         {
964           % /A
965           % <<
966           % \pdfdict_use:n { l_hyp/annot/A/Named }
967           % >>
968         }
969         {
970           #2
971           \Hy@xspace@end
972           \Hy@VerboseLinkStop
973         }
974       }
975       {
976         \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
977         #2
978       }
979       \group_end:
980     }
981     { \group_begin: #2 \group_end: }
982     \int_gdecr:N\g__hyp_linknestlevel_int
983   }
984   {\let\protect\relax#2}}
985 }
986
```

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.

```
987 \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`.

```
988 \cs_new_protected:Npn \_hyp_color_select:n #1
989 {
990   \tl_if_head_eq_charcode:nNTF {#1}[ %]
991   {
992     \_hyp_color_select_aux:wn #1
993   }
994   {
995     \color_select:n {#1}
996   }
997 }
998
999 \cs_new_protected:Npn \_hyp_color_select_aux:wn [#1] #2
1000 {
1001   \color_select:nn {#1}{#2}
1002 }
1003
1004 \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`.

```
1005 \cs_new_protected:Npn \_hyp_color_set:nn #1 #2
1006 {
1007   \tl_if_head_eq_charcode:nNTF {#2}[ %]
1008   {
1009     \_hyp_color_set_aux:nwn { #1 } #2
1010   }
1011   {
1012     \color_set:nn {#1} {#2}
1013   }
1014 }
1015
1016 \cs_new_protected:Npn \_hyp_color_set_aux:nwn #1 [#2] #3
1017 {
1018   \color_set:nnn {#1}{#2}{#3}
1019 }
1020
1021 \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.

```
1022 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1023   {
1024     \hook_gput_code:nnn
1025       {pdfannot/link/#2/begin}
1026       {hyp/color}
1027     {
1028       \bool_if:cT { l_hyp_annot_color#1_bool }
1029       {
1030         \group_begin:
1031         \color_select:n { hyp/color/#1}
1032       }
1033     }
1034     \hook_gput_code:nnn
1035       {pdfannot/link/#2/end}
1036       {hyp/color}
1037     {
1038       \bool_if:cT { l_hyp_annot_color#1_bool }
1039       {
1040         \group_end:
1041       }
1042     }
1043   }
```

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

```
1044 \keys_define:nn { hyp }
1045   {
1046     ,colorlinks .choice:
1047     ,colorlinks / true .meta:n =
1048       {
1049         ,pdfborder={0~0~0}
1050         ,pdfborderstyle=
1051         ,colorurl =#1
1052         ,colorlink =#1
1053         ,colorryn =#1
1054         ,colormenu =#1
1055         ,colorfile =#1
1056       }
1057     ,colorlinks / false .meta:n =
1058       {
1059         ,colorurl =#1
1060         ,colorlink =#1
1061         ,colorryn =#1
1062         ,colormenu =#1
1063         ,colorfile =#1
1064       }
1065     ,colorlinks .default:n = {true}
1066   }
```

```

colorurl (setup key)
colorlink (setup key) 1067 \seq_map_inline:Nn \c__hyp_annot_types_seq
colorrun (setup key) 1068 {
colormenu (setup key) 1069   \keys_define:nn { hyp }
colorfile (setup key) 1070   {
  urlcolor (setup key) 1071     ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
linkcolor (setup key) 1072     ,#1color .code:n =   { \__hyp_color_set:ne {hyp/color/#1}{##1} }
  runcolor (setup key) 1073   }
menucolor (setup key) 1074 }
filecolor (setup key) 1075
allcolors (setup key) 1076 \keys_define:nn { hyp }
1077 {
1078   ,allcolors .meta:n =
1079   {
1080     ,urlcolor=#1
1081     ,linkcolor=#1
1082     ,runcolor=#1
1083     ,filecolor=#1
1084     ,menucolor=#1
1085   }
1086   ,allcolors .value_required:n = true
1087 }

```

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)

```

1088 \keys_define:nn { hyp }
1089 {
1090   ,bordercolormodel .choices:nn =
1091   {rgb,cmyk}
1092   { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
1093   ,bordercolormodel .initial:n ={rgb}
1094 }

1095 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1096 {
1097   \keys_define:nn { hyp }
1098   {
1099     #1bordercolor .code:n =
1100     {
1101       \tl_if_empty:nTF { ##1 }
1102       {
1103         \pdfannot_dict_remove:nn
1104         {link/#2}
1105         { C }
1106       }
1107       {
1108         \__hyp_color_set:ne {hyp/color/#1border}{##1}
1109         \color_export:nVN

```

```

1110         {hyp/color/#1border}
1111         \g__hyp_bordercolormodel_str
1112         \l__hyp_tmpa_tl
1113         \pdfannot_dict_put:nne
1114         {link/#2}
1115         { C }
1116         { [\l__hyp_tmpa_tl] }
1117     }
1118 }
1119 }
1120 }
1121
1122 \keys_define:nn { hyp }
1123 {
1124     ,allbordercolors .meta:n =
1125     {
1126         ,linkbordercolor=#1
1127         ,urlbordercolor =#1
1128         ,filebordercolor=#1
1129         ,menubordercolor=#1
1130         ,runbordercolor =#1
1131     }
1132     ,allbordercolors .value_required:n = true
1133 }
1134

```

11.3.2 Borderwidth and -arc

```

1135 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1136 {
1137     \keys_define:nn { hyp }
1138     {
1139         #1border .code:n =
1140         {
1141             \tl_if_empty:nTF { ##1 }
1142             {
1143                 \pdfannot_dict_remove:nn
1144                 {link/#2}
1145                 { Border }
1146             }
1147             {
1148                 \pdfannot_dict_put:nnn
1149                 {link/#2}
1150                 { Border }
1151                 { [##1] }
1152             }
1153         }
1154     }
1155 }
1156 \keys_define:nn { hyp }
1157 {
1158     ,pdfborder .code:n =
1159     {
1160         \tl_if_empty:nTF { #1 }

```

```

1161     {
1162       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1163       {
1164         \pdfannot_dict_remove:nn
1165         {link/##2}
1166         { Border }
1167       }
1168     }
1169     {
1170       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1171       {
1172         \pdfannot_dict_put:nnn
1173         {link/##2}
1174         { Border }
1175         { [#1] }
1176       }
1177     }
1178   }
1179   ,pdfborder .initial:n = {0~0~1},
1180 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1181 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1182 {
runborderstyle (setup key) 1183   \keys_define:nn { hyp }
fileborderstyle (setup key) 1184   {
menuborderstyle (setup key) 1185     #1borderstyle .code:n =
1186     {
1187       \tl_if_empty:nTF { ##1 }
1188       {
1189         \pdfannot_dict_remove:nn
1190         {link/#2}
1191         { BS }
1192       }
1193       {
1194         \pdfannot_dict_put:nnn
1195         {link/#2}
1196         { BS }
1197         { <<##1>> }
1198       }
1199     }
1200   }
1201 }
1202 \keys_define:nn { hyp }
1203 {
1204   ,pdfborderstyle .code:n =
1205   {
1206     \tl_if_empty:nTF { #1 }
1207     {
1208       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1209       {

```

```

1210         \pdfannot_dict_remove:nn
1211         {link/##2}
1212         { BS }
1213     }
1214 }
1215 {
1216     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1217     {
1218         \pdfannot_dict_put:nnn
1219         {link/##2}
1220         { BS }
1221         { <<#1>> }
1222     }
1223 }
1224 }
1225 ,pdfborderstyle .initial:n = {},
1226 }

```

11.4 ocgcolorlinks

OCG colorlinks need objects and an entry in the catalog. Perhaps the objects need public names to avoid that ocg2 has to create duplicates? TODO

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

```

1227 \cs_new_protected:Npn \__hyp_ocg_init:
1228 {
1229     \pdf_object_new:n { __hyp/OCG/View }
1230     \pdf_object_new:n { __hyp/OCG/Print }
1231     \pdf_object_new:n { __hyp/OCG/config }
1232     \pdf_object_new:n { __hyp/OCG/refarray }
1233     \pdf_object_write:nne { __hyp/OCG/refarray } { array }
1234     {
1235         \pdf_object_ref:n { __hyp/OCG/View }
1236         \c_space_tl
1237         \pdf_object_ref:n { __hyp/OCG/Print }
1238     }
1239     \pdf_object_write:nnn { __hyp/OCG/View } { dict }
1240     {
1241         /Type/OCG
1242         /Name(View)
1243         /Usage
1244         <<
1245         /Print <</PrintState/OFF>>~
1246         /View <</ViewState/ON >>~
1247         >>
1248     }
1249     \pdf_object_write:nnn { __hyp/OCG/Print } { dict }
1250     {
1251         /Type/OCG
1252         /Name(Print)
1253         /Usage
1254         <<
1255         /Print <</PrintState/ON>>~

```

```

1256         /View <</ViewState/OFF>>~
1257     >>
1258 }
1259 \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG
1260 \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG
1261 \pdf_object_write:nne { __hyp/OCG/config } { dict }
1262 {
1263     /OFF[\pdf_object_ref:n { __hyp/OCG/Print }]
1264     /AS[
1265         <<
1266         /Event/View
1267         /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1268         /Category[/View]
1269         >>
1270         <<
1271         /Event/Print
1272         /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1273         /Category[/Print]
1274         >>
1275         <<
1276         /Event/Export
1277         /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1278         /Category[/Print]
1279         >>
1280     ]
1281 }
1282 \pdfmanagement_add:nne { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG
1283 \cs_gset:Npn \__hyp_ocg_init: {}
1284 }

```

(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.

```

1285 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1286 {
1287     \hook_gput_code:nnn
1288     {pdfannot/link/#2/begin}
1289     {hyp/ocg}
1290     {
1291         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1292         {
1293             \__hyp_ocg_init:
1294             \group_begin:
1295             \hbox_set:Nw \l__hyp_tmpa_box
1296         }
1297     }
1298     \hook_gput_code:nnn
1299     {pdfannot/link/#2/end}
1300     {hyp/ocg}
1301     {
1302         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1303         {
1304             \hbox_set_end:
1305             \mbox

```

```

1306         {
1307         \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1308         \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1309         \pdf_emc:
1310         \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1311         \group_begin:
1312         \color_select:n { hyp/color/#1 }
1313         \box_use_drop:N \l__hyp_tmpa_box
1314         \group_end:
1315         \pdf_emc:
1316         }
1317     \group_end:
1318 }
1319 }
1320 }

```

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

```

ocgcolorlink (setup key) 1321 \bool_lazy_or:nnTF
ocgcolorurl (setup key) 1322 { \pdf_version_compare_p:Nn > {1.4} }
ocgcolorfile (setup key) 1323 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
ocgcolormenu (setup key) 1324 {
ocgcolorrund (setup key) 1325 \keys_define:nn { hyp }
1326 {
1327     ,_ocgcolorlinks .meta:n =
1328     {
1329         ocgcolorlink=#1,
1330         ocgcolorurl=#1,
1331         ocgcolorfile=#1,
1332         ocgcolorrund=#1,
1333         ocgcolormenu=#1
1334     }
1335     ,_ocgcolorlinks .default:n = true
1336 }
1337 }
1338 {
1339 \keys_define:nn { hyp }
1340 {
1341     ,_ocgcolorlinks .code:n =
1342     {
1343         \msg_warning:nnee
1344         { hyp }
1345         { ignore-deprecated-or-unknown-option-in-pdf-version }
1346         { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1347     }
1348 }
1349 }
1350
1351 \keys_define:nn { hyp }
1352 {
1353     ,ocgcolorlinks .choice:
1354     ,ocgcolorlinks / true .meta:n =
1355     {
1356         pdfborder      ={{0~0~0}},
1357         pdfborderstyle ={},

```



```

1358     colorlinks      = false,
1359     _ocgcolorlinks = true
1360   }
1361   ,ocgcolorlinks / false .meta:n =
1362   {
1363     _ocgcolorlinks = false
1364   }
1365   ,ocgcolorlinks .default:n = {true}
1366 }
1367
1368 \seq_map_inline:Nn \c__hyp_annot_types_seq
1369 {
1370   \bool_lazy_or:nnTF
1371   { \pdf_version_compare_p:Nn > {1.4} }
1372   { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1373   {
1374     \keys_define:nn { hyp }
1375     {
1376       ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1377     }
1378   }
1379   {
1380     \keys_define:nn { hyp }
1381     {
1382       ,ocgcolor#1 .code:n=
1383       {
1384         \msg_warning:nnee
1385         { hyp }
1386         { ignore-deprecated-or-unknown-option-in-pdf-version }
1387         { ocgcolor#1 }
1388         { \pdf_version_major:.\pdf_version_minor: }
1389       }
1390     }
1391   }
1392 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1393 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1394 {
1395   \keys_define:nn { hyp }
1396   {
1397     ,#1highlight .choices:nn =
1398     { /I, /N, /O, /P}
1399     {
1400       \pdfannot_dict_put:nnn
1401       {link/#2}
1402       { H }
1403       { ##1 }
1404     }
1405   }
1406   ,#1highlight / .code:n =
1407   {

```

```

1408         \pdfannot_dict_remove:nn
1409             {link/#2}
1410             { H }
1411
1412     }
1413     ,#1highlight / unknown .code:n =
1414     {
1415         \msg_warning:nnee { hyp } { unknown-choice+empty }
1416         { #1highlight }
1417         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1418         { \exp_not:n {##1} }
1419     }
1420 }
1421 }
1422
1423
1424 \keys_define:nn { hyp }
1425 {
1426     ,pdfhighlight .choices:nn =
1427     { /I, /N, /O, /P}
1428     {
1429         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1430         {
1431             \pdfannot_dict_put:nnn
1432             {link/###2}
1433             { H }
1434             { #1 }
1435         }
1436     }
1437     ,pdfhighlight / .code:n =
1438     {
1439         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1440         {
1441             \pdfannot_dict_remove:nn
1442             {link/##2}
1443             { H }
1444         }
1445     }
1446     ,pdfhighlight .initial:n = {/I},
1447     ,pdfhighlight / unknown .code:n =
1448     {
1449         \msg_warning:nnee { hyp } { unknown-choice+empty }
1450         { pdfhighlight }
1451         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1452         { \exp_not:n {##1} }
1453     }
1454 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1455 \keys_define:nn { hyp }
hideurl (setup key)
hidefile (setup key)
hiderun (setup key)
hidemenu (setup key)

```

```

1456 {
1457   hidelinks .meta:n =
1458   {
1459     ,colorlinks      = false
1460     ,ocgcolorlinks  = false
1461     ,pdfborder       = { 0~0~0 }
1462     ,pdfborderstyle=
1463   }
1464 }
1465
1466 \seq_map_inline:Nn \c__hyp_annot_types_seq
1467 {
1468   \keys_define:nn { hyp }
1469   {
1470     hide#1 .meta:n =
1471     {
1472       ,color#1      = false
1473       ,ocgcolor#1  = false
1474       ,#1border     = { 0~0~0 }
1475       ,#1borderstyle =
1476     }
1477   }
1478 }

```

11.7 color schemes and settings

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

`colorscheme` (*setup key*)

```

1479 \keys_define:nn { hyp }
1480 {
1481   colorscheme .code:n =
1482   {
1483     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1484     {
1485       \keys_set:nn { hyp }
1486       {
1487         ##1 = ##2
1488       }
1489     }
1490   }
1491 }
1492 \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*)

```

1493 \keys_define:nn { hyp }
1494 {

```

```

1495     ,unicode      .code:n = {}
1496     ,pdfencoding .code:n = {}
1497     ,pdfversion  .code:n =
1498     {
1499         \msg_warning:nn { hyp }{ pdfversion-disabled }
1500     }
1501 }
1502 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
debug (setup key) 1503 \keys_define:nn { hyp }
draft (setup key) 1504 {
final (setup key) 1505     ,verbose .legacy_if_set:n = {Hy@verbose}
1506     ,debug .legacy_if_set:n = {Hy@verbose}
1507 }
1508 \keys_define:nn { hyp }
1509 {
1510     ,draft .code:n =
1511     {
1512         \Hy@drafttrue
1513         \PassOptionsToPackage{draft}{bookmark}
1514     }
1515     ,final .code:n =
1516     {
1517         \Hy@finaltrue
1518         \PassOptionsToPackage{final}{bookmark}
1519     }
1520 }

extension (setup key)
hypertextnames (setup key) 1521 \keys_define:nn { hyp }
naturalnames (setup key) 1522 {
pageanchor (setup key) 1523     ,extension .tl_set:N = \XR@ext
linktoc (setup key) 1524     ,extension .initial:n= pdf
linktocpage (setup key) 1525     ,hypertextnames .legacy_if_set:n = {Hy@hypertextnames}
plainpages (setup key) 1526     ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
localanchorname (setup key) 1527     ,localanchorname .legacy_if_set:n = {Hy@localanchorname}
linkfileprefix (setup key) 1528     ,naturalnames .legacy_if_set:n = {Hy@naturalnames}
1529     ,pageanchor .legacy_if_set:n = {Hy@pageanchor}
1530     ,plainpages .legacy_if_set:n = {Hy@plainpages}
1531 }
1532
1533 \keys_define:nn { hyp }
1534 {
1535     ,linktoc .choices:nn = { none, section, all, page }
1536     {
1537         \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1538     }
1539     ,linktoc / unknown .code:n =
1540     {

```

```

1541     \msg_warning:nneee { hyp } { unknown-choice }
1542     { linktoc }
1543     { none, section, all, page }
1544     { \exp_not:n {#1} }
1545   }
1546   ,linktocpage .choice:
1547   ,linktocpage / true .meta:n = {linktoc=page}
1548   ,linktocpage / false .meta:n = {linktoc=section}
1549   ,linktocpage .default:n = true
1550 }
1551

```

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

```

url (setup key) 1552 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1553 {
menu (setup key) 1554   \keys_define:nn { hyp }
run (setup key) 1555   {
1556     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1557   }
1558 }

```

```

1559 \keys_define:nn { hyp }
1560 {
1561   ,baseurl .code:n =
1562   {
1563     \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1564     \tl_if_empty:NTF \l__hyp_tmpa_tl
1565     {
1566       \pdfmanagement_remove:nn {Catalog} { URI }
1567     }
1568     {
1569       \pdfmanagement_add:nne {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1570     }
1571     \__hyp_store_metadata:nn {baseurl}{#1}
1572   }
1573   %only false does something ...
1574   ,bookmarks .choice:
1575   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1576   ,bookmarks / true .code:n = {}
1577   ,bookmarks .default:n = {true}
1578   ,bookmarksnumbered .legacy_if_set:n = {Hy@bookmarksnumbered}
1579   ,bookmarksopen .legacy_if_set:n = {Hy@bookmarksopen}
1580   ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1581   ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1582   ,pdfcenterwindow .choice:
1583   ,pdfcenterwindow / false .code:n =
1584   {
1585     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1586   }
1587   ,pdfcenterwindow / true .code:n =
1588   {
1589     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1590   }
1591   ,pdfcenterwindow / .code:n =

```

```

1592     {
1593     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1594     }
1595 ,pdfcenterwindow / unknown .code:n =
1596     {
1597     \msg_warning:nnee { hyp } { no-bool }
1598     { pdfcenterwindow }
1599     { \exp_not:n {#1} }
1600     }
1601 ,pdfcenterwindow .default:n = true
1602 ,pdfdirection .choice:
1603 ,pdfdirection / L2R .code:n =
1604     {
1605     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1606     }
1607 ,pdfdirection / R2L .code:n =
1608     {
1609     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1610     }
1611 ,pdfdirection / .code:n =
1612     {
1613     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1614     }
1615 ,pdfdirection / unknown .code:n =
1616     {
1617     \msg_warning:nnee { hyp } { unknown-choice+empty }
1618     { pdfdirection }
1619     { L2R , R2L }
1620     { \exp_not:n {#1} }
1621     }
1622 ,pdfdisplaydoctitle .choice:
1623 ,pdfdisplaydoctitle / false .code:n =
1624     {
1625     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1626     }
1627 ,pdfdisplaydoctitle / true .code:n =
1628     {
1629     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1630     }
1631 ,pdfdisplaydoctitle .default:n = true
1632 ,pdfduplex .choices:nn =
1633 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1634 {
1635 \pdf_version_compare:NnTF > {1.6}
1636 {
1637 \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1638 { PrintDuplex } { /#1 }
1639 }
1640 {
1641 \msg_warning:nnee
1642 {hyp}
1643 {ignore-deprecated-or-unknown-option-in-pdf-version}
1644 {pdfduplex}
1645 {\pdf_version:}

```

```

1646     }
1647   }%
1648   ,pdfduplex / .code:n =
1649   {
1650     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1651   }
1652   ,pdfduplex / unknown .code:n =
1653   {
1654     \msg_warning:nnee { hyp } { unknown-choice+empty }
1655     { pdfduplex }
1656     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1657     { \exp_not:n {#1} }
1658   }
1659   ,pdffitwindow .choice:
1660   ,pdffitwindow / false .code:n =
1661   {
1662     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1663   }
1664   ,pdffitwindow / true .code:n =
1665   {
1666     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1667   }
1668   ,pdffitwindow / .code:n =
1669   {
1670     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1671   }
1672   ,pdffitwindow .default:n = true
1673   ,pdffitwindow / unknown .code:n =
1674   {
1675     \msg_warning:nnee { hyp } { no-bool }
1676     { pdffitwindow }
1677     { \exp_not:n {#1} }
1678   }
1679   ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1680   ,pdflinkmargin .initial:n = {1pt}
1681   ,pdfmenubar .choice:
1682   ,pdfmenubar / true .code:n =
1683   {
1684     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1685   }
1686   ,pdfmenubar / false .code:n =
1687   {
1688     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1689     { HideMenubar } { true }
1690   }
1691   ,pdfmenubar / .code:n =
1692   {
1693     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1694   }
1695   ,pdfmenubar .default:n = true
1696   ,pdfmenubar / unknown .code:n =
1697   {
1698     \msg_warning:nnee { hyp } { no-bool }
1699     { pdfmenubar }

```

```

1700         { \exp_not:n {#1} }
1701     }
1702 ,pdfnewwindow .choice:
1703 ,pdfnewwindow / true .code:n =
1704     {
1705         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1706         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1707     }
1708 ,pdfnewwindow / false .code:n =
1709     {
1710         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1711         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1712     }
1713 ,pdfnewwindow / .code:n =
1714     {
1715         \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1716         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1717     }
1718 ,pdfnonfullscreenpagemode .choices:nn =
1719 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1720 {
1721     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1722     { NonFullScreenPageMode } {/#1}
1723 }
1724 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1725 {
1726     \pdf_version_compare:NnTF < {1.6}
1727     {
1728         %message
1729     }
1730     {
1731         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1732         {NonFullScreenPageMode}{/UseAttachments}
1733     }
1734 }
1735 ,pdfnonfullscreenpagemode / .code:n =
1736 {
1737     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1738 }
1739 ,pdfnonfullscreenpagemode / unknown .code:n =
1740 {
1741     \msg_warning:nnee { hyp } { unknown-choice+empty }
1742     { pdfnonfullscreenpagemode }
1743     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1744     { \exp_not:n {#1} }
1745 }
1746 ,pdfnumcopies .code:n =
1747 {
1748     \pdf_version_compare:NnTF > {1.6}
1749     {
1750         \tl_if_empty:nTF {#1}
1751         {
1752             \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1753         }

```



```

1754         {
1755             \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1756                 {NumCopies}{#1}
1757         }
1758     }
1759     {
1760         \msg_warning:nnee
1761             {hyp}
1762             {ignore-deprecated-or-unknown-option-in-pdf-version}
1763             {pdfnumcopies}
1764             {\pdf_version:}
1765     }
1766 }
1767 ,pdfpagelayout .choices:nn =
1768 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1769 { \pdfmanagement_add:nne {Catalog} { PageLayout }{ /#1 } }
1770 ,pdfpagelayout / .code:n =
1771 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1772 ,pdfpagelayout / unknown .code:n =
1773 {
1774     \msg_warning:nneee { hyp } { unknown-choice+empty }
1775     { pdfpagelayout }
1776     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1777     { \exp_not:n {#1} }
1778 }
1779 ,pdfpagemode .choices:nn =
1780 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1781 { \pdfmanagement_add:nne {Catalog} { PageMode }{ /#1 } }
1782 ,pdfpagemode / UseAttachments .code:n =
1783 {
1784     \pdf_version_compare:NnTF > {1.5}
1785     {
1786         \pdfmanagement_add:nne {Catalog} { PageMode }{ /UseAttachments }
1787     }
1788     {
1789         \msg_warning:nnee
1790             {hyp}
1791             {ignore-deprecated-or-unknown-value-in-pdf-version}
1792             {UseAttachments}
1793             {\pdf_version:}
1794     }
1795 }
1796 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1797 ,pdfpagemode / unknown .code:n =
1798 {
1799     \msg_warning:nneee { hyp } { unknown-choice+empty }
1800     { pdfpagemode }
1801     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1802     { \exp_not:n {#1} }
1803 }
1804 ,pdfpagescrop .code:n =
1805 {
1806     \tl_if_empty:nTF {#1} %or blank?
1807     {

```

```

1808         \pdfmanagement_remove:nn {Pages} { CropBox }
1809     }
1810     {
1811         \pdfmanagement_add:nne {Pages} { CropBox } { [#1] }
1812     }
1813 }
1814 ,pdfpicktraybypdfsize .choice:
1815 ,pdfpicktraybypdfsize / true .code:n =
1816 {
1817     \pdf_version_compare:NnTF > {1.6}
1818     {
1819         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1820         { PickTrayByPDFSize } { true }
1821     }
1822     {
1823         \msg_warning:nnee
1824         {hyp}
1825         {ignore-deprecated-or-unknown-option-in-pdf-version}
1826         {pdfpicktraybypdfsize}
1827         {\pdf_version:}
1828     }
1829 }
1830 ,pdfpicktraybypdfsize / false .code:n =
1831 {
1832     \pdf_version_compare:NnTF > {1.6}
1833     {
1834         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1835         { PickTrayByPDFSize } { false }
1836     }
1837     {
1838         \msg_warning:nnee
1839         {hyp}
1840         {ignore-deprecated-or-unknown-option-in-pdf-version}
1841         {pdfpicktraybypdfsize}
1842         {\pdf_version:}
1843     }
1844 }
1845 ,pdfpicktraybypdfsize / .code:n =
1846 {
1847     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1848 }
1849 ,pdfpicktraybypdfsize / unknown .code:n =
1850 {
1851     \msg_warning:nnee { hyp } { no-bool }
1852     { picktraybypdfsize }
1853     { \exp_not:n {#1} }
1854 }
1855 ,pdfprintarea .choices:nn =
1856 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1857 {
1858     \pdf_version_compare:NnTF < {2.0}
1859     {
1860         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1861         { PrintArea } { /#1 }

```

```

1862     }
1863     {
1864         \msg_warning:nnee
1865         {hyp}
1866         {ignore-deprecated-or-unknown-option-in-pdf-version}
1867         {pdfprintarea}
1868         {\pdf_version:}
1869     }
1870 }%
1871 ,pdfprintarea / .code:n =
1872 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1873 ,pdfprintarea / unknown .code:n =
1874 {
1875     \msg_warning:nnee { hyp } { unknown-choice+empty }
1876     { pdfprintarea }
1877     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1878     { \exp_not:n {#1} }
1879 }
1880 ,pdfprintclip .choices:nn =
1881 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1882 {
1883     \pdf_version_compare:NnTF < {2.0}
1884     {
1885         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1886         { PrintClip } { /#1 }
1887     }
1888     {
1889         \msg_warning:nnee
1890         {hyp}
1891         {ignore-deprecated-or-unknown-option-in-pdf-version}
1892         {pdfprintclip}
1893         {\pdf_version:}
1894     }
1895 }%
1896 ,pdfprintclip / .code:n =
1897 {
1898     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1899 }
1900 ,pdfprintclip / unknown .code:n =
1901 {
1902     \msg_warning:nnee
1903     { hyp }
1904     { unknown-choice+empty }
1905     { pdfprintclip }
1906     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1907     { \exp_not:n {#1} }
1908 }
1909 ,pdfprintpagerange .code:n =
1910 {
1911     \pdf_version_compare:NnTF > {1.6}
1912     {
1913         \tl_if_empty:nTF { #1}
1914         {
1915             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }

```

```

1916         { PrintPageRange }
1917     }
1918     {
1919         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1920         {PrintPageRange}{[#1]}
1921     }
1922 }
1923 {
1924     \msg_warning:nnee
1925     {hyp}
1926     {ignore-deprecated-or-unknown-option-in-pdf-version}
1927     {pdfprintpagerange}
1928     {\pdf_version:}
1929 }
1930 }
1931 ,pdfprintscaling .choices:nn =
1932 { None, AppDefault }
1933 {
1934     \pdf_version_compare:NnTF > {1.5}
1935     {
1936         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1937         { PrintScaling } { /#1 }
1938     }
1939     {
1940         \msg_warning:nnee
1941         {hyp}
1942         {ignore-deprecated-or-unknown-option-in-pdf-version}
1943         {pdfprintscaling}
1944         {\pdf_version:}
1945     }
1946 }%
1947 ,pdfprintscaling / .code:n =
1948 {
1949     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1950 }
1951 ,pdfprintscaling / unknown .code:n =
1952 {
1953     \msg_warning:nnee { hyp } { unknown-choice+empty }
1954     { pdfprintarea }
1955     { None, AppDefault }
1956     { \exp_not:n {#1} }
1957 }
1958 ,pdfremotestartview .code:n =
1959 {
1960     \tl_set:Ne \l__hyp_tmpa_tl {#1-null-null-null~}
1961     \exp_args:NNV
1962     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1963     {
1964         \tl_set:Ne \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1965     }
1966     {
1967         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1968         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1969     }

```

```

1970     }
1971 ,pdfremotestartview .initial:n = {Fit}
1972 % pdfstartpage is special as it shares code with pdfstartview
1973 ,pdfstartpage .code:n =
1974 {
1975     \tl_gset:Ne \g__hyp_dest_pdfstartpage_tl { #1 }
1976     \bool_if:nTF
1977     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1978     {
1979         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1980     }
1981     {
1982         \pdfmanagement_add:nne {Catalog} { OpenAction }
1983         {
1984             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1985         }
1986     }
1987 }
1988 ,pdfstartpage .initial:n =1
1989 ,pdfstartview .code:n =
1990 {
1991     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
1992     \exp_args:NNV
1993     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1994     {
1995         \tl_gset:Ne \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1996     }
1997     {
1998         \msg_warning:nmmm {hyp}{invalid-destination-value}{#1}{pdfstartview}
1999         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
2000     }
2001     \bool_if:nTF
2002     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
2003     {
2004         \pdfmanagement_remove:nn {Catalog} { OpenAction }
2005     }
2006     {
2007         \pdfmanagement_add:nne {Catalog} { OpenAction }
2008         {
2009             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
2010         }
2011     }
2012 }
2013 ,pdfstartview .initial:n = Fit
2014 ,pdftoolbar .choice:
2015 ,pdftoolbar / true .code:n =
2016 {
2017     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
2018 }
2019 ,pdftoolbar / false .code:n =
2020 {
2021     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
2022     { HideToolbar } { true }
2023 }

```

```

2024 ,pdftoolbar / true .code:n =
2025 {
2026   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
2027 }
2028 ,pdftoolbar .default:n = true
2029 ,pdftoolbar / unknown .code:n =
2030 {
2031   \msg_warning:nnee { hyp } { no-bool }
2032   { pdftoolbar }
2033   { \exp_not:n {#1} }
2034 }
2035 % pdfview see below.
2036 ,pdfviewarea .choices:nn =
2037 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2038 {
2039   \pdf_version_compare:NnTF < {2.0}
2040   {
2041     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2042     { ViewArea } { /#1 }
2043   }
2044   {
2045     \msg_warning:nnee
2046     {hyp}
2047     {ignore-deprecated-or-unknown-option-in-pdf-version}
2048     {pdfviewarea}
2049     {\pdf_version:}
2050   }
2051 }%
2052 ,pdfviewarea / .code:n =
2053 {
2054   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
2055 }
2056 ,pdfviewarea / unknown .code:n =
2057 {
2058   \msg_warning:nneee { hyp } { unknown-choice+empty }
2059   { pdfviewarea }
2060   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2061   { \exp_not:n {#1} }
2062 }
2063 ,pdfviewclip .choices:nn =
2064 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2065 {
2066   \pdf_version_compare:NnTF < {2.0}
2067   {
2068     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2069     { ViewClip } { /#1 }
2070   }
2071   {
2072     \msg_warning:nnee
2073     {hyp}
2074     {ignore-deprecated-or-unknown-option-in-pdf-version}
2075     {pdfviewclip}
2076     {\pdf_version:}
2077   }

```

```

2078     }%
2079 ,pdfviewclip / .code:n =
2080     {
2081         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
2082     }
2083 ,pdfviewclip / unknown .code:n =
2084     {
2085         \msg_warning:nneee { hyp } { unknown-choice+empty }
2086         { pdfviewclip }
2087         { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2088         { \exp_not:n {#1} }
2089     }
2090 ,pdfwindowui .choice:
2091 ,pdfwindowui / true .code:n =
2092     {
2093         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2094     }
2095 ,pdfwindowui / false .code:n =
2096     {
2097         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2098         { HideWindowUI } { true }
2099     }
2100 ,pdfwindowui / .code:n =
2101     {
2102         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2103     }
2104 ,pdfwindowui / unknown .code:n =
2105     {
2106         \msg_warning:nnee { hyp } { no-bool }
2107         { pdfwindowui }
2108         { \exp_not:n {#1} }
2109     }
2110 ,pdfwindowui .default:n = true
2111 }

```

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

```

2112 \keys_define:nn { hyp }
2113 {
2114     ,pdfview .code:n =
2115     {
2116         \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
2117         \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
2118         {
2119             { xyz }
2120             {
2121                 \int_compare:nNnTF { \seq_count:N \l__hyp_tmpa_seq } > { 1 }
2122                 {
2123                     \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2124                     \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2125                     {
2126                         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2127                     }
2128                     {
2129                         \tl_set:Ne \l__hyp_dest_pdfview_tl

```

```

2130         {
2131             \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2132         }
2133     }
2134 }
2135 {
2136     \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2137 }
2138 }
2139 { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2140 { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2141 { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2142 { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2143 { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2144 { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2145 { fitr }
2146 {
2147     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2148     {
2149         \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2150     }
2151     {
2152         %ensure 4 values ...
2153         \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2154         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2155         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2156         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2157         \hbox_set_to_wd:Nnn \l__hyp_dest_box
2158         {
2159             \fp_eval:n
2160             {
2161                 round
2162                 (
2163                     abs
2164                     (
2165                         \seq_item:Nn\l__hyp_tmpa_seq{4}
2166                         -
2167                         (\seq_item:Nn\l__hyp_tmpa_seq{2})
2168                     ),
2169                     3
2170                 )
2171             }bp
2172         }{}
2173         \box_set_dp:Nn \l__hyp_dest_box
2174         {
2175             \fp_eval:n
2176             {
2177                 round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2178             }bp
2179         }
2180         \box_set_ht:Nn \l__hyp_dest_box
2181         {
2182             \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2183         }

```



```

2184         }
2185     }
2186 }
2187 {
2188     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2189     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2190 }
2191 }
2192 ,pdfview .initial:n = {xyz}
2193 }

```

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.

```

2194 \keys_define:nn { hyp }
2195 {
2196     ,pdflang .code:n =
2197     {
2198         \tl_if_empty:nF { #1 }
2199         {
2200             \pdfmanagement_add:nne {Catalog} { Lang } { (#1) }
2201             \AddToDocumentProperties[document]{lang}{#1}
2202         }
2203     }
2204 }

```

12.3.1 “info keys”

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

```

2205 \regex_new:N\l__hyp_optlang_regex
2206 \regex_set:Nn\l__hyp_optlang_regex {\A\[[A-Za-z\-\-]+\]\(.*)}
2207 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2208 {
2209     \keys_define:nn { hyp }
2210     {
2211         pdf#1 .code:n =
2212         {
2213             \tl_if_blank:nTF {##1}
2214             {
2215                 \str_case:nnF { #1 }
2216                 {
2217                     {creator}
2218                     {
2219                         \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2220                         \pdfmanagement_add:nne {Info}{Creator}{()}
2221                     }
2222                     {producer}

```

```

2223         {
2224             \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2225             \pdfmanagement_add:nne {Info}{Producer}{()}
2226         }
2227     }
2228     {
2229         \pdfmanagement_remove:nn {Info}{#2}
2230     }
2231 }
2232 {
2233     \tl_set:Ne\l__hyp_tmpa_tl {\clist_item:nn{##1}{1}}
2234     \exp_args:NNV
2235     \regex_extract_once:NnN \l__hyp_optlang_regex \l__hyp_tmpa_tl\l__hyp_tmpa_s
2236     \seq_if_empty:NTF\l__hyp_tmpa_seq
2237     {
2238         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2239     }
2240     {
2241         \exp_args:Ne
2242         \__hyp_text_pdfstring_info:nN {\seq_item:Nn \l__hyp_tmpa_seq{3}}\l__hyp_t
2243     }
2244     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2245     {
2246         \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2247     }
2248 }
2249     \__hyp_store_metadata:nn {pdf#1}{##1}
2250 }
2251 }
2252 \keys_define:nn { hyp / info }
2253 {
2254     #2 .code:n =
2255     {
2256         \tl_if_blank:nTF {##1}
2257         {
2258             \pdfmanagement_remove:nn {Info}{#2}
2259         }
2260         {
2261             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2262             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2263             {
2264                 \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2265             }
2266         }
2267         \exp_args:Ne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2268     }
2269     ,unknown .code:n =
2270     {
2271         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2272         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2273         {
2274             \exp_args:Nno
2275             \pdfmanagement_add:nne {Info}
2276             { \l_keys_key_str } {\l__hyp_tmpa_str}

```

```

2277     }
2278   }
2279 }
2280 }
2281 \__hyp_setup_info_key:nn {author} {Author}
2282 \__hyp_setup_info_key:nn {title} {Title}
2283 \__hyp_setup_info_key:nn {producer} {Producer}
2284 \__hyp_setup_info_key:nn {creator} {Creator}
2285 % ignored key: addtopdfcreator
2286 \__hyp_setup_info_key:nn {subject} {Subject}
2287 \__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)
2288 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2289 {
2290   \keys_define:nn { hyp }
2291   {
2292     pdf#1 .code:n =
2293     {
2294       \tl_if_blank:nTF {##1}
2295       {
2296         \pdfmanagement_remove:nn {Info}{#2}
2297       }
2298       {
2299         \pdfmanagement_add:nne {Info}{#2}{(##1)}
2300       }
2301       \__hyp_store_metadata:nn {pdf#1}{##1}
2302       \AddToDocumentProperties[document]{#1}{##1}
2303     }
2304   }
2305   \keys_define:nn { hyp / info }
2306   {
2307     #2 .code:n =
2308     {
2309       \tl_if_blank:nTF {##1}
2310       {
2311         \pdfmanagement_remove:nn {Info}{#2}
2312       }
2313       {
2314         \pdfmanagement_add:nne {Info}{#2}{(##1)}
2315       }
2316       \exp_args:Ne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2317     }
2318   }
2319 }
2320
2321 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2322 \__hyp_setup_info_date_key:nn {moddate} {ModDate}
2323 \keys_define:nn { hyp }
2324 {
2325   pdfmetadate .code:n = { \__hyp_store_metadata:nn {pdfmetadate}{#1} }
2326 }

```

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

```

2327 \keys_define:nn { hyp }
2328 {
2329   ,pdftrapped .code:n =
2330   {
2331     \exp_args:Nne
2332     \keys_set:nn { hyp } { _pdftrapped = \str_uppercase:n { #1 } }
2333   }
2334   ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2335   {
2336     \pdfmanagement_add:nne {Info}{Trapped}
2337     {/
2338       \str_uppercase:f { \str_head:n { #1 } }
2339       \str_lowercase:f { \str_tail:n { #1 } }
2340     }
2341     \__hyp_store_metadata:ne {pdftrapped}
2342     {
2343       \str_uppercase:f { \str_head:n { #1 } }
2344       \str_lowercase:f { \str_tail:n { #1 } }
2345     }
2346   }
2347   ,_pdftrapped / unknown .code:n =
2348   {
2349     \msg_warning:nneee { hyp } { unknown-choice }
2350     { pdftrapped }
2351     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2352     { \exp_not:n {#1} }
2353   }
2354 }

```

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

```

2355 \keys_define:nn { hyp }
2356 {
2357   pdfinfo .code:n =
2358   {
2359     \keys_set:nn { hyp / info } { #1 }
2360   }
2361 }

```

Now we set some default values

```

2362 \keys_set:nn { hyp } {pdfcreator = LaTeX-with-hyperref}
2363 \keys_set:nn { hyp } {pdfauthor = }
2364 \keys_set:nn { hyp } {pdftitle = }
2365 \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:

```

2366 \clist_map_inline:nn
2367 {
2368   ,pdfcopyright
2369   ,pdftype

```

```

2370 ,pdflicenseurl
2371 ,pdfauthortitle
2372 ,pdfcaptionwriter
2373 ,pdfmetalang
2374 ,pdfsource
2375 ,pdfdocumentid
2376 ,pdfinstanceid
2377 ,pdfversionid
2378 ,pdfrendition
2379 ,pdfpublication
2380 ,pdfpubtype
2381 ,pdfbytes
2382 ,pdfnumpages
2383 ,pdfissn
2384 ,pdfeissn
2385 ,pdfisbn
2386 ,pdfbookedition
2387 ,pdfpublisher
2388 ,pdfvolumenum
2389 ,pdfissuenum
2390 ,pdfpagerange
2391 ,pdfdoi
2392 ,pdfurl
2393 ,pdfidentifier
2394 ,pdfsubtitle
2395 ,pdfpubstatus
2396 ,pdfcontactaddress
2397 ,pdfcontactcity
2398 ,pdfcontactregion
2399 ,pdfcontactpostcode
2400 ,pdfcontactcountry
2401 ,pdfcontactphone
2402 ,pdfcontactemail
2403 ,pdfcontacturl
2404 ,pdfdate
2405 }
2406 {
2407   \keys_define:nn { hyp }
2408   {
2409     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2410   }
2411 }
2412

```

12.5 Transitions

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

```

2413 \keys_define:nn { hyp }
2414 {
2415   pdfpageduration .code:n =
2416   {
2417     \tl_if_blank:nTF { #1 }
2418     {
2419       \pdfmanagement_remove:nn {Page}{Dur}

```

```

2420     }
2421     {
2422     \pdfmanagement_add:nnn {Page}{Dur}{#1}
2423     }
2424   }
2425 }

```

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

```

2426 \keys_define:nn { hyp }
2427 {
2428   pdfpagetransition .code:n =
2429   {
2430     \tl_if_blank:nTF {#1}
2431     {
2432       \pdfmanagement_remove:nn {Page}{Trans}
2433     }
2434     {
2435       \group_begin:
2436       \keys_set:nn { hyp / trans }{style=R,#1}
2437       \pdf_object_unnamed_write:ne { dict }
2438       {
2439         \pdfdict_use:n {l__hyp_page/Trans}
2440       }
2441       \pdfmanagement_add:mne {Page}{Trans}{\pdf_object_ref_last:}
2442       \group_end:
2443     }

```

```

2444     }
2445   }
2446   \keys_define:nn { hyp / trans }
2447   {
2448     ,style .choices:nn =
2449     { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2450     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2451     ,style / unknown .code:n =
2452     {
2453       \msg_warning:nneee { hyp } { unknown-choice }
2454       { trans / style }
2455       { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2456       { \exp_not:n {#1} }
2457     }
2458     ,duration .code:n =
2459     {
2460       \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{#1}
2461     }
2462     ,direction .choices:nn =
2463     { H,V}
2464     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2465     ,direction .choices:nn =
2466     { 0,90,180,270,315}
2467     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2468     ,direction / None .code:n =
2469     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2470     ,direction / unknown .code:n =
2471     {
2472       \msg_warning:nneee { hyp } { unknown-choice }
2473       { trans / direction }
2474       {
2475         H~(horizontal,~only~Split,~Blinds),
2476         V~(vertical,~only~Split,~Blinds),
2477         0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2478         90~(bottom~to~top,~only~Wipe),
2479         180~(right~to~left,~only~Wipe),
2480         270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2481         315~(top~left~to~bottom,~only~Glitter),
2482         None~(only~Fly)
2483       }
2484       { \exp_not:n {#1} }
2485     }
2486     ,motion .choices:nn =
2487     { I,0}
2488     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2489     ,motion / unknown .code:n =
2490     {
2491       \msg_warning:nneee { hyp } { unknown-choice }
2492       { trans / motion }
2493       { I~(inwards) , 0~(outwards) }
2494       { \exp_not:n {#1} }
2495     }
2496     ,scale .code:n =
2497     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }

```

```

2498 ,opaque .choices:nn = {true,false}
2499   { \pdfdict_put:nnn { l__hyp_page/Trans }{ B } { #1} }
2500 ,opaque / unknown .code:n =
2501   {
2502     \msg_warning:nneee { hyp } { unknown-choice }
2503     { trans / B }
2504     { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2505     { \exp_not:n {#1} }
2506   }
2507 % try to set unknown keys as style
2508 ,unknown .code:n =
2509   {
2510     % warning ...
2511     \exp_args:Nne\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2512   }
2513 }

```

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

```

2514 \keys_set_known:nv{ hyp }{opt@hyperref.sty}

```

Unfinished Form field code

```

2515 \NewDocumentCommand \MakeFieldObject { m m }
2516   {
2517     \pdfxform_new:nnn { #2 }{ } { #1 }
2518   }
2519
2520
2521 \prop_new:N \g__hyp_AcroForm_CoFields_prop
2522 \prop_new:N \g__hyp_AcroForm_Fields_prop
2523
2524 \let\HyField@afields\@empty
2525 \let\HyField@cofields\@empty
2526 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2527
2528 % the value doesn't matter, but with a prop we avoid duplicates and it is
2529 % clearly faster than removing them from a sequence
2530 \def\HyField@AuxAddToFields#1
2531   {
2532     \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2533   }%
2534
2535 %fields with empty key get a value too -- lets hope that
2536 %this give the expected behaviour
2537 \def\HyField@AuxAddToCoFields #1 #2
2538   {
2539     \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2540   }
2541
2542 \Hy@AtBeginDocument
2543   {
2544     \if@filesw
2545       \immediate\write\@mainaux{%
2546         \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2547       }%
2548     \immediate\write\@mainaux{%

```



```

2549     \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2550     }%
2551     \fi
2552     \let\HyField@AfterAuxOpen\@firstofone
2553     }%
2554
2555 \def\HyField@AddToFields
2556 {
2557     \exp_args:N\HyField__hypAddToFields
2558     {
2559         \pdfannot_box_ref_last:
2560     }
2561     \ifx\Fld@calculate@code\@empty
2562     \else
2563     \begingroup
2564         \Hy@safe@activestruer
2565         \edef\Hy@temp{%
2566             \endgroup
2567             \if@filesw
2568                 \write\@mainaux
2569                 {
2570                     \string\HyField@AuxAddToCoFields
2571                     {
2572                         \Fld@calculate@sortkey
2573                     }
2574                     {
2575                         \pdfannot_box_ref_last:
2576                     }
2577                 }
2578             \fi
2579         }%
2580     \Hy@temp
2581     \fi
2582     }%
2583
2584 \def\HyField__hypAddToFields#1{
2585     \HyField@AfterAuxOpen{%
2586         \if@filesw
2587             \write\@mainaux{%
2588                 \string\HyField@AuxAddToFields{#1}%
2589             }%
2590         \fi
2591     }%
2592     }%
2593
2594 \ExplSyntaxOff
2595 \ExplSyntaxOn
2596
2597 \def\@Form[#1]
2598 {
2599     \kvsetkeys{Form}{#1}
2600     \pdf@ifdraftmode{}
2601     {
2602         \Hy@FormObjects

```

```

2603     \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2604     {
2605         \pdfmanagement_add:nne { Catalog / AcroForm } { Fields }{##1}
2606         %\pdfmanagement_show:n { Catalog / AcroForm }
2607     }
2608     \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2609     {
2610         \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2611         {
2612             \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2613         }
2614         \seq_sort:Nn \l__hyp_tmpa_seq
2615         {
2616             \str_compare:nNnTF {##1} > {##2}
2617             { \sort_return_swapped: }
2618             { \sort_return_same: }
2619         }
2620         \seq_map_inline:Nn \l__hyp_tmpa_seq
2621         {
2622             \pdfmanagement_add:nne { Catalog / AcroForm }
2623             { CO }
2624             {
2625                 \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2626             }
2627         }
2628     }
2629     \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2630     {ZaDb} {\pdf_object_ref:n {__hyp/Font/ZaDb} }
2631     \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2632     {Helv} {\pdf_object_ref:n {__hyp/Font/Helv} }
2633     \pdfmanagement_add:nne {Catalog /AcroForm}
2634     {DA}{(/Helv~10~Tf~0~g)}
2635     \pdfmeta_standard_verify:nTF {form_no_NeedAppearances}
2636     {
2637         \legacy_if:nT { HyField@NeedAppearances }
2638         {
2639             \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2640         }
2641     }
2642     {
2643         \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2644     }
2645 }
2646 }
2647 \ExplSyntaxOff
2648 \let\@endForm\@empty
2649 \let\HyAnn@AbsPageLabel\@empty
2650 \let\Fld@pageobjref\@empty
2651
2652 \ExplSyntaxOn
2653 \newcount\HyAnn@Count
2654 \HyAnn@Count=\z@
2655 \def\HyAnn@AbsPageLabel
2656 {

```

```

2657 \global\advance\HyAnn@Count by\@ne
2658 %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2659 %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2660 %\zref@refused{HyAnn@\the\HyAnn@Count}%
2661 \__hyp_property_record:ee {HyAnn@\the\HyAnn@Count}{abspage}
2662 \property_ref_undefined_warn:ee {HyAnn@\the\HyAnn@Count}{abspage}
2663 }%
2664 \prg_generate_conditional_variant:Nnn \property_if_recorded:nm {ee} {T}
2665 \def\Fld@pageobjref
2666 {
2667   \property_if_recorded:eeT {HyAnn@\the\HyAnn@Count}{abspage}
2668   {
2669     /P~\pdf_pageobject_ref:e
2670     {
2671       \property_ref:ee{HyAnn@\the\HyAnn@Count}{abspage}
2672     }
2673   }
2674 }
2675 \ExplSyntaxOff
2676 \ExplSyntaxOn
2677 %% check if the attr should be set through
2678 %% hooks.
2679 %% check if options are missing.
2680 \def\@TextField[#1]#2{% parameters, label
2681   \def\Fld@name{#2}%
2682   \let\Fld@default\@empty
2683   \let\Fld@value\@empty
2684   \def\Fld@width{\DefaultWidthofText}%
2685   \def\Fld@height{%
2686     \ifFld@multiline
2687       \DefaultHeightofTextMultiline
2688     \else
2689       \DefaultHeightofText
2690     \fi
2691   }%
2692   \begingroup
2693     \expandafter\HyField@SetKeys\expandafter{%
2694       \DefaultOptionsofText,#1%
2695     }%
2696     \PDFForm@Name
2697     \HyField@FlagsText
2698     \ifFld@hidden\def\Fld@width{1sp}\fi
2699     \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2700     \LayoutTextField{#2}{%
2701       \leavevmode
2702       \HyAnn@AbsPageLabel
2703       \Hy@escapeform\PDFForm@Text
2704       \pdfannot_box:nnnn
2705         {\Fld@width}
2706         {\Fld@height}
2707         {Opt} %is this correct?
2708         {\PDFForm@Text}
2709       \MakeTextField{\Fld@width}{\Fld@height}
2710     \HyField@AddToFields

```

```

2711     }%
2712 \endgroup
2713 }
2714 \providecommand\@curropt{}
2715 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2716   \def\Fld@name{#2}
2717   \let\Fld@default\relax
2718   \let\Fld@value\relax
2719   \def\Fld@width{\DefaultWidthofChoiceMenu}
2720   \def\Fld@height{\DefaultHeightofChoiceMenu}
2721   \begingroup
2722     \Fld@menulength=0 %
2723     \@tempdima\z@
2724     \clist_map_variable:nNn { #3 } \@curropt
2725     %\@for\@curropt:=#3\do
2726     {%
2727       \expandafter\Fld@checkequals\@curropt==\%
2728       \Hy@StepCount\Fld@menulength
2729       \settowidth{\@tempdimb}{\@currDisplay}%
2730       \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2731     }%
2732     \advance\@tempdima by~15\p@
2733     \begingroup
2734     \HyField@SetKeys{#1}
2735     \edef\x{\endgroup
2736       \noexpand\expandafter
2737       \noexpand\HyField@SetKeys
2738       \noexpand\expandafter{%
2739         \expandafter\noexpand\cename DefaultOptionsof%
2740         \ifFld@radio
2741           Radio%
2742         \else
2743           \ifFld@combo
2744             \ifFld@popdown
2745               PopdownBox%
2746             \else
2747               ComboBox%
2748             \fi
2749           \else
2750             ListBox%
2751           \fi
2752         \fi
2753       \endcsname
2754     }%
2755     }\x
2756     \HyField@SetKeys{#1}%
2757     \PDFForm@Name
2758     \ifFld@hidden\def\Fld@width{1sp}\fi
2759     \ifx\Fld@value\relax
2760       \let\Fld@value\Fld@default
2761     \fi
2762     \LayoutChoiceField{#2}{%
2763       \ifFld@radio
2764         \HyField@FlagsRadioButton

```

```

2765     \__hypRadio{#3}%
2766 \else
2767 \begingroup
2768   \HyField@FlagsChoice
2769   \ifdim\Fld@width<\@tempdima
2770     \ifdim\@tempdima<1cm\@tempdima1cm\fi
2771     \edef\Fld@width{\the\@tempdima}%
2772   \fi
2773   \ifFld@combo
2774   \else
2775     \@tempdima=\the\Fld@menulength\Fld@charsize
2776     \advance\@tempdima by-\Fld@borderwidth bp %
2777     \advance\@tempdima by-\Fld@borderwidth bp %
2778     \edef\Fld@height{\the\@tempdima}%
2779   \fi
2780   \__hypListBox{#3}%
2781 \endgroup
2782 \fi
2783 }%
2784 \endgroup
2785 }
2786
2787 \def\__hypRadio#1{%
2788   \Fld@listcount=0-%
2789   %\show\Fld@default
2790   \EdefEscapeName\Fld@default{\Fld@default}%
2791   \clist_map_variable:nNn { #1 } \@curropt
2792   %\@for\@curropt:=#1\do
2793   {%
2794     \expandafter\Fld@checkequals\@curropt==\%
2795     \EdefEscapeName\@currValue{\@currValue}%
2796     \Hy@StepCount\Fld@listcount
2797     \@currDisplay\space
2798     \leavevmode
2799     \HyAnn@AbsPageLabel
2800     \Hy@escapeform\PDFForm@Radio
2801     \pdfxform_if_exist:nF { __hyp_xform_Ding }
2802     {
2803       \pdfxform_new:nnn { __hyp_xform_Ding } {}
2804       {
2805         \group_begin:
2806         \fontfamily{pzd}
2807         \fontencoding{U}
2808         \fontseries{m}
2809         \fontshape{n}
2810         \selectfont
2811         \char123
2812         \group_end:
2813       }
2814     }
2815     \pdfannot_box:nnne
2816     {\Fld@width}
2817     {\Fld@height}
2818     {Opt} %is this correct?

```

```

2819     {
2820         \PDFForm@Radio
2821         /AP
2822         <<
2823         /N
2824         <<
2825         /\currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2826         %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2827         >>
2828         >>
2829     }
2830     {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2831     \int_compare:nNnT { \Fld@listcount} = { 1 }
2832     { \HyField@AddToFields }
2833     \c_space_tl % deliberate space between radio buttons
2834     % to do: --> should be configurable
2835 }%
2836 }
2837
2838 \newcount\Fld@listcount
2839 \def\__hypListbox#1
2840 {
2841     \HyField@PDFChoices{#1}
2842     \mode_leave_vertical:
2843     \HyAnn@AbsPageLabel
2844     \Hy@escapeform\PDFForm@List
2845     \pdf_link_user:nnn
2846     {widget} %perhaps we need more types??
2847     {\PDFForm@List}
2848     {\MakeChoiceField{\Fld@width}{\Fld@height}}
2849     \HyField@AddToFields
2850 }
2851
2852
2853 \def\@PushButton[#1]#2{% parameters, label
2854 \def\Fld@name{#2}%
2855 \group_begin:
2856 \exp_args:No\HyField@SetKeys
2857 {
2858 \DefaultOptionsofPushButton,#1
2859 }
2860 \PDFForm@Name
2861 \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2862 {
2863 \HyField@FlagsPushButton
2864 \legacy_if:nT {Fld@hidden}
2865 {
2866 \def\Fld@width{1sp}
2867 }
2868 \LayoutPushButtonField
2869 {
2870 \mode_leave_vertical:
2871 \HyAnn@AbsPageLabel
2872 \Hy@escapeform\PDFForm@Push

```

```

2873         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2874     \pdfannot_box:nnnn
2875         {\box_wd:N\l_tmpa_box}
2876         {\box_ht:N\l_tmpa_box}
2877         {\box_dp:N\l_tmpa_box} %is this correct?
2878         {\PDFForm@Push}
2879         {\box_use:N\l_tmpa_box}
2880     \HyField@AddToFields
2881 }
2882 }
2883 {
2884     \msg_error:nn { hyp }{ pdfa-no-push-button }
2885     \LayoutPushButtonField
2886     {
2887         \mode_leave_vertical:
2888         \MakeButtonField{#2}
2889     }
2890 }
2891 \group_end:
2892 }
2893
2894 \def\@Submit[#1]#2
2895 {
2896     \def\Fld@width {\DefaultWidthofSubmit}
2897     \def\Fld@height{\DefaultHeightofSubmit}
2898     \group_begin:
2899         \exp_args:No\HyField@SetKeys
2900         {
2901             \DefaultOptionsofSubmit,#1
2902         }
2903     \HyField@FlagsPushButton
2904     \HyField@FlagsSubmit
2905     \legacy_if:nT { Fld@hidden }
2906     {
2907         \def\Fld@width{1sp}
2908     }
2909     \mode_leave_vertical:
2910     \HyAnn@AbsPageLabel
2911     \Hy@escapeform\PDFForm@Submit
2912     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2913     \pdfxform_if_exist:nF
2914     { __hyp_xform_Submit }
2915     {
2916         \pdfxform_new:nnn { __hyp_xform_Submit }{}
2917         {
2918             \fbox{\color_select:n{yellow}\textsf{Submit}}
2919         }
2920         \pdfxform_new:nnn { __hyp_xform_SubmitP }{}
2921         {
2922             \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2923         }
2924     }
2925     \pdfannot_box:nnnn
2926     {\box_wd:N\l_tmpa_box}

```

```

2927     {\box_ht:N\l_tmpa_box}
2928     {\box_dp:N\l_tmpa_box} %is this correct?
2929     {
2930         \PDFForm@Submit
2931         /AP<<
2932             /N~\pdfxform_ref:n {\_hyp_xform_Submit}~
2933             /D~\pdfxform_ref:n {\_hyp_xform_SubmitP}
2934         >>
2935     }
2936     \HyField@AddToFields
2937     \box_use:N\l_tmpa_box
2938
2939     \group_end:
2940 }
2941
2942 \def\@Reset[#1]#2
2943 {
2944     \def\Fld@width {\DefaultWidthofReset}
2945     \def\Fld@height{\DefaultHeightofReset}
2946     \group_begin:
2947     \exp_args:No\HyField@SetKeys
2948     {
2949         \DefaultOptionsofReset,#1
2950     }
2951     \mode_leave_vertical:
2952     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2953     {
2954         \HyField@FlagsPushButton
2955         \legacy_if:nT { Fld@hidden }
2956         { \def\Fld@width{1sp} }
2957         \HyAnn@AbsPageLabel
2958         \Hy@escapeform\PDFForm@Reset
2959         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2960         \pdfannot_box:nnnn
2961             {\box_wd:N\l_tmpa_box}
2962             {\box_ht:N\l_tmpa_box}
2963             {\box_dp:N\l_tmpa_box} %is this correct?
2964             { \PDFForm@Reset }
2965         \HyField@AddToFields
2966         \box_use:N \l_tmpa_box
2967     }
2968     {
2969         \msg_error:nn { hyp }{ pdfa-no-reset-button }
2970         \MakeButtonField{#2}
2971     }
2972     \group_end:
2973 }
2974
2975 \def\@CheckBox[#1]#2
2976 {% parameters, label
2977     \def\Fld@name{#2}
2978     \def\Fld@default{0}
2979     \group_begin:
2980     \def\Fld@width {\DefaultWidthofCheckBox}

```



```

2981 \def\Fld@height{\DefaultHeightofCheckBox}
2982 \exp_args:No\HyField@SetKeys
2983 {
2984   \DefaultOptionsofCheckBox,#1
2985 }
2986 \PDFForm@Name
2987 \HyField@FlagsCheckBox
2988 \legacy_if:nT { Fld@hidden }
2989 {
2990   \def\Fld@width{1sp}
2991 }
2992 \LayoutCheckField{#2}
2993 {
2994   \mode_leave_vertical:
2995   \HyAnn@AbsPageLabel
2996   \Hy@escapeform\PDFForm@Check
2997   \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
2998   {
2999     \pdfxform_new:nnn
3000     {__hyp_xform_CheckMarkYes}{-}
3001     {
3002       \group_begin:
3003       \fontfamily{pzd}
3004       \fontencoding{U}
3005       \fontseries{m}
3006       \fontshape{n}
3007       \selectfont
3008       \char51
3009       \group_end:
3010     }
3011     \pdfxform_new:nnn
3012     {__hyp_xform_CheckMarkOff}{-}
3013     {
3014       \group_begin:
3015       \fontfamily{pzd}
3016       \fontencoding{U}
3017       \fontseries{m}
3018       \fontshape{n}
3019       \selectfont
3020       \phantom{\char51} %perhaps xetex needs some small glyph ..
3021       \group_end:
3022     }
3023   }
3024   \pdfannot_box:nnnn
3025   {\Fld@width}
3026   {\Fld@height}
3027   {Opt} %is this correct?
3028   {\PDFForm@Check}
3029   \HyField@AddToFields %check if this works with xelatex ...
3030 }
3031 \group_end:
3032 }
3033 \ExplSyntaxOff
3034

```

```

3035 %hm. Should a luatex driver use type1 fonts in fields????
3036 \ExplSyntaxOn
3037 \def\Hy@FormObjects
3038 {
3039   \pdf_object_new:n   {__hyp/Encoding/pdfdoc }
3040   \pdf_object_new:n   {__hyp/Font/ZaDb }
3041   \pdf_object_new:n   {__hyp/Font/Helv }
3042   \pdf_object_write:nne {__hyp/Encoding/pdfdoc } { dict }
3043   {
3044     /Type/Encoding
3045     /Differences[
3046       24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
3047       /ring/tilde
3048       \c_space_tl
3049       39/quotesingle
3050       \c_space_tl
3051       96/grave %
3052       \iow_newline:
3053       128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
3054       /fraction/guilsinglleft/guilsinglright/minus/perthousand
3055       /quotedblbase/quotedblleft/quotedblright/quoteleft
3056       /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
3057       /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
3058       \iow_newline:
3059       164/currency
3060       \c_space_tl
3061       166/brokenbar
3062       \c_space_tl
3063       168/dieresis/copyright/ordfeminine
3064       \c_space_tl
3065       172/logicalnot/.notdef/registered/macron/degree/plusminus
3066       /twosuperior/threesuperior/acute/mu
3067       \c_space_tl
3068       183/periodcentered/cedilla/onesuperior/ordmasculine
3069       \c_space_tl
3070       188/onequarter/onehalf/threequarters
3071       \iow_newline:
3072       192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
3073       /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
3074       /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
3075       /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
3076       /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
3077       /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
3078       /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
3079       /iacute/icircumflex/idieresis/eth/ntilde/ograve/oacute
3080       /ocircumflex/otilde/odieresis/divide/oslash/ugrave
3081       /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
3082     ]
3083   }
3084   \pdf_object_write:nnn {__hyp/Font/ZaDb } { dict }
3085   {
3086     /Type/Font
3087     /Subtype/Type1
3088     /Name/ZaDb

```

```

3089         /BaseFont/ZapfDingbats
3090     }
3091     \pdf_object_write:nne {__hyp/Font/Helv } { dict }
3092     {
3093         /Type/Font
3094         /Subtype/Type1
3095         /Name/Helv
3096         /BaseFont/Helvetica
3097         /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
3098     }
3099     \global\let\Hy@FormObjects\relax
3100 }
3101 \ExplSyntaxOff
3102 \providecommand*{\Fld@pageobjref}{ }
3103 \ifcsname pdf@escapestring\endcsname
3104   \def\Hy@escapeform#1{%
3105     \ifHy@pdfescapeform
3106       \let\Hy@escapestring\pdfescapestring
3107     \else
3108       \let\Hy@escapestring\@firstofone
3109     \fi
3110   }%
3111   \Hy@escapeform{ }%
3112 \else
3113   \let\Hy@escapestring\@firstofone
3114   \def\Hy@escapeform#1{%
3115     \ifHy@pdfescapeform
3116       \def\Hy@escapestring##1{%
3117         \noexpand\Hy@escapestring{\noexpand##1}%
3118       }%
3119       \edef\Hy@temp{#1}%
3120       \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{ }\@nil
3121     \def\Hy@escapestring##1{%
3122       \@ifundefined{Hy@esc@\string##1}{%
3123         ##1%
3124         \ThisShouldNotHappen
3125       }{%
3126         \csname Hy@esc@\string##1\endcsname
3127       }%
3128     }%
3129   \else
3130     \let\Hy@escapestring\@firstofone
3131   \fi
3132 }%
3133 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3134   \ifx\#3\%
3135     \else
3136       \expandafter
3137       \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3138       \Hy@ReturnAfterFi{%
3139         \Hy__hypescapeform#3\@nil
3140       }%
3141     \fi
3142   }%

```

```

3143 \fi
3144 \def\PDFForm@Name{%
3145   \PDFForm__hypName\Fld@name
3146   \ifx\Fld@altname\relax
3147   \else
3148     \PDFForm__hypName\Fld@altname
3149   \fi
3150   \ifx\Fld@mappingname\relax
3151   \else
3152     \PDFForm__hypName\Fld@mappingname
3153   \fi
3154 }
3155 \def\PDFForm__hypName#1{%
3156   \begingroup
3157     \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3158     \ifHy@unicode
3159       \Hy@unicodedefalse
3160     \fi
3161     \fi
3162     \pdfstringdef\Hy@gtemp#1%
3163   \endgroup
3164   \let#1\Hy@gtemp
3165 }
3166 \def\Fld@X@additionalactions{%
3167   \ifx\Fld@keystroke@code\@empty
3168   \else
3169     /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3170   \fi
3171   \ifx\Fld@format@code\@empty
3172   \else
3173     /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3174   \fi
3175   \ifx\Fld@validate@code\@empty
3176   \else
3177     /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3178   \fi
3179   \ifx\Fld@calculate@code\@empty
3180   \else
3181     /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3182   \fi
3183   \ifx\Fld@onfocus@code\@empty
3184   \else
3185     /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3186   \fi
3187   \ifx\Fld@onblur@code\@empty
3188   \else
3189     /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3190   \fi
3191   \ifx\Fld@onmousedown@code\@empty
3192   \else
3193     /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3194   \fi
3195   \ifx\Fld@onmouseup@code\@empty
3196   \else

```

```

3197     /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3198 \fi
3199 \ifx\Fld@onenter@code\@empty
3200 \else
3201     /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3202 \fi
3203 \ifx\Fld@onexit@code\@empty
3204 \else
3205     /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3206 \fi
3207 }
3208 \ExplSyntaxOn
3209 \def\Fld@additionalactions
3210 {%
3211     \exp_args:N\str_if_eq:nnF {\Fld@X@additionalactions}{}
3212     {
3213         \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3214         {/AA<<\Fld@X@additionalactions>>}
3215     }
3216 }
3217 \ExplSyntaxOff
3218 \def\Fld@annotnames{%
3219     /T(\Fld@name)%
3220     \ifx\Fld@altname\relax
3221     \else
3222         /TU(\Fld@altname)%
3223     \fi
3224     \ifx\Fld@mappingname\relax
3225     \else
3226         /TM(\Fld@mappingname)%
3227     \fi
3228 }
3229 \ExplSyntaxOn
3230 \def\PDFForm@Check
3231 {
3232     /Subtype/Widget
3233     ~\Fld@annotflags
3234     ~\Fld@pageobjref
3235     ~\Fld@annotnames
3236     /FT/Btn
3237     \Fld@flags
3238     /Q~\Fld@align
3239     /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3240     /AP
3241     <<
3242         /N
3243         <<
3244             /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3245             /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3246         >>
3247     >>
3248     /MK<<
3249     \int_compare:nNnF {\Fld@rotation}={0}
3250     {

```

```

3251         /R~\Fld@rotation
3252     }
3253     \tl_if_empty:NF\Fld@bordercolor
3254     {
3255         /BC[\Fld@bordercolor]
3256     }
3257     \tl_if_empty:NF\Fld@bcolor
3258     {
3259         /BG[\Fld@bcolor]
3260     }
3261     /CA(\Hy@escapestring{\Fld@cbsymbol})%
3262 >>
3263 /DA
3264 (
3265     /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3266     \tl_if_empty:NF \Fld@color
3267     {
3268         \c_space_tl \Fld@color
3269     }
3270 )
3271 /H/P
3272 \legacy_if:nTF {Fld@checked}
3273 {
3274     /V/Yes /AS/Yes
3275 }
3276 {
3277     /V/Off /AS/Off
3278 }
3279 \Fld@additionalactions
3280 }
3281 \ExplSyntaxOff
3282 \ExplSyntaxOn
3283 \def\PDFForm@Push
3284 {
3285     /Subtype/Widget
3286     ~\Fld@annotflags
3287     ~\Fld@pageobjref
3288     ~\Fld@annotnames
3289     /FT/Btn
3290     ~\Fld@flags
3291     /H/P
3292     /BS<</W~\Fld@borderwidth/S~\Fld@borderstyle>>
3293     \bool_if:nT
3294     {
3295         !\int_compare_p:nNn {\Fld@rotation} = {0}
3296         ||
3297         \tl_if_exist_p:N \Fld@bordercolor
3298     }
3299     {
3300         /MK
3301         <<
3302         \int_compare:nNnF {\Fld@rotation} = {0}
3303         {
3304             /R~\Fld@rotation

```

```

3305         }
3306         \tl_if_exist:NT \Fld@bordercolor
3307         {
3308             /BC[\Fld@bordercolor]
3309         }
3310     >>
3311     }
3312     /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3313     \Fld@additionalactions
3314 }
3315
3316 \ExplSyntaxOff
3317 \def\PDFForm@List{%
3318     /Subtype/Widget%
3319     \Fld@annotflags
3320     \Fld@pageobjref
3321     \Fld@annotnames
3322     /FT/Ch%
3323     \Fld@flags
3324     /Q \Fld@align
3325     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3326     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3327         \ifx\Fld@bordercolor\relax\else 1\fi
3328         \ifx\Fld@bcolor\relax \else 1\fi
3329         \space
3330     \else
3331         /MK<<%
3332         \ifnum\Fld@rotation=\z@
3333         \else
3334             /R \Fld@rotation
3335         \fi
3336         \ifx\Fld@bordercolor\relax
3337         \else
3338             /BC[\Fld@bordercolor]%
3339         \fi
3340         \ifx\Fld@bcolor\relax
3341         \else
3342             /BG[\Fld@bcolor]%
3343         \fi
3344         >>%
3345     \fi
3346     /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3347         \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3348     \Fld@choices
3349     \Fld@additionalactions
3350 }
3351 \ExplSyntaxOn
3352 \def\PDFForm@Radio
3353 {
3354     /Subtype/Widget
3355     ~\Fld@annotflags
3356     ~\Fld@pageobjref
3357     ~\Fld@annotnames
3358     /FT/Btn

```

```

3359 \Fld@flags
3360 /H/P
3361 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3362 /MK<<
3363 \ifnum\Fld@rotation=\z@
3364 \else
3365 /R~\Fld@rotation
3366 \fi
3367 \ifx\Fld@bordercolor\relax
3368 \else
3369 /BC[\Fld@bordercolor]%
3370 \fi
3371 \ifx\Fld@bcolor\relax
3372 \else
3373 /BG[\Fld@bcolor]%
3374 \fi
3375 /CA(\Hy@escapestring{\Fld@radiosymbol})%
3376 >>
3377 /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3378 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3379 \ifx\Fld@default\@empty
3380 /V/Off%
3381 /DV/Off%
3382 \else
3383 /V/\Fld@default
3384 /DV/\Fld@default
3385 \fi
3386 \Fld@additionalactions
3387 }
3388 \ExplSyntaxOff
3389 \ExplSyntaxOn
3390 % Does an appearance dict make sense here?
3391 \def\PDFForm@Text
3392 {
3393 /Subtype/Widget
3394 ~\Fld@annotflags
3395 ~\Fld@pageobjref
3396 ~\Fld@annotnames
3397 /FT/Tx
3398 ~\Fld@flags
3399 /Q~\Fld@align
3400 /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3401 \bool_if:nT
3402 {
3403 !\int_compare_p:nNn {\Fld@rotation} = {0}
3404 ||
3405 \tl_if_exist_p:N \Fld@bordercolor
3406 ||
3407 \tl_if_exist_p:N \Fld@bcolor
3408 }
3409 {
3410 /MK
3411 <<
3412 \int_compare:nNnF {\Fld@rotation} = {0}

```



```

3413         {
3414             /R~\Fld@rotation
3415         }
3416         \tl_if_exist:NT \Fld@bordercolor
3417         {
3418             /BC[\Fld@bordercolor]
3419         }
3420         \tl_if_exist:NT \Fld@bcolor
3421         {
3422             /BG[\Fld@bcolor]
3423         }
3424     >>
3425 }
3426 /DA
3427 (
3428     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3429     \tl_if_empty:NF {\c_space_tl\Fld@color}
3430 )
3431 /DV(\Hy@escapestring{\Fld@default})
3432 /V(\Hy@escapestring{\Fld@value})
3433 ~\Fld@additionalactions
3434 \int_compare:nNnT { \Fld@maxlen}>{0}
3435 {
3436     /MaxLen~\Fld@maxlen
3437 }
3438 }
3439 \ExplSyntaxOff
3440
3441 \def\PDFForm@Submit{%
3442     /Subtype/Widget%
3443     \Fld@annotflags
3444     \Fld@pageobjref
3445     \Fld@annotnames
3446     /FT/Btn%
3447     \Fld@flags
3448     /H/P%
3449     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3450     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3451         \ifx\Fld@bordercolor\relax\else 1\fi
3452         \space
3453     \else
3454         /MK<<%
3455         \ifnum\Fld@rotation=\z@
3456         \else
3457             /R \Fld@rotation
3458         \fi
3459         \ifx\Fld@bordercolor\relax
3460         \else
3461             /BC[\Fld@bordercolor]%
3462         \fi
3463     >>%
3464     \fi
3465     /A<<%
3466     /S/SubmitForm%

```

```

3467 /F<<%
3468 /FS/URL%
3469 /F(\Hy@escapestring{\Form@action})%
3470 >>%
3471 \Fld@submitflags
3472 >>%
3473 \Fld@additionalactions
3474 }
3475 \ExplSyntaxOn
3476 \def\PDFForm@Reset{%
3477 /Subtype/Widget%
3478 \Fld@annotflags
3479 \Fld@pageobjref
3480 \Fld@annotnames
3481 /FT/Btn%
3482 \Fld@flags
3483 /H/P%
3484 /DA(/Helv-\strip@pt\Fld@charsize\space Tf-0-0~1-rg)%
3485 \ifcase0\ifnum\Fld@rotation=z@ \else 1\fi
3486 \ifx\Fld@bordercolor\relax\else 1\fi
3487 \space
3488 \else
3489 /MK<<%
3490 \ifnum\Fld@rotation=z@
3491 \else
3492 /R~\Fld@rotation
3493 \fi
3494 \ifx\Fld@bordercolor\relax
3495 \else
3496 /BC[\Fld@bordercolor]%
3497 \fi
3498 >>%
3499 \fi
3500 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3501 /A<</S/ResetForm>>%
3502 \Fld@additionalactions
3503 }%
3504
3505
3506 %these patterns are used in hyperref checks.
3507 %it is unclear if they are really useful and if a backend support is
3508 %needed.
3509 \str_case:VnF \c_sys_backend_str
3510 {
3511 { pdfmode }
3512 {
3513 \def\HyPat@ObjRef
3514 {
3515 [0-9]*[1-9][0-9]*~0~R
3516 }
3517 }
3518 { dvipdfmx }
3519 {
3520 \def\HyPat@ObjRef

```

```

3521     {
3522         @[\~]+
3523     }
3524 }
3525 { xdvipdfmx }
3526 {
3527     \def\HyPat@ObjRef
3528     {
3529         @[\~]+
3530     }
3531 }
3532 }
3533 { %also set in hyperref sty, so probably not needed.
3534     \def\HyPat@ObjRef/{.+}
3535 }
3536
3537
3538 \ExplSyntaxOff
3539 % UF: removed Hy@writebookmark
3540 %     \Hy@currentbookmarklevel{0}
3541 %     \Hy@numberline
3542 %     \__hypwritetorep
3543 %     counter{bookmark@seq@number}
3544 % removed \HyPsd@SanitizeForOutFile, not needed
3545 % removed \currentpdfbookmark, defined by bookmark,
3546 % should use \newcommand there
3547 % removed \subpdfbookmark, defined by bookmark,
3548 % should use \newcommand there
3549 % removed \belowpdfbookmark, defined by bookmark,
3550 % should use \newcommand there
3551 % removed \pdfbookmark, defined by bookmark,
3552 % \BOOKMARK
3553 % \@BOOKMARK
3554 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3555 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3556 %% removed \ReadBookmarks / unneeded with bookmark.
3557 %% removed \Hy@OutlineName
3558 %% removed \check@bm@number
3559 %% removed \calc@bm@number
3560
3561 \ifHy@implicit
3562 \else
3563     \expandafter\endinput
3564 \fi
3565 \newlength\Hy@SectionHShift
3566 \def\Hy@SectionAnchorHref#1{%
3567     \ifx\protect\@typeset@protect
3568         \Hy__hypSectionAnchor{#1}%
3569     \fi
3570 }
3571 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%
3572     \leavevmode
3573     \hbox to 0pt{%
3574         \kern-\Hy@SectionHShift

```

```

3575     \Hy@raisedlink{%
3576       \hyper@anchorstart{#1}\hyper@anchorend
3577     }%
3578     \hss
3579   }%
3580 }
3581 \@ifundefined{hyper@nopatch@sectioning}
3582 {
3583   \let\H@old@ssect\@ssect
3584   \def\@ssect#1#2#3#4#5{%
3585     \Hy@MakeCurrentHrefAuto{section*}%
3586     \setlength{\Hy@SectionHShift}{#1}%
3587     \begingroup
3588       \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3589       \toks\tw@\expandafter{%
3590         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3591         #5%
3592       }%
3593     \edef\x{\endgroup
3594       \the\toks@{\the\toks\tw@}}%
3595     }\x
3596   }
3597   \let\H@old@schapter\@schapter
3598   \def\@schapter#1{%
3599     \begingroup
3600       \let\@mkboth\@gobbletwo
3601       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3602       \Hy@raisedlink{%
3603         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3604       }%
3605     \endgroup
3606     \H@old@schapter{#1}%
3607   }
3608   \@ifundefined{@chapter}{-}{%
3609     \let\Hy@org@chapter\@chapter
3610     \def\@chapter{%
3611       \def\Hy@next{%
3612         \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3613         \Hy@raisedlink{%
3614           \hyper@anchorstart{\@currentHref}\hyper@anchorend
3615         }%
3616       }%
3617       \ifnum\c@secnumdepth>\m@ne
3618         \@ifundefined{if@mainmatter}%
3619         \iftrue{\csname if@mainmatter\endcsname}%
3620         \let\Hy@next\relax
3621       \fi
3622     \fi
3623     \Hy@next
3624     \Hy@org@chapter
3625   }%
3626 }
3627 \let\H@old@part\@part
3628 \begingroup\expandafter\expandafter\expandafter\endgroup

```

```

3629 \expandafter\ifx\csname chapter\endcsname\relax
3630 \let\Hy@secnum@part\z@
3631 \else
3632 \let\Hy@secnum@part\m@ne
3633 \fi
3634 \def\@part{%
3635 \ifnum\Hy@secnum@part>\c@secnumdepth
3636 \phantomsection
3637 \fi
3638 \H@old@part
3639 }
3640 \let\H@old@spart\@spart
3641 \def\@spart#1{%
3642 \Hy@MakeCurrentHrefAuto{part*}%
3643 \Hy@raisedlink{%
3644 \hyper@anchorstart{\@currentHref}\hyper@anchorend
3645 }%
3646 \H@old@spart{#1}%
3647 }
3648 \let\H@old@sect\@sect
3649 \def\@sect#1#2#3#4#5#6[#7]#8{%
3650 \ifnum #2>\c@secnumdepth
3651 \expandafter\@firstoftwo
3652 \else
3653 \expandafter\@secondoftwo
3654 \fi
3655 {%
3656 \Hy@MakeCurrentHrefAuto{section*}%
3657 \setlength{\Hy@SectionHShift}{#3}%
3658 \begingroup
3659 \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3660 \toks\tw@\expandafter{%
3661 \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3662 #8%
3663 }%
3664 \edef\x{\endgroup
3665 \the\toks@\the\toks\tw@}%
3666 }\x
3667 }{%
3668 \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3669 }%
3670 }
3671 }{}
3672 \expandafter\def\csname Parent-4\endcsname{}
3673 \expandafter\def\csname Parent-3\endcsname{}
3674 \expandafter\def\csname Parent-2\endcsname{}
3675 \expandafter\def\csname Parent-1\endcsname{}
3676 \expandafter\def\csname Parent0\endcsname{}
3677 \expandafter\def\csname Parent1\endcsname{}
3678 \expandafter\def\csname Parent2\endcsname{}
3679 \expandafter\def\csname Parent3\endcsname{}
3680 \expandafter\def\csname Parent4\endcsname{}
3681 %%
3682 %% End of file 'hgeneric-testphase.def'.

```

```

3683 </package>
3684 <*colorscheme>
3685 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3686 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3687 % linkcolor=
3688 %,filecolor=
3689 %,urlcolor=
3690 %,menucolor=
3691 %,runcolor=
3692 %,linkbordercolor=
3693 %,filebordercolor=
3694 %,urlbordercolor=
3695 %,menubordercolor=
3696 %,runbordercolor=
3697
3698 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }
3699 {
3700   linkcolor      = [rgb]{1,0,0}, %red
3701   filecolor      = [rgb]{0,1,1}, %cyan
3702   urlcolor       = [rgb]{1,0,1}, %magenta
3703   menucolor      = [rgb]{1, 0, 0}, %red
3704   runcolor       = [rgb]{0,1,1}, %cyan
3705 %-----
3706   linkbordercolor = [rgb]{1, 0 ,0 },
3707   filebordercolor = [rgb]{0, .5, .5},
3708   urlbordercolor  = [rgb]{0, 1, 1},
3709   menubordercolor = [rgb]{1, 0, 0},
3710   runbordercolor  = [rgb]{0, .7, .7}
3711 }
3712
3713 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3714 {
3715   linkcolor      = [rgb]{0,0.2,0.6},
3716   filecolor      = [rgb]{0.8,0,0.8},
3717   urlcolor       = [rgb]{0.8,0,0.8},
3718   menucolor      = [rgb]{0,0.2,0.6},
3719   runcolor       = [rgb]{0.8,0,0.8},
3720 %----- %-----
3721   linkbordercolor = [rgb]{0,0.2,0.6},
3722   filebordercolor = [rgb]{0.8,0,0.8},
3723   urlbordercolor  = [rgb]{0.8,0,0.8},
3724   menubordercolor = [rgb]{0,0.2,0.6},
3725   runbordercolor  = [rgb]{0.8,0,0.8}
3726 }
3727
3728 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3729 { %two colors: intern/extern
3730   linkcolor      = [rgb]{0.79216, 0, 0.12549},
3731   filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3732   urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3733   menucolor      = [rgb]{0.79216, 0, 0.12549 },
3734   runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3735 %----- %-----
3736   linkbordercolor = [rgb]{0.79216, 0, 0.12549},

```

```

3737   filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3738   urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3739   menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3740   runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3741 }
3742
3743 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3744 { %all darkgray
3745   linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3746   filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3747   urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3748   menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3749   runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3750   %----- %-----
3751   linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3752   filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3753   urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3754   menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3755   runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3756 }
3757
3758 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3759 { %dvipsnam.def
3760   linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3761   filecolor      = [rgb]{1, 0, 0}, %Red
3762   urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3763   menucolor      = [rgb]{1, 0, 0}, %Red
3764   runcolor       = [rgb]{1, 0, 0}, %Red
3765   %----- %-----
3766   linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3767   filebordercolor = [rgb]{1, 0, 0}, %Red
3768   urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3769   menubordercolor = [rgb]{1, 0, 0}, %Red
3770   runbordercolor  = [rgb]{1, 0, 0} %Red
3771 }
3772
3773 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3774 { %dvipsnam.def
3775   linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3776   filecolor      = [rgb]{0, 1, 0}, %Green
3777   urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3778   menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3779   runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3780   %----- %-----
3781   linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3782   filebordercolor = [rgb]{0, 1, 0}, %Green
3783   urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3784   menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3785   runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3786 }
3787
3788
3789 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3790 {

```

```

3791 linkcolor      = [rgb]{0.50196, 0, 0.02353},
3792 filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3793 urlcolor       = [rgb]{0.54118, 0, 0.52941},
3794 menucolor      = [rgb]{0.44706, 0.45882, 0},
3795 runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3796 %-----      %-----
3797 linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3798 filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3799 urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3800 menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3801 runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3802 }
3803
3804 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3805 {
3806 linkcolor      = [rgb]{0,0,0},
3807 filecolor      = [rgb]{0,0,0},
3808 urlcolor       = [rgb]{0,0,0},
3809 menucolor      = [rgb]{0,0,0},
3810 runcolor       = [rgb]{0,0,0},
3811 %-----      %-----
3812 linkbordercolor = [rgb]{0,0,0},
3813 filebordercolor = [rgb]{0,0,0},
3814 urlbordercolor  = [rgb]{0,0,0},
3815 menubordercolor = [rgb]{0,0,0},
3816 runbordercolor  = [rgb]{0,0,0}
3817 }
3818 </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, 789
<code>\\$</code>	279
<code>\%</code>	790
<code>\-</code>	2206
<code>\.</code>	550, 554, 556
@curropt commands:	
<code>\@curropt:</code>	2725, 2792
<code>\[</code>	2206
<code>\]</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, 939, 2727, 2794, 3134
<code>_</code>	548, 550, 554, 556
<code>\]</code>	2206
A	
<code>\A</code>	548, 2206
<code>\Acrobatmenu</code>	18, 174
<code>\addcontentsline</code>	13
<code>\AddToDocumentProperties</code> ..	389, 2201, 2302
<code>\AddToHook</code>	427, 444, 450
<code>\AddToHookNext</code>	199
<code>\advance</code>	2657, 2732, 2776, 2777
<code>allcolors</code> (hypersetup key) ..	<u>1067</u>
<code>\author</code>	2
B	
<code>\b</code>	552
<code>\begingroup</code>	210, 278, 376, 2563, 2692, 2721, 2733, 2767, 3156, 3587, 3599, 3628, 3658
<code>\belowpdfbookmark</code>	3549
<code>\bgroup</code>	275, 376
<code>\BOOKMARK</code>	3552
<code>bookmarkstype</code> (hypersetup key) ..	13
bool commands:	
<code>\bool_if:NTF</code>	268, 297, 356, 707, 733, 752, 772, 783, 820, 885, 950, 1028, 1038, 1291, 1302
<code>\bool_if:nTF</code>	900, 1976, 2001, 3293, 3401
<code>\bool_lazy_and:nnTF</code>	424
<code>\bool_lazy_or:nnTF</code>	1321, 1370
<code>\bool_new:N</code> ...	204, 205, 534, 538, 542
<code>\bool_set_true:N</code>	543
<code>bordercolormodel</code> (hypersetup key) ..	13, <u>1088</u>
box commands:	
<code>\box_dp:N</code>	644, 2877, 2928, 2963
<code>\box_ht:N</code>	643, 2876, 2927, 2962
<code>\box_new:N</code>	464, 545
<code>\box_set_dp:Nn</code>	2173
<code>\box_set_ht:Nn</code>	2180
<code>\box_use:N</code> ...	1308, 2879, 2937, 2966
<code>\box_use_drop:N</code>	1313
<code>\box_wd:N</code>	642, 2875, 2926, 2961
<code>\l_tmpa_box</code>	2873, 2875, 2876, 2877, 2879, 2912, 2926, 2927, 2928, 2937, 2959, 2961, 2962, 2963, 2966
C	
<code>\catcode</code>	279, 280
<code>\char</code>	2811, 3008, 3020
<code>\chardef</code>	159
<code>\cite</code>	34
clist commands:	
<code>\clist_item:nn</code>	2233
<code>\clist_map_function:nN</code>	125, 134
<code>\clist_map_inline:nn</code>	2366
<code>\clist_map_variable:nNn</code> ..	2724, 2791
color commands:	
<code>\color_export:nnN</code> ..	41, 435, 987, 1109
<code>\color_select:n</code>	995, 1031, 1312, 2918, 2922
<code>\color_select:nn</code>	1001
<code>\color_set:nn</code>	2, 394, 1012
<code>\color_set:nnn</code>	2, 393, 1018
color names:	
<code>hyp/annot/file</code>	<u>531</u>
<code>hyp/annot/link</code>	<u>531</u>
<code>hyp/annot/menu</code>	<u>531</u>
<code>hyp/annot/run</code>	<u>531</u>
<code>hyp/annot/url</code>	<u>531</u>
<code>colorfile</code> (hypersetup key)	<u>1067</u>
<code>colorlink</code> (hypersetup key)	<u>1067</u>
<code>colorlinks</code> (hypersetup key)	<u>1044</u>
<code>colormenu</code> (hypersetup key)	<u>1067</u>
<code>colorrund</code> (hypersetup key)	<u>1067</u>
<code>colorscheme</code> (hypersetup key)	1, <u>1479</u>
<code>colorurl</code> (hypersetup key)	<u>1067</u>
cs commands:	
<code>\cs_generate_variant:Nn</code> ..	156, 157, 158, 391, 463, 586, 817, 987, 1004, 1021
<code>\cs_gset:Npn</code>	1283
<code>\cs_gset_eq:NN</code>	410, 421
<code>\cs_if_exist:NTF</code>	13
<code>\cs_if_exist_p:N</code>	425

<code>\ExplSyntaxOn</code>	<code>\hbox_set_end:</code>	1304
... 8, 11, 2595, 2652, 2676, 3036,	<code>\hbox_set_to_wd:Nnn</code>	2157
3208, 3229, 3282, 3351, 3389, 3475	<code>hidefile</code> (hypersetup key)	1455
<code>extension</code> (hypersetup key)	<code>hidelink</code> (hypersetup key)	1455
13, 1521	<code>hidelinks</code> (hypersetup key)	1455
	<code>hidemenu</code> (hypersetup key)	1455
F	<code>hiderun</code> (hypersetup key)	1455
<code>\fbox</code>	<code>hideurl</code> (hypersetup key)	1455
<code>\fi</code>	hook commands:	
2578, 2581, 2590, 2690, 2698, 2699,	<code>\hook_gput_code:nnn</code>	
2730, 2748, 2751, 2752, 2758, 2761,	... 403, 414, 1024, 1034, 1287, 1298	
2770, 2772, 2779, 2782, 3109, 3131,	<code>\hook_new:n</code>	392, 561, 656
3141, 3143, 3149, 3153, 3160, 3161,	<code>\hook_new_pair:nn</code>	
3170, 3174, 3178, 3182, 3186, 3190,	... 261, 289, 313, 330, 348	
3194, 3198, 3202, 3206, 3223, 3227,	<code>\hook_use:n</code>	
3326, 3327, 3328, 3335, 3339, 3343,	. 265, 285, 294, 310, 317, 327, 334,	
3345, 3347, 3366, 3370, 3374, 3378,	344, 353, 372, 578, 662, 671, 718, 741	
3385, 3450, 3451, 3458, 3462, 3464,	<code>\href</code>	4, 5, 19, 263
3485, 3486, 3493, 3497, 3499, 3564,	<code>\hreflaunch</code>	5
3569, 3621, 3622, 3633, 3637, 3654	<code>\hrefpdf</code>	5, 19, 314, 384
<code>file</code> (hypersetup key)	<code>\hrefrun</code>	5, 331, 385
10, 1552	<code>\hrefurl</code>	5, 19, 291, 383
file commands:	<code>\hss</code>	3578
<code>\file_input:n</code>	Hy internal commands:	
9	<code>\Hy_hyescapeform</code> ..	3120, 3133, 3139
<code>fileborderstyle</code> (hypersetup key)	<code>\Hy_hypSectionAnchor</code> ...	3568, 3571
14, 1181	HyField internal commands:	
<code>filecolor</code> (hypersetup key)	<code>\HyField_hypAddToFields</code> .	2557, 2584
1067	hyp commands:	
<code>final</code> (hypersetup key)	<code>\l_hyp_annot_colorfile_bool</code> ...	532
1503	<code>\l_hyp_annot_colorlink_bool</code> ...	532
<code>\fontencoding</code>	<code>\l_hyp_annot_colormenu_bool</code> ...	532
2807, 3004, 3016	<code>\l_hyp_annot_colorrrun_bool</code>	532
<code>\fontfamily</code>	<code>\l_hyp_annot_colorurl_bool</code>	532
2806, 3003, 3015	<code>\l_hyp_annot_ocgcolorfile_bool</code> .	536
<code>\fontseries</code>	<code>\l_hyp_annot_ocgcolorlink_bool</code> .	536
2808, 3005, 3017	<code>\l_hyp_annot_ocgcolormenu_bool</code> .	536
<code>\fontshape</code>	<code>\l_hyp_annot_ocgcolorrrun_bool</code> ..	536
2809, 3006, 3018	<code>\l_hyp_annot_ocgcolorurl_bool</code> ..	536
fp commands:	<code>\l_hyp_annot_colorurl_bool</code> ...	536
<code>\fp_eval:n</code>	<code>\l_hyp_annot_colorlink_bool</code> ...	536
2131, 2159, 2175	<code>\l_hyp_annot_colorrrun_bool</code>	532
	<code>\l_hyp_annot_colorurl_bool</code>	532
G	<code>\l_hyp_annot_ocgcolorfile_bool</code> .	536
<code>\gdef</code>	<code>\l_hyp_annot_ocgcolorlink_bool</code> .	536
281, 282	<code>\l_hyp_annot_ocgcolormenu_bool</code> .	536
<code>\GetDocumentProperties</code>	<code>\l_hyp_annot_ocgcolorrrun_bool</code> ..	536
3	<code>\l_hyp_annot_ocgcolorurl_bool</code> ..	536
<code>\global</code>	<code>\l_hyp_annot_colorfile_bool</code> ...	532
2657, 3099	<code>\l_hyp_annot_colorlink_bool</code> ...	532
group commands:	<code>\l_hyp_annot_colorrrun_bool</code>	532
<code>\group_begin:</code>	<code>\l_hyp_annot_colorurl_bool</code>	532
266, 295, 318,	<code>\l_hyp_annot_ocgcolorfile_bool</code> .	536
335, 354, 576, 713, 725, 739, 745,	<code>\l_hyp_annot_ocgcolorlink_bool</code> .	536
777, 806, 825, 873, 890, 932, 955,	<code>\l_hyp_annot_ocgcolormenu_bool</code> .	536
981, 1030, 1294, 1311, 2435, 2805,	<code>\l_hyp_annot_ocgcolorrrun_bool</code> ..	536
2855, 2898, 2946, 2979, 3002, 3014	<code>\l_hyp_annot_ocgcolorurl_bool</code> ..	536
<code>\group_end:</code>	<code>\l_hyp_annot_colorurl_bool</code> ...	532
309, 326,	<code>\l_hyp_annot_colorlink_bool</code> ...	532
343, 371, 583, 721, 725, 757, 761,	<code>\l_hyp_annot_colorrrun_bool</code>	532
803, 806, 870, 873, 930, 932, 979,	<code>\l_hyp_annot_colorurl_bool</code>	532
981, 1040, 1314, 1317, 2442, 2812,	<code>\l_hyp_annot_ocgcolorfile_bool</code> .	536
2891, 2939, 2972, 3009, 3021, 3031	<code>\l_hyp_annot_ocgcolorlink_bool</code> .	536
	<code>\l_hyp_annot_ocgcolormenu_bool</code> .	536
	<code>\l_hyp_annot_ocgcolorrrun_bool</code> ..	536
	<code>\l_hyp_annot_ocgcolorurl_bool</code> ..	536
	<code>\l_hyp_current_dest_name_tl</code>	
	... 655, 714, 740	
	hyp internal commands:	
	<code>\g__hyp_AcroForm_CoFields_prop</code> ..	
	... 2521, 2539, 2608, 2610, 2625	
	<code>\g__hyp_AcroForm_Fields_prop</code> ...	
	... 2522, 2532, 2603	
	<code>\l__hyp_annot_GoTo_bool</code> 707, 733, 752	
	<code>\l__hyp_annot_GoTo_bool\l__-</code>	
	<code>hyp_annot_URI_bool\l__-</code>	
	<code>hyp_annot_GoToR_bool\l__-</code>	
	<code>hyp_annot_Named_bool\l__-</code>	
	<code>hyp_annot_Launch_bool</code>	540
<code>\hbox</code>	<code>\l__hyp_annot_GoToR_bool</code>	820
376, 3573	<code>\l__hyp_annot_Launch_bool</code>	885
hbox commands:		
<code>\hbox_overlap_right:n</code>		1308
<code>\hbox_set:Nn</code>		2873, 2912, 2959
<code>\hbox_set:Nw</code>		1295

\l__hyp_annot_Named_bool	950	\l__hyp_filename_tmpa_tl	469, 826, 828, 833, 834, 839, 894, 895, 913
\c__hyp_annot_types_seq	476, 532, 536, 1067, 1368, 1466	__hyp_href_pdf_aux:nn	320, 323
\l__hyp_annot_URI_bool	772	\l__hyp_href_pdf_destination_tl	207, 218, 257, 325
\g__hyp_bordercolormodel_str	437, 531, 1092, 1111	\l__hyp_href_pdf_page_tl	208, 226, 850
__hyp_check_link_nesting:TF	616, 624, 626, 710, 736, 754, 775, 823, 888, 953	__hyp_href_run_aux:nn	337, 340
__hyp_citebordercolor_hook_init:	399, 412, 421	\l__hyp_href_run_parameter_tl	209, 230, 342
__hyp_citecolor_hook_init:	397, 401, 410	__hyp_href_url_aux:n	363, 366
__hyp_clist_display:n	118, 125, 134	__hyp_href_url_aux:nn	304, 306
__hyp_color_select:n	41, 988, 988, 1004	\l__hyp_href_url_encode_bool	204, 215, 253, 268, 297, 356
__hyp_color_select_aux:wn	988, 992, 999	__hyp_href_url_format:	210, 216, 258, 369
__hyp_color_set:nn	40, 41, 397, 399, 434, 1005, 1005, 1021, 1072, 1108	\l__hyp_href_url_ismap_bool	205, 229, 783
__hyp_color_set_aux:nwn	1005, 1009, 1016	\l__hyp_href_url_protocol_tl	206, 217, 256, 308, 370
\c__hyp_colorscheme_daleif_prop	3713	__hyp_if_outer_link:	610
\c__hyp_colorscheme_henryford_prop	3804	__hyp_if_outer_link:TF	626
\c__hyp_colorscheme_julian_prop	3728	\l__hyp_link_Contents_tl	27, 35, 36, 511, 519, 526, 717, 788
\c__hyp_colorscheme_phelype_prop	3789	__hyp_link_goto_begin:nw	678, 719, 742
\c__hyp_colorscheme_szabolcsA_prop	3758	__hyp_link_goto_end:	700, 720, 756
\c__hyp_colorscheme_szabolcsB_prop	3773	\g__hyp_linknestlevel_int	609, 612, 709, 727, 735, 763, 774, 808, 822, 875, 887, 933, 952, 982
\c__hyp_colorscheme_tivv_prop	3743	g__hyp_linknestlevel_int	609
\l__hyp_dest_box	32, 545, 642, 643, 644, 2157, 2173, 2180	\c__hyp_map_annot_hyp_prop	476
\l__hyp_dest_name_tmpa_tl	469, 681, 682, 688, 692, 694, 697, 843, 856	\c__hyp_map_hyp_annot_prop	476, 1022, 1095, 1135, 1162, 1170, 1181, 1208, 1216, 1285, 1393, 1429, 1439, 1552
\l__hyp_dest_pdfremotestartview_tl	495, 851, 1964, 1968	__hyp_ocg_init:	1227, 1227, 1283, 1293
\g__hyp_dest_pdfstartpage_tl	495, 1975, 1977, 1984, 2002, 2009	\l__hyp_optlang_regex	2205, 2206, 2235
\g__hyp_dest_pdfstartview_tl	495, 1977, 1984, 1995, 1999, 2002, 2009	l__hyp_page/Trans	559
\l__hyp_dest_pdfview_tl	530, 660, 669, 2126, 2129, 2136, 2139, 2140, 2141, 2142, 2143, 2144, 2149, 2153, 2189	__hyp_PageLabels_gpush:	593, 593, 606
\c__hyp_dest_startview_regex	546, 1962, 1993	\l__hyp_para_tmpa_tl	472, 899, 902, 913
\c__hyp_dest_undefined_tl	475, 687, 688	\l__hyp_para_tmpa_tl\l__hyp_text_tmpa_str	469
__hyp_destination:nn	32, 629, 629, 660, 669	__hyp_property_record:nn	456, 457, 463, 2661
		__hyp_secondoftwewithopt:wnn	382, 383, 384, 385
		__hyp_setup_info_date_key:nn	2288, 2321, 2322
		__hyp_setup_info_key:nn	2207, 2281, 2282, 2283, 2284, 2286, 2287
		__hyp_store_metadata:nn	386, 391, 1571, 2249, 2267, 2301, 2316, 2325, 2341, 2409

_hyp_text_cleanup:N	566, 566, 580	colorfile	1067
\l_hyp_text_enc_dest_print_tl	498, 842	colorlink	1067
\l_hyp_text_enc_dest_tl	32, 498, 636, 693	colorlinks	1044
\l_hyp_text_enc_file_print_tl	498, 893	colormenu	1067
\l_hyp_text_enc_info_print_tl	498, 518, 525, 591	colorrun	1067
\l_hyp_text_enc_para_print_tl	498, 898	colorscheme	1, 1479
\l_hyp_text_enc_uri_print_tl	270, 273, 299, 302, 358, 361, 498, 780, 1563	colorurl	1067
_hyp_text_pdfstring:nnN	516, 523, 574, 574, 586, 588, 591, 634, 691, 778, 840, 891, 896, 1563	debug	1503
_hyp_text_pdfstring_info:nN	589, 589, 2238, 2242, 2261, 2271	destlabel	13
_hyp_text_purify:nN	562, 562, 579	draft	1503
_hyp_text_string_from_unicode:nN	570, 570, 581	extension	13, 1521
\g_hyp_text_tmpa_str	474, 582, 584	file	10, 1552
\l_hyp_text_tmpa_str	473, 579, 580, 581, 582	fileborderstyle	14, 1181
\l_hyp_tmpa_box	464, 1295, 1308, 1313	filecolor	1067
\l_hyp_tmpa_int	464	final	1503
\l_hyp_tmpa_seq	464, 1962, 1964, 1993, 1995, 2116, 2117, 2121, 2123, 2147, 2154, 2155, 2156, 2165, 2167, 2177, 2182, 2235, 2236, 2242, 2612, 2614, 2620	hidefile	1455
\l_hyp_tmpa_str	464, 2238, 2242, 2244, 2246, 2261, 2262, 2264, 2271, 2272, 2276	hidelink	1455
\l_hyp_tmpa_tl	464, 637, 641, 649, 1112, 1116, 1563, 1564, 1569, 1960, 1962, 1991, 1993, 2123, 2124, 2131, 2233, 2235	hidelinks	1455
\l_hyp_uri_tmpa_tl	469, 781, 782	hidemenu	1455
hyp/anchor	655	hiderun	1455
hyp/annot/file (color name)	531	hideurl	1455
hyp/annot/link (color name)	531	hypertexnames	1521
hyp/annot/menu (color name)	531	link	10, 1552
hyp/annot/run (color name)	531	linkborder	14
hyp/annot/url (color name)	531	linkborderstyle	14, 1181
hyp/text/pdfstring	561	linkcolor	1067
\hypercalcbp	12, 18, 162	linkfileprefix	1521
\HyperDestNameFilter	13, 635, 692	linktoc	1521
\hypersetup	1, 2, 5, 9, 10, 13, 19, 68, 101, 181	linktocpage	1521
\hypersetup keys:		localanchorname	1521
allcolors	1067	menu	10, 1552
bookmarkstyle	13	menuborder	14
bordercolormodel	13, 1088	menuborderstyle	14, 1181
		menucolor	1067
		naturalnames	1521
		nested-links	10
		nesting	14
		ocgcolorfile	1321
		ocgcolorlink	1321
		ocgcolorlinks	1321
		ocgcolormenu	1321
		ocgcolorrun	1321
		ocgcolorurl	1321
		pageanchor	1521
		pdfauthor	2205
		pdfborder	14
		pdfborderstyle	14, 1181
		pdfcreationdate	14, 2288
		pdfcreator	2205
		pdfencoding	1493
		pdfinfo	2355
		pdfkeywords	2205
		pdflang	14, 2194
		pdflinkmargin	14

pdfmetadate	14, 2288	
pdfmoddate	14, 2288	
pdfproducer	2205	
pdfremotestartview	12	
pdfstartview	12	
pdfsubject	2205	
pdftitle	2205	
pdftrapped	2327	
pdfversion	1493	
pdfview	12 , 2112	
plainpages	1521	
run	10 , 1552	
runborder	14	
runborderstyle	14, 1181	
runcolor	1067	
unicode	1493	
url	10 , 1552	
urlborder	14	
urlborderstyle	14, 1181	
urlcolor	1067	
verbose	1503	
hypertextnames (hypersetup key)	1521	
hypListbox internal commands:		
<code>_hypListbox</code>	2780 , 2839	
hypRadio internal commands:		
<code>_hypRadio</code>	2765 , 2787	
hypwritetorep internal commands:		
<code>_hypwritetorep</code>	3542	
I		
<code>\ifcase</code>	3326 , 3450 , 3485	
<code>\ifcsname</code>	3103	
<code>\ifdim</code>	2730 , 2769 , 2770	
<code>\ifnum</code>	3157 , 3326 , 3332 , 3363 , 3450 , 3455 , 3485 , 3490 , 3617 , 3635 , 3650	
<code>\iftrue</code>	3619	
<code>\ifx</code>	2561 , 2699 , 2759 , 3134 , 3146 , 3150 , 3167 , 3171 , 3175 , 3179 , 3183 , 3187 , 3191 , 3195 , 3199 , 3203 , 3220 , 3224 , 3327 , 3328 , 3336 , 3340 , 3347 , 3367 , 3371 , 3378 , 3379 , 3451 , 3459 , 3486 , 3494 , 3567 , 3629	
<code>\immediate</code>	2545 , 2548	
int commands:		
<code>\int_compare:nNnTF</code>	612 , 2121 , 2147 , 2831 , 3249 , 3302 , 3412 , 3434	
<code>\int_compare_p:nNn</code>	3295 , 3403	
<code>\int_eval:n</code>	849	
<code>\int_gdecr:N</code>	727 , 763 , 808 , 875 , 933 , 982	
<code>\int_gincr:N</code>	709 , 735 , 774 , 822 , 887 , 952	
<code>\int_max:nn</code>	850	
<code>\int_new:N</code>	467 , 609	
iow commands:		
<code>\iow_newline:</code>	3052 , 3058 , 3071	
K		
<code>\kern</code>	3574	
keys commands:		
<code>\keys_define:nn</code>	187 , 195 , 213 , 251 , 395 , 620 , 1044 , 1069 , 1076 , 1088 , 1097 , 1122 , 1137 , 1156 , 1183 , 1202 , 1325 , 1339 , 1351 , 1374 , 1380 , 1395 , 1424 , 1455 , 1468 , 1479 , 1493 , 1503 , 1508 , 1521 , 1533 , 1554 , 1559 , 2112 , 2194 , 2209 , 2252 , 2290 , 2305 , 2323 , 2327 , 2355 , 2407 , 2413 , 2426 , 2446	
<code>\l_keys_key_str</code>	192 , 2276 , 2511	
<code>\keys_set:nn</code>	184 , 221 , 240 , 267 , 296 , 319 , 336 , 355 , 405 , 416 , 1485 , 1492 , 2332 , 2359 , 2362 , 2363 , 2364 , 2365 , 2436 , 2511	
<code>\keys_set_known:nn</code>	2514	
<code>\kvsetkeys</code>	183 , 2599	
L		
<code>\label</code>	9 , 13	
<code>\LayoutCheckField</code>	2992	
<code>\LayoutChoiceField</code>	2762	
<code>\LayoutPushButtonField</code>	2868 , 2885	
<code>\LayoutTextField</code>	2700	
<code>\leavevmode</code>	2701 , 2798 , 3572	
legacy commands:		
<code>\legacy_if:nTF</code>	169 , 442 , 452 , 601 , 2637 , 2864 , 2905 , 2955 , 2988 , 3272	
<code>\let</code>	383 , 384 , 385 , 729 , 798 , 810 , 866 , 877 , 926 , 935 , 984 , 2524 , 2525 , 2552 , 2648 , 2649 , 2650 , 2682 , 2683 , 2717 , 2718 , 2760 , 3099 , 3106 , 3108 , 3113 , 3130 , 3164 , 3583 , 3597 , 3600 , 3609 , 3620 , 3627 , 3630 , 3632 , 3640 , 3648	
link (hypersetup key)	10 , 1552	
linkborder (hypersetup key)	14	
linkborderstyle (hypersetup key)	14, 1181	
linkcolor (hypersetup key)	1067	
linkfileprefix (hypersetup key)	1521	
linktoc (hypersetup key)	1521	
linktocpage (hypersetup key)	1521	
localanchorname (hypersetup key)	1521	
<code>\long</code>	7	
M		
<code>\MakeButtonField</code>	2873 , 2888 , 2912 , 2959 , 2970	
<code>\MakeChoiceField</code>	2848	
<code>\MakeFieldObject</code>	2515	
<code>\MakeRadioField</code>	2830	
<code>\MakeTextField</code>	2709	
<code>\mbox</code>	1305	

menu (hypersetup key)	10 , 1552
menuborder (hypersetup key)	14
menuborderstyle (hypersetup key)	14 , 1181
menucolor (hypersetup key)	1067
mode commands:	
\mode_if_horizontal:TF	631, 653
\mode_leave_vertical:	264, 293, 316, 333, 352, 680, 792, 858, 915, 958, 2842, 2870, 2887, 2909, 2951, 2994
msg commands:	
\msg_error:nn	426, 2884, 2969
\msg_info:nnn	2219, 2224
\msg_line_context:	83
\g_msg_module_name_prop	12
\msg_new:nnn	51, 58, 63, 67, 71, 78, 85, 92, 99, 106, 112, 119, 128, 138, 149
\msg_new:nnnn	15, 26, 37
\msg_warning:nn	171, 1499
\msg_warning:nnn	180, 684, 976
\msg_warning:nnnn	1343, 1384, 1597, 1641, 1675, 1698, 1760, 1789, 1823, 1838, 1851, 1864, 1889, 1924, 1940, 1967, 1998, 2031, 2045, 2072, 2106, 2188
\msg_warning:nnnnn	191, 1415, 1449, 1541, 1617, 1654, 1741, 1774, 1799, 1875, 1902, 1953, 2058, 2085, 2349, 2453, 2472, 2491, 2502
N	
naturalnames (hypersetup key)	1521
nested-links (hypersetup key)	10
nesting (hypersetup key)	14
\newcommand	165, 3546, 3548, 3550
\newcount	2653, 2838
\NewDocumentCommand	2515
\NewExpandableDocumentCommand	382
\newlength	3565
\noexpand	2736, 2737, 2738, 2739, 3117
\nolinkurl	4
O	
ocgcolorfile (hypersetup key)	1321
ocgcolorlink (hypersetup key)	1321
ocgcolorlinks (hypersetup key)	1321
ocgcolormenu (hypersetup key)	1321
ocgcolorrun (hypersetup key)	1321
ocgcolorurl (hypersetup key)	1321
P	
pageanchor (hypersetup key)	1521
\paperwidth	3
\PassOptionsToPackage	454, 1513, 1518
pdf commands:	
\pdf_bdcobject:nn	1307, 1310
\pdf_destination:nn	27, 156, 648
\pdf_destination:nnnn	641
\pdf_emc:	1309, 1315
\pdf_link_user:nnn	2845
\pdf_name_from_unicode_e:n	246, 830, 960
\pdf_object_if_exist:nTF	828
\pdf_object_new:n	1229, 1230, 1231, 1232, 3039, 3040, 3041
\pdf_object_ref:n	157, 839, 1235, 1237, 1259, 1260, 1263, 1267, 1272, 1277, 1282, 2630, 2632, 3097
\pdf_object_ref_last:	860, 2441
\pdf_object_unnamed_write:nn	859, 2437
\pdf_object_write:nnn	1233, 1239, 1249, 1261, 3042, 3084, 3091
\pdf_pageobject_ref:n	158, 1984, 2009, 2669
\pdf_string_from_unicode:nnN	572
\pdf_version:	1645, 1764, 1793, 1827, 1842, 1868, 1893, 1928, 1944, 2049, 2076
\pdf_version_compare:NnTF	1635, 1726, 1748, 1784, 1817, 1832, 1858, 1883, 1911, 1934, 2039, 2066
\pdf_version_compare_p:Nn	904, 1322, 1371
\pdf_version_major:	168, 1323, 1346, 1372, 1388
\pdf_version_minor:	167, 1346, 1388
pdfannot commands:	
\pdfannot_box:nnnn	2704, 2815, 2874, 2925, 2960, 3024
\pdfannot_box_ref_last:	2559, 2575
\pdfannot_dict_put:nnn	716, 788, 793, 860, 916, 961, 1113, 1148, 1172, 1194, 1218, 1400, 1431
\pdfannot_dict_remove:nn	1103, 1143, 1164, 1189, 1210, 1408, 1441
\pdfannot_link:nnn	794, 861, 917, 962
\pdfannot_link_goto_begin:nw	697
\pdfannot_link_goto_end:	702
\pdfannot_link_margin:n	8, 1679
\c_pdfannot_link_types_seq	540
pdfauthor (hypersetup key)	2205
\pdfbookmark	3551
pdfborder (hypersetup key)	14
pdfborderstyle (hypersetup key)	14, 1181
pdfcreationdate (hypersetup key)	14, 2288
pdfcreator (hypersetup key)	2205
\pdfdest	11
pdfdict commands:	
\pdfdict_new:n	559, 766, 813, 879, 944

\pdfdict_put:nnn	238, 245, 560, 767, 768, 782, 785, 814, 815, 830, 836, 846, 856, 880, 881, 895, 910, 945, 946, 959, 1705, 1706, 1710, 1711, 2450, 2460, 2464, 2467, 2469, 2488, 2497, 2499
\pdfdict_remove:nn	235, 907, 1715, 1716
\pdfdict_use:n	793, 859, 916, 922, 961, 966, 2439
pdfencoding (hypersetup key)	1493
\pdfescapestring	3106
pdffile commands:	
\pdffile_embed_file:nnn	817, 831
PDFForm internal commands:	
\PDFForm_hypName	3145, 3148, 3152, 3155
pdfinfo (hypersetup key)	2355
pdfkeywords (hypersetup key)	2205
pdflang (hypersetup key)	14, 2194
pdflinkmargin (hypersetup key)	14
pdfmanagement commands:	
\pdfmanagement_add:nn	1688
\pdfmanagement_add:nnn	595, 1259, 1260, 1282, 1569, 1589, 1605, 1609, 1629, 1637, 1666, 1721, 1731, 1755, 1769, 1781, 1786, 1811, 1819, 1834, 1860, 1885, 1919, 1936, 1982, 2007, 2021, 2041, 2068, 2097, 2200, 2220, 2225, 2246, 2264, 2275, 2299, 2314, 2336, 2422, 2441, 2605, 2622, 2629, 2631, 2633, 2639
\pdfmanagement_if_active_p:	425
\pdfmanagement_remove:nn	1566, 1585, 1593, 1613, 1625, 1650, 1662, 1670, 1684, 1693, 1737, 1752, 1771, 1808, 1847, 1872, 1898, 1915, 1949, 1979, 2004, 2017, 2026, 2054, 2081, 2093, 2102, 2229, 2258, 2296, 2311, 2419, 2432, 2643
\pdfmanagement_show:n	2606
pdfmeta commands:	
\pdfmeta_standard_verify:nnTF	956, 2861, 2952
\pdfmeta_standard_verify:nTF	2635, 3213
pdfmetadate (hypersetup key)	14, 2288
pdfmoddate (hypersetup key)	14, 2288
pdfproducer (hypersetup key)	2205
pdfremotestartview (hypersetup key)	12
pdfstartview (hypersetup key)	12
\pdfstringdef	8, 9, 23, 29, 3162
\pdfstringdefDisableCommands	383, 384, 385
pdfsubject (hypersetup key)	2205
pdftitle (hypersetup key)	2205
pdftrapped (hypersetup key)	2327
pdfversion (hypersetup key)	1493
pdfview (hypersetup key)	12, 2112
pdfxform commands:	
\pdfxform_if_exist:nTF	2801, 2913, 2997
\pdfxform_new:nnn	2517, 2803, 2916, 2920, 2999, 3011
\pdfxform_ref:n	2825, 2826, 2932, 2933, 3244, 3245
\phantom	3020
\phantomsection	13, 19, 3636
plainpages (hypersetup key)	1521
prg commands:	
\prg_do_nothing:	410, 421
\prg_generate_conditional_	
variant:Nnn	2664
\prg_new_conditional:Npnn	610
\prg_return_false:	613
\prg_return_true:	614
prop commands:	
\prop_const_from_keyval:Nn	478, 486, 3698, 3713, 3728, 3743, 3758, 3773, 3789, 3804
\prop_gput:Nnn	12, 2532, 2539
\prop_if_empty:NTF	2608
\prop_item:Nn	2625
\prop_map_inline:Nn	1022, 1095, 1135, 1162, 1170, 1181, 1208, 1216, 1285, 1393, 1429, 1439, 1483, 1552, 2603, 2610
\prop_new:N	2521, 2522
property commands:	
\property_if_recorded:nn	2664
\property_if_recorded:nnTF	2667
\property_record:nn	460
\property_ref:nn	2671
\property_ref_undefined_warn:nn	2662
\protect	729, 798, 810, 866, 877, 926, 935, 984, 3567
\providecommand	163, 164, 2546, 2549, 2714, 3102
\ProvidesFile	3
R	
\ReadBookmarks	3556
\refstepcounter	12
regex commands:	
\regex_const:Nn	546
\regex_extract_once:NnN	2235
\regex_extract_once:NnNTF	1962, 1993
\regex_new:N	2205
\regex_set:Nn	2206

<code>\relax</code>	729, 798, 810, 866, 877, 926, 935, 984, 2717, 2718, 2759, 3099, 3146, 3150, 3220, 3224, 3327, 3328, 3336, 3340, 3367, 3371, 3451, 3459, 3486, 3494, 3620, 3629	<code>\str_gset:Nn</code>	1092
<code>\RemoveFromHook</code>	1575	<code>\str_gset_eq:NN</code>	582
<code>\RenewDocumentCommand</code>	174	<code>\str_head:n</code>	2338, 2343
<code>\RequirePackage</code>	6, 32, 446, 3554	<code>\str_if_eq:nnTF</code>	638, 2244, 2262, 2272, 3211
<code>run</code> (hypersetup key)	10, 1552	<code>\str_if_eq_p:nn</code>	902, 1323, 1372
<code>runborder</code> (hypersetup key)	14	<code>\str_lowercase:n</code>	2117, 2267, 2316, 2339, 2344
<code>runborderstyle</code> (hypersetup key) ..	14, 1181	<code>\str_new:N</code>	468, 473, 474, 531
<code>runcolor</code> (hypersetup key)	1067	<code>\str_set:Nn</code>	564
S		<code>\str_set_convert:Nnnn</code>	8
<code>\selectfont</code>	2810, 3007, 3019	<code>\str_set_eq:NN</code>	584
seq commands:		<code>\str_tail:n</code>	2339, 2344
<code>\seq_const_from_clist:Nn</code>	476	<code>\str_uppercase:n</code>	2332, 2338, 2343
<code>\seq_count:N</code>	2121, 2147	<code>\string</code>	2546, 2549, 2570, 2588, 3122, 3126, 3137
<code>\seq_get_right:NN</code>	2123	<code>\subpdfbookmark</code>	3547
<code>\seq_if_empty:NTF</code>	2236	sys commands:	
<code>\seq_item:Nn</code>	1964, 1995, 2117, 2165, 2167, 2177, 2182, 2242	<code>\c_sys_backend_str</code>	3509
<code>\seq_map_inline:Nn</code>	532, 536, 540, 1067, 1368, 1466, 2620	T	
<code>\seq_new:N</code>	466	TeX and L ^A T _E X 2 _ε commands:	
<code>\seq_put_right:Nn</code>	2154, 2155, 2156, 2612	<code>\@BOOKMARK</code>	3553
<code>\seq_set_split:Nnn</code>	2116	<code>\@CheckBox</code>	2975
<code>\seq_sort:Nn</code>	2614	<code>\@ChoiceMenu</code>	2715
<code>\setbox</code>	376	<code>\@Form</code>	2597
<code>\setlength</code>	3586, 3657	<code>\@PushButton</code>	2853
<code>\setpdfinkmargin</code>	8	<code>\@Reset</code>	2942
<code>\settowidth</code>	2729	<code>\@Submit</code>	2894
<code>\show</code>	2789	<code>\@TextField</code>	2680
socket commands:		<code>\@bookmarksopenlevel</code>	1580
<code>\socket_assign_plug:nn</code>	528, 529	<code>\@bsphack</code>	459
<code>\socket_new:nn</code>	512, 513	<code>\@chapter</code>	3609, 3610
<code>\socket_new_plug:nnn</code>	514, 521	<code>\@currDisplay</code>	2729, 2797
<code>\socket_use:nn</code>	787	<code>\@currValue</code>	2795, 2825
<code>\socket_use:nnn</code>	715	<code>\@currentHref</code>	3590, 3603, 3614, 3644, 3661
sort commands:		<code>\@curropt</code> ..	2714, 2724, 2727, 2791, 2794
<code>\sort_return_same:</code>	2618	<code>\@empty</code> ..	432, 2524, 2525, 2561, 2648, 2649, 2650, 2682, 2683, 2699, 3167, 3171, 3175, 3179, 3183, 3187, 3191, 3195, 3199, 3203, 3347, 3378, 3379
<code>\sort_return_swapped:</code>	2617	<code>\@endForm</code>	2648
<code>\space</code>	2797, 3329, 3346, 3347, 3377, 3378, 3452, 3484, 3487	<code>\@esphack</code>	461
<code>\spacefactor</code>	631, 653	<code>\@firstofone</code> ..	2552, 3108, 3113, 3130
<code>\stockwidth</code>	3	<code>\@firstoftwo</code>	3651
str commands:		<code>\@for</code>	2725, 2792
<code>\c_colon_str</code>	939	<code>\@gobbletwo</code>	3600
<code>\c_hash_str</code>	789	<code>\@hyper@launch</code>	39, 939
<code>\c_percent_str</code>	790	<code>\@ifnextchar</code>	275
<code>\str_case:nnTF</code>	2215, 3509	<code>\@ifundefined</code> ..	3122, 3581, 3608, 3618
<code>\str_case_e:nnTF</code>	2117	<code>\@mainaux</code>	2545, 2548, 2568, 2587
<code>\str_compare:nNnTF</code>	2616	<code>\@mkboth</code>	3600
		<code>\@one</code>	2657

\@nil	3120, 3133, 3139	\Fld@height	2685, 2706, 2709, 2720, 2778, 2817, 2830, 2848, 2897, 2945, 2981, 3026
\@part	3627, 3634	\Fld@keystroke@code	3167, 3169
\@pdfauthor	23	\Fld@listcount	2788, 2796, 2831, 2838
\@pdfborder	163	\Fld@mappingname	3150, 3152, 3224, 3226
\@pdfborderstyle	164	\Fld@maxlen	3434, 3436
\@savsf	631, 653	\Fld@menulength	2722, 2728, 2775
\@schapter	3597, 3598	\Fld@name	2681, 2716, 2854, 2977, 3145, 3219
\@secondoftwo	3653	\Fld@onblur@code	3187, 3189
\@sect	3648, 3649	\Fld@onclick@code	3312
\@spart	3640, 3641	\Fld@onenter@code	3199, 3201
\@ssect	3583, 3584	\Fld@onexit@code	3203, 3205
\@tempdima	2723, 2730, 2732, 2769, 2770, 2771, 2775, 2776, 2777, 2778	\Fld@onfocus@code	3183, 3185
\@tempdimb	2729, 2730	\Fld@onmousedown@code	3191, 3193
\@typeset@protect	3567	\Fld@onmouseup@code	3195, 3197
\BKM@color	432, 438	\Fld@pageobjref	2650, 2665, 3102, 3234, 3287, 3320, 3356, 3395, 3444, 3479
\c@secnumdepth	3617, 3635, 3650	\Fld@radiosymbol	3375
\calc@bm@number	3559	\Fld@rotation	3249, 3251, 3295, 3302, 3304, 3326, 3332, 3334, 3363, 3365, 3403, 3412, 3414, 3450, 3455, 3457, 3485, 3490, 3492
\check@bm@number	3558	\Fld@submitflags	3471
\define@key	429	\Fld@validate@code	3175, 3177
\Fld@additionalactions	3209, 3279, 3313, 3349, 3386, 3433, 3473, 3502	\Fld@value	2683, 2699, 2718, 2759, 2760, 3432
\Fld@align	3238, 3324, 3399	\Fld@width	2684, 2698, 2705, 2709, 2719, 2758, 2769, 2771, 2816, 2830, 2848, 2866, 2896, 2907, 2944, 2956, 2980, 2990, 3025
\Fld@altname	3146, 3148, 3220, 3222	\Fld@X@additionalactions	3166, 3211, 3214
\Fld@annotflags	3233, 3286, 3319, 3355, 3394, 3443, 3478	\Form@action	3469
\Fld@annotnames	3218, 3235, 3288, 3321, 3357, 3396, 3445, 3480	\H@old@part	3627, 3638
\Fld@bcolor	3257, 3259, 3340, 3342, 3371, 3373, 3407, 3420, 3422	\H@old@schapter	3597, 3606
\fld@bcolor	3328	\H@old@sect	3648, 3659, 3668
\Fld@bordercolor	3253, 3255, 3297, 3306, 3308, 3327, 3336, 3338, 3367, 3369, 3405, 3416, 3418, 3451, 3459, 3461, 3486, 3494, 3496	\H@old@spart	3640, 3646
\Fld@borderstyle	3239, 3292, 3325, 3361, 3400, 3449, 3500	\H@old@ssect	3583, 3588
\Fld@borderwidth	2776, 2777, 3239, 3292, 3325, 3361, 3400, 3449, 3500	\href@	275, 281
\Fld@calculate@code	2561, 3179, 3181	\href@split	281, 282
\Fld@calculate@sortkey	2572	\Hy@abspage	605
\Fld@cbsymbol	3261	\Hy@activeanchorfalse	676
\Fld@charsize	2775, 3265, 3346, 3377, 3428, 3484	\Hy@activeanchortrue	667
\Fld@checkequals	2727, 2794	\Hy@AtBeginDocument	2526, 2542
\Fld@choices	3348	\Hy@bookmarkstype	1581
\Fld@color	3266, 3268, 3347, 3378, 3429	\Hy@chapapp	3601, 3612
\Fld@default	2682, 2699, 2717, 2760, 2789, 2790, 2978, 3379, 3383, 3384, 3431	\Hy@colorlink	35
\Fld@flags	3237, 3290, 3323, 3359, 3398, 3447, 3482	\Hy@currentbookmarklevel	3540
\Fld@format@code	3171, 3173	\Hy@DisableOption	173
		\Hy@drafttrue	1512

\Hy@escapeform	2703, 2800, 2844, 2872, 2911, 2958, 2996, 3104, 3111, 3114	\HyField@AddToFields .	2555, 2710, 2832, 2849, 2880, 2936, 2965, 3029
\Hy@escapestring	3106, 3108, 3113, 3116, 3117, 3120, 3121, 3130, 3133, 3169, 3173, 3177, 3181, 3185, 3189, 3193, 3197, 3201, 3205, 3261, 3312, 3375, 3431, 3432, 3469	\HyField@afields	2524
\Hy@finaltrue	1517	\HyField@AfterAuxOpen	2526, 2552, 2585
\Hy@FormObjects	2602, 3037, 3099	\HyField@AuxAddToCoFields	2537, 2549, 2570
\Hy@gttemp	3162, 3164	\HyField@AuxAddToFields	2530, 2546, 2588
\Hy@href	275	\HyField@cofields	2525
\Hy@href@nextactionraw	239	\HyField@FlagsCheckBox	2987
\Hy@href@page	227	\HyField@FlagsChoice	2768
\Hy@linkfileprefix	1526	\HyField@FlagsPushButton	2863, 2903, 2954
\Hy@linktoc	1537	\HyField@FlagsRadioButton	2764
\Hy@MakeCurrentHref	200	\HyField@FlagsSubmit	2904
\Hy@MakeCurrentHrefAuto	3585, 3601, 3612, 3642, 3656	\HyField@FlagsText	2697
\Hy@next	3611, 3620, 3623	\HyField@PDFChoices	2841
\Hy@numberline	166, 3541	\HyField@SetKeys	2693, 2734, 2737, 2756, 2856, 2899, 2947, 2982
\Hy@org@chapter	3609, 3624	\HyPat@ObjRef .	3513, 3520, 3527, 3534
\Hy@OutlineName	3557	\hyper@@link	283
\Hy@OutlineRerunCheck	3555	\hyper@anchor	655
\Hy@pdfmajorversion	168	\hyper@anchorend	655, 3576, 3603, 3614, 3644
\Hy@pdfminorversion	167	\hyper@anchorstart	655, 3576, 3603, 3614, 3644
\Hy@pdfstringtrue	30, 577	\hyper@link	34, 35, 705
\Hy@pdfversion	3157	\hyper@linkend	34, 35, 750
\Hy@pstringdef	587, 3137	\hyper@linkfile	325, 818
\Hy@PutCatalog	593	\hyper@linklaunch ..	38, 342, 883, 942
\Hy@raisedlink	3575, 3602, 3613, 3643	\hyper@linknamed	40, 176, 948
\Hy@RestoreLastskip	652	\hyper@linkstart	34, 35, 731
\Hy@ReturnAfterFi	7, 3138	\hyper@linkurl	308, 369, 770
\Hy@safe@activestru	791, 2564	\hyper@normalise	275, 304, 320, 337, 363, 376
\Hy@SaveLastskip	632	\HyPL@Labels	595, 605
\Hy@secnum@part	3630, 3632, 3635	\HyPL@StorePageLabel	593
\Hy@SectionAnchorHref	3566, 3590, 3661	\HyPsd@SanitizeForOutFile	3544
\Hy@SectionHShift	3565, 3574, 3586, 3657	\if@filesw	2544, 2567, 2586
\Hy@StepCount	2728, 2796	\ifFld@combo	2743, 2773
\Hy@temp	2565, 2580, 3119, 3120	\ifFld@hidden	2698, 2758
\Hy@unicodefalse	3159	\ifFld@multiline	2686
\Hy@VerboseAnchor	633	\ifFld@popdown	2744
\Hy@VerboseLinkStart	712, 738	\ifFld@radio	2740, 2763
\Hy@VerboseLinkStop	722, 758, 801, 868, 928, 972	\ifHy@implicit	3561
\Hy@VersionChecked	159	\ifHy@pdfescapeform	3105, 3115
\Hy@WrapperDef	629	\ifHy@unicode	3158
\Hy@xspace@end	719, 800, 867, 927, 971	\kv@set@family@handler	179
\HyAnn@AbsPageLabel	2649, 2655, 2702, 2799, 2843, 2871, 2910, 2957, 2995	\m@ne	3617, 3632
\HyAnn@Count	2653, 2654, 2657, 2658, 2659, 2660, 2661, 2662, 2667, 2671	\OBJ@OCG@view	165
		\p@	2732
		\pdf@ifdraftmode	2600
		\PDF@SetupDoc	160
		\PDFForm@Check	2996, 3028, 3230

