

The `hyperref-generic` module

A generic driver for `hyperref`

The L^AT_EX Project*

Version 0.97a, released 2026-04-21

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

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

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

```
\RequirePackage{pdfmanagement} %loads the PDF management and activates it
\SetKeys[document/metadata]
{
  %% options, e.g., pdf version:
  pdfversion=1.7,
}
```

If a special backend is needed it can be passed as option:

```
\RequirePackage[backend=dvipdfmx]{pdfmanagement}
```

The `pdfmanagement` is also loaded if `\DocumentMetadata` is used e.g. for tagging

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

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

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

1 Avoiding transition problems

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

- Package options are processed at the end of the driver, Class options are ignored. But not every option already works as package options, in some cases `hyperref` interferes. So it is recommended for most options—with the exception of a few mentioned below in section 9—to set them in `\hypersetup`, not as package option.
- This driver uses the `l3color` module for the colors. All colors defined with `\color_set:mn` 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 it can be passed with `\DocumentMetadata` or the `document/metadata` key:

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

or

```
\RequirePackage{pdfmanagement}  
\SetKeys[document/metadata]{lang=de-DE}
```

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

- The code supports two new keys now to set the reservation of rights relative to text & data mining (TDM): `pdftdmreservation` which should be either 0 or 1, and `pdftdmpolicy` which should be an url pointing to the policy, see <https://pdfa.org/resource/xmp-extension-schema-templates/#tdmrep>.

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 `pdfLATEX` 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 file names don't work at all with a simple file name specification GoToR links now use a full filespec dictionary. This works better, but still no every PDF viewer support this correctly. on various system.

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

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

6.2 Splits

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

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

For this reason three new commands have been create:

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

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

6.3 Options

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

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

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

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

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

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

Some main changes are

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

7.1 Background information

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

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

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

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

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

While link colors have been traditionally more or less under the control of `hyperref`, the situation with other format options, like the font, is more complicated. The font

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

7.2 New Keys

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

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

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

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

The default is `phelype`.

7.3 Public interfaces

```
\l_hyp_annot_colorlink_bool  
\l_hyp_annot_colorurl_bool  
\l_hyp_annot_colorfile_bool  
\l_hyp_annot_colorrurl_bool  
\l_hyp_annot_colormenu_bool  
\l_hyp_annot_ocgcolorlink_bool  
\l_hyp_annot_ocgcolorurl_bool  
\l_hyp_annot_ocgcolorfile_bool  
\l_hyp_annot_ocgcolorrurl_bool  
\l_hyp_annot_ocgcolormenu_bool
```

These boolean are used by the `colorlinks` and `ocgcolorlinks` and related keys. These keys insert hook code in the `pdfannot/link/<type>/begin` and `pdfannot/link/<type>/end` hooks. `<type>` is one of `GoTo`, `URI`, `GoToR`, `Named` or `Launch`.

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

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

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

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

7.4 Changed behaviour

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

Color keys accept the following input syntax:

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

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

frenchlinks The option `frenchlinks` does nothing at all.

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

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

8 PDF strings

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

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

The new method is under heavy development!

Important differences here are

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

9 Package options from hyperref

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

Options that still must be set as package options are

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

- `psdextra` this loads some extra definitions used by `\pdfstringdef`. The new driver uses `\pdfstringdef` only for the bookmarks, for other strings it is not relevant.

Options that can be without problems set as package options are

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

Ignored options:

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

10 Disabling links

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

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

`nested-links` (*setup key*)

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

11 Draftmode

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

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

12 Dropped options

A number of options are ignored by this driver

pdfversion The `pdfversion` should be set in `\DocumentMetadata` or with `\SetKeys[document/metadata`

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` or `\SetKeys[document/metadata]`:

```
\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 `cmymk` (four numbers). While the PDF reference allows four numbers, PDF readers don't necessarily handle this correctly, so the value can be wrong.

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

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

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

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

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

will create

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

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

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

urlborderstyle (*setup key*)

fileborderstyle (*setup key*)

runborderstyle (*setup key*)

menuborderstyle (*setup key*)

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

File I

hyperref-generic driver implementation

```

1 <@=hyp>
2 <*headertestphase>
3 \ProvidesFile{hgeneric-testphase.def}[2026-04-21 v0.97a %
4   generic Hyperref driver for the LaTeX PDF management bundle]
5 </headertestphase>
6 <*header>
7 \ProvidesFile{hgeneric.def}[2026-04-21 v0.97a %
8   generic Hyperref driver for the LaTeX PDF management bundle]
9 </header>
10 <*package>
11 \RequirePackage{etoolbox} %why?

```

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

```

12 \long\def\Hy@ReturnAfterFi#1\fi{\fi#1}
13 \ExplSyntaxOn
14 \file_input:n {hyperref-colorschemes.def}
15 \ExplSyntaxOff

```

1 messages

Redirect the message name:

```

16 \ExplSyntaxOn
17 \prop_gput:Nnn \g_msg_module_name_prop { hyp }{ hyperref }

```

At first a message for the testing of the resource management

```
18 \msg_new:nnnn
19 { hyp }
20 { missing-resource-management }
21 { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
22 {
23   Activate~it~with \\
24   \tl_to_str:n{\DocumentMetadata{<options>}}\\
25   before~\tl_to_str:n{\documentclass} or \\
26   \RequirePackage{pdfmanagement}
27 }
```

The pdfversion should be set in \DocumentMetadata or \SetKeys[document/metadata]

```
28 \msg_new:nnnn
29 { hyp }
30 { pdfversion-disabled }
31 {
32   This~hyperref~driver~ignores~the~pdfversion~key!\\
33   Set~the~pdfversion~in~\token_to_str:N \DocumentMetadata\\or
34   with \tl_to_str:n{\SetKeys[document/metadata]}
35 }
36 {
37   For example:\\
38   \tl_to_str:n
39   {
40     \DocumentMetadata { pdfversion=1.7 }
41   }
42 }
```

A generic message for ignored keys.

```
43 \msg_new:nnn
44 { hyp }
45 { key-dropped }
46 {
47   This~hyperref~driver~ignores~the~key~#1!\\
48   Please~check~the~documentation.
49 }
```

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

```
50 \msg_new:nnn
51 { hyp }
52 { pdfa-no-push-button }
53 { PDF/A:~Push~button~with~JavaScript~is~prohibited }
54
55 \msg_new:nnn
56 { hyp }
57 { pdfa-no-reset-button }
58 { PDF/A:~Reset~action~is~prohibited }
```

pdf/A message for not allowed Named actions

```
59 \msg_new:nnn
60 { hyp }
61 { pdfa-no-named-action }
62 { PDF/A:~Named~action~#1~is~prohibited }
```


A message if the destination name is empty.

```
63 \msg_new:nnn
64 { hyp }
65 { empty-destination-name }
66 {
67   Empty-destination~name,\
68     using~'#1'
69 }
```

A message if the destination check fails

```
70 \msg_new:nnn
71 { hyp }
72 { invalid-destination-value }
73 {
74   Invalid~value~'#1'~of~'#2'  \
75   is~replaced~by~'Fit'~\msg_line_context:.
76 }
```

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

```
77 \msg_new:nnn
78 { hyp }
79 { ignore-deprecated-or-unknown-option-in-pdf-version }
80 {
81   Option~'#1'~is~unknown~or~deprecated~in\
82   pdf~version~#2.~Ignored.
83 }
84 \msg_new:nnn
85 { hyp }
86 { ignore-deprecated-or-unknown-value-in-pdf-version }
87 {
88   Value~'#1'~is~unknown~or~deprecated~in\
89   pdf~version~#2.~Ignored.
90 }
91 \msg_new:nnn
92 { hyp }
93 { replace-deprecated-or-unknown-value-in-pdf-version }
94 {
95   Value~'#1'~is~unknown~or~deprecated~in\
96   pdf~version~#2. Value~'#3'~is used instead.
97 }
```

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:

```
98 \msg_new:nnn
99 { hyp }
100 { unknown-key }
101 {
102   unknown~key~#2~of~module~'#1'~set~to~'#3'.
103 }
104 \msg_new:nnn
105 { hyp }
106 { unknown-key-to-Hyp }
107 {
108   ignored~in~family~Hyp~unknown~key~#1.
109 }
```

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

```

110 \cs_new:Npn \__hyp_clist_display:n #1 {*~#1\}
111 \msg_new:nnn
112 { hyp }
113 { unknown-choice }
114 {
115   Value-'#3'-is-invalid-for-key-'#1'.\}
116   The-key-accepts-only-the-choices\}
117   \clist_map_function:nN { #2 }\__hyp_clist_display:n
118 }
119
120 \msg_new:nnn
121 { hyp }
122 { unknown-choice+empty }
123 {
124   Value-'#3'-is-invalid-for-key-'#1'.\}
125   The-key-accepts-only-the-choices\}
126   \clist_map_function:nN { #2 }\__hyp_clist_display:n
127   An-empty-value-removes-the-setting.
128 }
129
130 \msg_new:nnn
131 { hyp }
132 { no-bool }
133 {
134   Value-'#2'-is-invalid-for-key-'#1'.\}
135   The-key-accepts-only-the-choices\}
136   *~true\}
137   *~false \}
138   *~and-an-empty-value-which-removes-the-setting.\}
139   No-value-is-equivalent-to-using-'true'.
140 }

```

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

```

141 \msg_new:nnn
142 { hyp }
143 { empty-info-value }
144 {
145   Empty-value-for-key-#1.\}
146   This-isn't-honored-by-all-backends.
147 }

```

2 Variants

```

148 \cs_generate_variant:Nn\pdf_destination:n {nf}
149 \cs_generate_variant:Nn\pdf_object_ref:n {e}
150 \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

```

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

```

```

152 %\cs_set_protected:Npn \PDF@SetupDoc{}
153 %\PDF@FinishDoc{}% dummy needed for hyperref ...

```

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

\hypercalcbp

```

154 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n

```

(End of definition for \hypercalcbp. This function is documented on page 19.)

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

```

155 \providecommand\@pdfborder{}
156 \providecommand\@pdfborderstyle{}
157 \newcommand\OBJ@OCG@view {} % needed in hyperref
158 \def\Hy@numberline#1{#1\c_space_tl} %needed by bookmark

```

The pdfversion should be set in \DocumentMetadata or \SetKeys [document/metadata] but we must copy it to the hyperref command:

```

159 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
160 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:
161 \legacy_if:nT { Hy@setpdfversion }
162 {
163   \msg_warning:nn { hyp }{ pdfversion-disabled }
164 }
165 \Hy@DisableOption{pdfversion}

```

\Acrobatmenu should use the new internal link command

```

166 \RenewDocumentCommand \Acrobatmenu { m m }
167 {
168   \hyper@linknamed {#1} {#2}
169 }

```

\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!

```

170 % TODO should go at some time ...
171 % \kv@set@family@handler{Hyp}
172 % { \msg_warning:nne {hyp}{unknown-key-to-Hyp}{#1} }
173 \cs_set_protected:Npn \hypersetup #1
174 {
175   %\kvsetkeys{Hyp} {#1}
176   \keys_set:nn { hyp }{ #1 }
177 }
178 % TODO for now unknown keys should only give warnings.
179 \keys_define:nn { hyp }
180 {
181   unknown .code:n =
182   {
183     \msg_warning:nneee { hyp } { unknown-key }
184     { hyp }{ \l_keys_key_str } { #1 }
185   }
186 }

```

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.

```

187 \keys_define:nn { hyp }
188 {
189   next-anchor .code:n =
190   {
191     \AddToHookNext{__hyp/dest/make}
192     {\Hy@MakeCurrentHref{#1}}
193   }
194 }

```

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`)

```

195
196 \bool_new:N \l__hyp_href_url_encode_bool
197 \bool_new:N \l__hyp_href_url_ismap_bool
198 \tl_new:N \l__hyp_href_url_protocol_tl
199 \tl_new:N \l__hyp_href_pdf_destination_tl
200 \tl_new:N \l__hyp_href_pdf_page_tl
201 \tl_new:N \l__hyp_href_run_parameter_tl
202 \cs_new_protected:Npn \__hyp_href_url_format: {\begingroup\Url}
203
204
205 \keys_define:nn { hyp / href }
206 {
207   ,urlencode .bool_set:N = \l__hyp_href_url_encode_bool
208   ,format .code:n = { \cs_set:Nn \__hyp_href_url_format: {#1} },
209   ,protocol .tl_set:N = \l__hyp_href_url_protocol_tl
210   ,destination .tl_set:N = \l__hyp_href_pdf_destination_tl
211   ,pdfremotestartview .code:n =
212   {
213     \keys_set:nn { hyp }
214     { pdfremotestartview = #1 }
215   }
216   ,page .code:n =
217   {
218     \tl_set:Nn \l__hyp_href_pdf_page_tl {#1}
219     \tl_set:Nn \Hy@href@page {#1}
220   }
221   ,ismap .bool_set:N = \l__hyp_href_url_ismap_bool
222   ,run-parameter .tl_set:N = \l__hyp_href_run_parameter_tl
223   ,nextactionraw .code:n =
224   { %perhaps some safety match later, see hyperref code
225     \tl_if_empty:nTF {#1}
226     {
227       \pdfdict_remove:nn{l_hyp/annot/A}{Next}
228     }
229     {
230       \pdfdict_put:nnn{l_hyp/annot/A}{Next}{#1}
231       \tl_set:Nn \Hy@href@nextactionraw {/Next~#1}
232       \keys_set:nn {hyp }{ pdfnewwindow = true}

```

```

233     }
234   }
235   ,afrelationship .code:n =
236   {
237     \pdfdict_put:nne
238     { l_pdffile/Filespec}{AFRelationship}{ \pdf_name_from_unicode_e:n {#1}}
239   }
240
241 }
242
243 \keys_define:nn { hyp }
244 {
245   ,href / urlencode .bool_set:N = \l__hyp_href_url_encode_bool
246   ,href / urlencode .default:n = {true}
247   ,href / urlencode .initial:n = {false}
248   ,href / protocol .tl_set:N = \l__hyp_href_url_protocol_tl
249   ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
250   ,href / format .code:n = { \cs_set:Nn \_hyp_href_url_format:{#1} }
251 }
252
253 \hook_new_pair:nn{cmd/href/before}{cmd/href/after}
254
255 \DeclareRobustCommand*{\href}[1][ ]{%
256   \mode_leave_vertical:
257   \hook_use:n{cmd/href/before}
258   \group_begin:
259   \keys_set:nn { hyp / href } {#1}
260   \bool_if:NTF \l__hyp_href_url_encode_bool
261   {
262     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
263   }
264   {
265     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
266   }
267   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
268 }
269
270 \begingroup
271 \catcode`\$=6 %
272 \catcode`\#=12 %
273 \gdef\href@\$1{\expandafter\href@split$1##\}%
274 \gdef\href@split$1#$2#$3\\$4{%
275   \hyper@@link{$1}{$2}{$4}%<---_hyp-docstrip doubling!
276   \endgroup
277   \hook_use:n{cmd/href/after}
278 }%
279 \endgroup
280
281 \hook_new_pair:nn{cmd/hrefurl/before}{cmd/hrefurl/after}
282
283 \DeclareRobustCommand*{\hrefurl}[1][ ]
284 {
285   \mode_leave_vertical:
286   \hook_use:n{cmd/href/before}

```

```

287 \group_begin:
288 \keys_set:nn { hyp / href } {#1}
289 \bool_if:NTF \l__hyp_href_url_encode_bool
290 {
291   \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
292 }
293 {
294   \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
295 }
296 \hyper@normalise\__hyp_href_url_aux:nn}
297
298 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
299 {
300   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
301   \group_end:
302   \hook_use:n{cmd/href/after}
303 }
304
305 \hook_new_pair:nn{cmd/hrefpdf/before}{cmd/hrefpdf/after}
306 \DeclareRobustCommand*{\hrefpdf}[1] []
307 {
308   \mode_leave_vertical:
309   \hook_use:n{cmd/hrefpdf/before}
310   \group_begin:
311   \keys_set:nn { hyp / href } {#1}
312   \hyper@normalise\__hyp_href_pdf_aux:nn
313 }
314
315 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
316 {
317   \exp_args:Nno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
318   \group_end:
319   \hook_use:n{cmd/hrefpdf/after}
320 }
321
322 \hook_new_pair:nn{cmd/hrefrun/before}{cmd/hrefrun/after}
323 \DeclareRobustCommand*{\hrefrun}[1] []
324 {
325   \mode_leave_vertical:
326   \hook_use:n{cmd/hrefrun/before}
327   \group_begin:
328   \keys_set:nn { hyp / href } {#1}
329   \hyper@normalise\__hyp_href_run_aux:nn
330 }
331
332 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
333 {
334   \exp_args:Nno\hyper@linklaunch{#1}{#2}{\l__hyp_href_run_parameter_tl}
335   \group_end:
336   \hook_use:n{cmd/hrefrun/after}
337 }
338
339
340 \hook_new_pair:nn{cmd/url/before}{cmd/url/after}

```

```

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

```

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

```

367
368 \def\urldef#1#2{\begingroup\def\url{\hyper@normalise\url@}\setbox\z@\hbox\bgroup
369   \def\Url@HyperHook##1\endgroup{\Url@def{#1}{#2}}%
370   % Because hyperref breaks \urldef and does not define its own (Grrrr!)...
371   \def\url@##1{\egroup\endgroup\DeclareRobustCommand#1{#2{##1}}}%
372   #2}
373

```

make the new commands compatible with `\pdfstringdef`:

```

374 \NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
375 \pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
376 \pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}
377 \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.

```

378 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
379 {
380   %\tl_set:cn {#1}{#2}

```

```

381     \AddToDocumentProperties[hyperref]{#1}{#2}
382   }
383 \cs_generate_variant:Nn \_hyp_store_metadata:nn {en,ne,ee,no,eo}

```

4.2 citecolor

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

```

384 \hook_new:n{hyp/link/cite}
385 %\color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
386 %\color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
387 \keys_define:nn { hyp }
388   {
389     ,citecolor .code:n = {\_hyp_color_set:ne {hyp/color/cite}{#1}\_hyp_citecolor_hook_init:}
390     ,citebordercolor
391       .code:n = {\_hyp_color_set:ne {hyp/color/citeborder}{#1}\_hyp_citebordercolor_hook_init:}
392   }
393 \cs_new_protected:Npn \_hyp_citecolor_hook_init:
394   {
395     \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
396     {
397       \keys_set:nn { hyp }
398       {
399         linkcolor      = hyp/color/cite
400       }
401     }
402 \cs_gset_eq:NN \_hyp_citecolor_hook_init: \prg_do_nothing:
403   }
404 \cs_new_protected:Npn \_hyp_citebordercolor_hook_init:
405   {
406     \hook_gput_code:nnn { hyp/link/cite }{hyp/citeborder}
407     {
408       \keys_set:nn { hyp }
409       {
410         linkbordercolor = hyp/color/citeborder
411       }
412     }
413 \cs_gset_eq:NN \_hyp_citebordercolor_hook_init: \prg_do_nothing:
414   }
415

```

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

```

416 \bool_lazy_and:nnF
417   { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
418   { \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

```

419 \AddToHook { package/bookmark/after}
420 {
421   \define@key{BKM}{color}
422   {
423     \tl_if_blank:nTF {#1}
424     { \cs_set_eq:NN\BKM@color\@empty }
425     {
426       \__hyp_color_set:ne {__hyp/tmpa}{#1}
427       \color_export:nVN
428         {__hyp/tmpa}
429         \g__hyp_bordercolormodel_str
430         \BKM@color
431     }
432   }
433   \keys_define:nn {hyp}
434   {
435     bookmarksnumbered .legacy_if_set:n = {BKM@numbered}
436     ,bookmarksopen .code:n = {\bookmarksetup{open=#1}}
437     ,bookmarksopen .default:n = true
438     ,bookmarksdepth .code:n = {\bookmarksetup{depth=#1}}
439     ,bookmarksopenlevel .code:n = {\bookmarksetup{openlevel=#1}}
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 %
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

```

\_hyp_tmpa_tl
\_hyp_tmpb_tl
\_hyp_tmpa_seq
\_hyp_tmpa_int
\_hyp_tmpa_box
\_hyp_tmpa_str
464 \box_new:N \_hyp_tmpa_box
465 \tl_new:N \_hyp_tmpa_tl
466 \tl_new:N \_hyp_tmpb_tl
467 \seq_new:N \_hyp_tmpa_seq
468 \int_new:N \_hyp_tmpa_int
469 \str_new:N \_hyp_tmpa_str

```

(End of definition for _hyp_tmpa_tl and others.)

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

TODO: document and check use!

```

\_hyp_dest_name_tmpa_tl
\_hyp_uri_tmpa_tl
\_hyp_filename_tmpa_tl
\_hyp_text_tmpa_str \g_hyp_text_tmpa_str
470 \tl_new:N \_hyp_dest_name_tmpa_tl
471 \tl_new:N \_hyp_uri_tmpa_tl
472 \tl_new:N \_hyp_filename_tmpa_tl
473 \tl_new:N \_hyp_para_tmpa_tl
474 \str_new:N \_hyp_text_tmpa_str
475 \str_new:N \g_hyp_text_tmpa_str

```

(End of definition for _hyp_dest_name_tmpa_tl and others.)

7.2 Constants

This variable is used if a destination name is empty.

```

476 \tl_const:Nn \c_hyp_dest_undefined_tl {UNDEFINED}

```

(End of definition for \c_hyp_dest_undefined_tl.)

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

```

\_hyp_annot_types_seq
\_hyp_map_annot_hyp_prop
\_hyp_map_hyp_annot_prop
477 \seq_const_from_clist:Nn \c_hyp_annot_types_seq
478 {url,link,file,menu,run}
479 \prop_const_from_keyval:Nn \c_hyp_map_annot_hyp_prop
480 {
481   URI = url,
482   GoTo = link,
483   GoToR = file,
484   Named = menu,

```

```

485     Launch= run
486   }
487 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
488   {
489     url    = URI,
490     link   = GoTo,
491     file   = GoToR,
492     menu   = Named,
493     run    = Launch
494   }
495

```

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

7.3 Variables

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

```

\g__hyp_dest_pdfstartpage_tl
\g__hyp_dest_pdfstartview_tl
\l__hyp_dest_pdfremotestartview_tl

```

```

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

```

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

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

```

\l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_info_print_tl
499 \tl_new:N \l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_dest_tl
500 \tl_new:N \l__hyp_text_enc_info_print_tl
\l__hyp_text_enc_dest_print_tl
501 \tl_new:N \l__hyp_text_enc_dest_tl
\l__hyp_text_enc_file_print_tl
502 \tl_new:N \l__hyp_text_enc_dest_print_tl
\l__hyp_text_enc_para_print_tl
503 \tl_new:N \l__hyp_text_enc_file_print_tl
504 \tl_new:N \l__hyp_text_enc_para_print_tl
505
506 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
507 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
508 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
509 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
510 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
511 \tl_set:Nn \l__hyp_text_enc_para_print_tl {utf8/string}

```

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

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

```

512 \tl_new:N\l__hyp_link_Contents_tl
513 \socket_new:nn {hyp/link/GoTo/Contents}{2}
514 \socket_new:nn {hyp/link/URI/Contents}{1}
515 \socket_new:nn {hyp/link/GoToR/Contents}{1}

```

```

516 \socket_assign_plug:nn{hyp/link/GoTo/Contents}{noop}
517 \socket_assign_plug:nn{hyp/link/GoToR/Contents}{noop}
518 \socket_assign_plug:nn{hyp/link/URI/Contents}{noop}
519 \socket_new_plug:nnn {hyp/link/GoTo/Contents}{default}
520 {
521   \__hyp_text_pdfstring:noN
522   { Go~to~destination~#1 }
523   { \l__hyp_text_enc_info_print_tl }
524   \l__hyp_link_Contents_tl
525   \pdfannot_dict_put:nne {link/GoTo}{Contents}
526   {\l__hyp_link_Contents_tl}
527 }
528 \socket_new_plug:nnn {hyp/link/GoToR/Contents}{default}
529 {
530   \__hyp_text_pdfstring:noN
531   { Open~file~#1 }
532   { \l__hyp_text_enc_info_print_tl }
533   \l__hyp_link_Contents_tl
534   \pdfannot_dict_put:nne {link/GoToR}{Contents}
535   {\l__hyp_link_Contents_tl}
536 }
537 \socket_new_plug:nnn {hyp/link/URI/Contents}{default}
538 {
539   \__hyp_text_pdfstring:noN
540   { #1 }
541   { \l__hyp_text_enc_info_print_tl }
542   \l__hyp_link_Contents_tl
543   \pdfannot_dict_put:nne {link/URI}{Contents}
544   {\l__hyp_link_Contents_tl}
545 }
546 \pdfmeta_standard_family:nn{UA}
547 {
548   \pdfmeta_standard_verify:nF{link_Contents}
549   {
550     \socket_assign_plug:nn{hyp/link/GoTo/Contents}{default}
551     \socket_assign_plug:nn{hyp/link/GoToR/Contents}{default}
552     \socket_assign_plug:nn{hyp/link/URI/Contents}{default}
553   }
554 }

```

`\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.

```
555 \tl_new:N \l__hyp_dest_pdfview_tl
```

(End of definition for `\l__hyp_dest_pdfview_tl`.)

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

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

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

`\g__hyp_bordercolormodel_str` (*color name*)

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

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

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

```
556 \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.

```
557 \seq_map_inline:Nn \c__hyp_annot_types_seq
558 {
559   \bool_new:c {l_hyp_annot_color#1_bool}
560 }
```

(End of definition for `\l_hyp_annot_colorlink_bool` and others. These variables are documented on page 8.)

`\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.

```
561 \seq_map_inline:Nn \c__hyp_annot_types_seq
562 {
563   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
564 }
```

(End of definition for `\l_hyp_annot_ocgcolorlink_bool` and others. These variables are documented on page 8.)

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

```
565 \seq_map_inline:Nn \c_pdfannot_link_types_seq
566 {
567   \bool_new:c {l__hyp_annot_#1_bool}
568   \bool_set_true:c {l__hyp_annot_#1_bool}
569 }
```

(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.

```
570 \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.

```
571 \regex_const:Nn \c__hyp_dest_startview_regex
572 {
573   \A\ *
574   (?
575     (?XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
576     |
577     (?Fit\b|FitB\b)
578     |
579     (?:\bFitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1}
580     |
581     (?FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
582   )
583 }
```

(End of definition for `\c__hyp_dest_startview_regex`.)

7.7 PDF dictionaries

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

```
584 \pdfdict_new:n {l__hyp_page/Trans}
585 \pdfdict_put:nnn {l__hyp_page/Trans}{Type}{/Trans}
```

(End of definition for `l__hyp_page/Trans`.)

8 PDF string conversion

This is now done with `\pdf_purify:nN` which is defined and documented in `l3pdftools`.

After the content is purified the last step has to convert the string to a PDF encoding. This should be done with `cspdf_string_from_unicode:nnN` in `l3pdftools` and typically uses one of these methods:

<code>utf8/string</code>	<code>(lit)</code> (<code>utf8/string</code>)
<code>utf8/string-raw</code>	<code>lit</code> (<code>utf8/string</code>)
<code>utf8/URI</code>	(percent encoded url)
<code>utf8/URI-raw</code>	percent encoded url
<code>utf16/hex</code>	<HEX> (<code>utf16/hex</code>)
<code>utf16/hex-raw</code>	HEX (<code>utf16/hex</code>)
<code>utf16/string</code>	(lit) (<code>utf16/string</code>)
<code>utf16/string-raw</code>	lit (<code>utf16/string</code>)

`__hyp_text_string_from_unicode:nN` As normally we have to duplicate the argument we define a helper command:

```
586 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
587 {
588   \pdf_string_from_unicode:nVN { #1 } #2 #2
589 }
```

(End of definition for `__hyp_text_string_from_unicode:nN`.)

This command combines the two parts: #1=input, #2= handler shortcut #3= output str variable

`__hyp_text_pdfstring:nnN`

```
590 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
591 {
592   \pdf_purify:nN {#1}#3
593   \pdf_string_from_unicode:nVN { #2 } #3 #3
594 }
595 \cs_generate_variant:Nn \__hyp_text_pdfstring:nnN {noN,eoN}
596
```

(End of definition for `__hyp_text_pdfstring:nnN`.)

!!! temporary until all instances are gone

```
597 \cs_new_protected:Npn \Hy@pstringdef #1 #2
598 { \__hyp_text_pdfstring:nnN {#2} {utf8/string-raw}#1 }
```

This is a special version for info keys:

```
\_hyp_text_pdfstring_info:nN
```

```
599 \cs_new_protected:Npn \_hyp_text_pdfstring_info:nN #1 #2
600   {
601     \_hyp_text_pdfstring:non { #1 }{\ \l__hyp_text_enc_info_print_tl } #2
602   }
603 \cs_generate_variant:Nn \_hyp_text_pdfstring_info:nN {eN,oN}
```

(End of definition for _hyp_text_pdfstring_info:nN.)

9 Pagelabels

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

The current implementation in hyperref/hyperref drivers:

xetex: hxdetex.def, line 80-110

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

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

dvipdfm: identical to xetex

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

luatex identical to pdftex

The code in hyperref inspects \thepage and tries to figure out the numbering system and the prefix. E.g. A-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
604 \cs_new_protected:Npn \_hyp_PageLabels_gpush:
605   {
606     \pdfmanagement_add:nne {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
607   }
608
609 \def\Hy@PutCatalog #1 {}
610
611
612 \legacy_if:nT { Hy@pdfpagelabels }
```

```

613 {
614   \cs_set_protected:Npn \HyPL@StorePageLabel #1
615     {
616       \tl_gput_right:Ne \HyPL@Labels { \the\Hy@abspage<<#1>> }
617       \__hyp_PageLabels_gpush:
618     }
619 }

```

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

10 Core Hyperref Commands

Every hyperref has to define eight core command:

```

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

```

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

10.1 Link level

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

`g__hyp_linknestlevel_int`

```

620 \int_new:N \g__hyp_linknestlevel_int

```

(End of definition for `g__hyp_linknestlevel_int`.)

```

621 \prg_new_conditional:Npnn \__hyp_if_outer_link: {TF}
622 {
623   \int_compare:nNnTF { \g__hyp_linknestlevel_int } > {1}
624     { \prg_return_false: }
625     { \prg_return_true: }
626 }
627 \cs_new:Npn \__hyp_check_link_nesting:TF #1 #2
628 {
629   \use_i:nn {#1}{#2}
630 }
631 \keys_define:nn { hyp }
632 {
633   nested-links .choice:,
634   nested-links / true .code:n =
635     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \use_i:nn },

```



```

636   nested-links / false .code:n =
637     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \__hyp_if_outer_link:TF },
638   nested-links .default:n = {true}
639 }

```

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 `\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
640 \Hy@WrapperDef \__hyp_destination:nn #1 #2
641 {
642   \mode_if_horizontal:T { \@savsf\spacefactor }
643   \Hy@SaveLastskip      %defined in hyperref
644   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
645   \__hyp_text_pdfstring:noN
646   { \HyperDestNameFilter{#1} }
647   { \l__hyp_text_enc_dest_tl }
648   \l__hyp_tmpa_tl
649   \str_if_eq:nnTF {#2} {fitrbox}
650   {
651     \exp_args:NV
652     \pdf_destination:nnnn \l__hyp_tmpa_tl
653     { \box_wd:N \l__hyp_dest_box }
654     { \box_ht:N \l__hyp_dest_box }
655     { \box_dp:N \l__hyp_dest_box }
656   }
657   {
658     \exp_args:NV
659     \pdf_destination:nf
660     { \l__hyp_tmpa_tl }
661     { #2 }
662   }
663   \Hy@RestoreLastskip  %defined in hyperref
664   \mode_if_horizontal:T { \spacefactor\@savsf }
665 }

```

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

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

```

\hyper@anchor
\hyper@anchorstart 666 \tl_new:N\l_hyp_current_dest_name_tl
\hyper@anchorend 667 \hook_new:n{hyp/anchor}
hyp/anchor 668 \cs_new_protected:Npn \hyper@anchor #1
\l_hyp_current_dest_name_tl 669 {
670 \exp_args:NnV
671 \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
672 \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
673 \hook_use:n{hyp/anchor}
674 }
675
676 \cs_new_protected:Npn \hyper@anchorstart #1
677 {
678 \Hy@activeanchortrue
679 \exp_args:NnV
680 \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
681 \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
682 \hook_use:n{hyp/anchor}
683 }
684
685 \cs_new_protected:Npn \hyper@anchorend
686 {
687 \Hy@activeanchorfalse
688 }

```

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

10.3 GoTo Links

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

At first the internal link commands:

```

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

```

```

708     \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
709   }
710
711 \cs_new_protected:Npn \__hyp_link_goto_end:
712   {
713     \pdfannot_link_goto_end:
714   }

```

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

```

715 \cs_set_protected:Npn \hyper@link #1 #2 #3 %#1 context, #2=destination name, #3 content
716   {
717     \bool_if:NTF \l__hyp_annot_GoTo_bool
718     {
719       \int_gincr:N\g__hyp_linknestlevel_int
720       \__hyp_check_link_nesting:TF
721       {
722         \Hy@VerboseLinkStart{#1}{#2}
723         \group_begin:
724         \tl_set:Nn \l_hyp_current_dest_name_tl {#2}

```

this socket adds something to the `/Contents` key.

```

725         \socket_use:nnn{hyp/link/GoTo/Contents}{#2}{#3}
726         \hook_use:n {hyp/link/#1}
727         \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
728         \__hyp_link_goto_end:
729         \group_end:
730         \Hy@VerboseLinkStop
731       }
732     {
733       \group_begin: #3\group_end:
734     }

```

```

735     \int_gdecr:N\g__hyp_linknestlevel_int
736   }
737   {\let\protect\relax#3}}
738 }
739 \cs_set_protected:Npn \hyper@linkstart #1 #2 %#1 context, #2=destination name
740 {
741   \bool_if:NT \l__hyp_annot_GoTo_bool
742   {
743     \int_gincr:N\g__hyp_linknestlevel_int
744     \__hyp_check_link_nesting:TF
745     {
746       \Hy@VerboseLinkStart{#1}{#2}% only for debug
747       \group_begin:
748       \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
749       \socket_use:nnn{hyp/link/GoTo/Contents}{#2}{}
750       \hook_use:n {hyp/link/#1}
751       \__hyp_link_goto_begin:nw {#2}
752     }
753     {
754       \group_begin:
755     }
756   }
757 }
758 \cs_set_protected:Npn \hyper@linkend
759 {
760   \bool_if:NT \l__hyp_annot_GoTo_bool
761   {
762     \__hyp_check_link_nesting:TF
763     {
764       \__hyp_link_goto_end:
765       \group_end:
766       \Hy@VerboseLinkStop
767     }
768     {
769       \group_end:
770     }
771     \int_gdecr:N\g__hyp_linknestlevel_int
772   }
773 }

```

10.4 URI links

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

```

774 \pdfdict_new:n {l_hyp/annot/A/URI}
775 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
776 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
777
778 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
779 {
780   \bool_if:NTF \l__hyp_annot_URI_bool
781   {
782     \int_gincr:N\g__hyp_linknestlevel_int
783     \__hyp_check_link_nesting:TF
784     {

```

```

785     \group_begin:
786     \__hyp_text_pdfstring:noN {#2}{ \l__hyp_text_enc_uri_print_tl }\l__hyp_uri_tmpa_tl
787     \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
788     \bool_if:NT \l__hyp_href_url_ismap_bool
789     {
790         \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
791     }

```

This socket adds something to the /Contents key.

```

792     \socket_use:nn{hyp/link/URI/Contents}{#2}
793     \cs_set_eq:NN \# \c_hash_str
794     \cs_set_eq:NN \% \c_percent_str
795     \Hy@safe@activestrue
796     \mode_leave_vertical:
797     \pdfannot_dict_put:nne {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
798     \pdfannot_link:nen { URI }
799     {
800     }
801     {
802         \let\protect\relax
803         #1
804         \Hy@xspace@end
805         \Hy@VerboseLinkStop %where is the start??
806     }
807     \group_end:
808     }
809     {
810         \group_begin: #1 \group_end:
811     }
812     \int_gdecr:N\g__hyp_linknestlevel_int
813     }
814     {\let\protect\relax#1}}
815 }
816

```

10.5 GoToR Links files

```

817 \pdfdict_new:n {l_hyp/annot/A/GoToR}
818 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
819 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
820
821 \cs_generate_variant:Nn \pdffile_embed_file:nnn {noe}
822 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
823 {
824     \bool_if:NTF \l__hyp_annot_GoToR_bool
825     {
826         \int_gincr:N\g__hyp_linknestlevel_int
827         \__hyp_check_link_nesting:TF
828         {
829             \group_begin:
830             \tl_set:Ne \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
831             \exp_args:Ne
832             \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
833             {

```

```

834         \pdfdict_put:nne { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n
835         \pdffile_embed_file:noe
836         {}
837         {\l__hyp_filename_tmpa_tl }
838         {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
839     }
840 \pdfdict_put:nne
841 {l_hyp/annot/A/GoToR}
842 {F}
843 {\pdf_object_ref:e {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
844 \__hyp_text_pdfstring:noN
845 { #3 }
846 { \l__hyp_text_enc_dest_print_tl }
847 \l__hyp_dest_name_tmpa_tl

```

This socket adds something to the /Contents key.

```

848 \socket_use:nn{hyp/link/GoToR/Contents}{#2}
849 \tl_if_blank:eTF {#3}
850 {
851     \pdfdict_put:nne {l_hyp/annot/A/GoToR}{D}
852     {
853     [
854         \int_eval:n
855         { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
856         /\l__hyp_dest_pdfremotestartview_tl
857     ]
858     }
859 }
860 {
861     \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
862 }
863 \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>

```

864 \pdf_object_unnamed_write:ne{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
865 \pdfannot_dict_put:nne {link/GoToR}{A}{\pdf_object_ref_last:}
866 \pdfannot_link:nnn %expansion??
867 { GoToR }
868 {
869 }
870 {
871     \let\protect\relax
872     #1\Hy@xspace@end
873     \Hy@VerboseLinkStop %where is the start??
874 }
875 \group_end:
876 }
877 {
878     \group_begin: #1 \group_end:
879 }
880 \int_gdecr:N\g__hyp_linknestlevel_int
881 }
882 {{\let\protect\relax#1}}
883 }

```

10.6 Launch links

We define `\hyper@linklaunch` for naming consistency

```

884 \pdfdict_new:n {l_hyp/annot/A/Launch}
885 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
886 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
887
888 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
889 {
890   \bool_if:NTF \l__hyp_annot_Launch_bool
891   {
892     \int_gincr:N\g__hyp_linknestlevel_int
893     \__hyp_check_link_nesting:TF
894     {
895       \group_begin:
896       \__hyp_text_pdfstring:noN
897       { #1 }
898       { \l__hyp_text_enc_file_print_tl }
899       \l__hyp_filename_tmpa_tl
900       \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
901       \__hyp_text_pdfstring:noN
902       { #3 }
903       { \l__hyp_text_enc_para_print_tl }
904       \l__hyp_para_tmpa_tl
905       \bool_if:nTF
906       {
907         \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
908         ||
909         \pdf_version_compare_p:Nn > {1.9}
910       }
911       {
912         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
913       }
914       {
915         \pdfdict_put:nne
916         {l_hyp/annot/A/Launch}
917         {Win}
918         {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
919       }
920       \mode_leave_vertical:
921       \pdfannot_dict_put:nne {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}
922       \pdfannot_link:nen
923       { Launch }
924       {
925         % /A
926         % <<
927         % \pdfdict_use:n {l_hyp/annot/A/Launch}
928         % >>
929       }
930       {
931         \let\protect\relax
932         #2\Hy@xspace@end
933         \Hy@VerboseLinkStop %where is the start??
934       }

```

```

935     \group_end:
936     }
937     { \group_begin: #2 \group_end: }
938     \int_gdecr:N\g__hyp_linknestlevel_int
939   }
940   {\let\protect\relax#2}}
941 }

```

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

```

942 \use:e
943   { % filename, anchor text, linkname
944     \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str #1 \exp_not:N \ \ #2 #3
945   }
946   {
947     \hyper@linklaunch {#1}{#2}{#3}
948   }

```

10.7 Named links (menu)

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

```

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

```



```

982         #2
983         }
984         \group_end:
985     }
986     { \group_begin: #2 \group_end: }
987     \int_gdecr:N\g__hyp_linknestlevel_int
988 }
989 {\let\protect\relax#2}}
990 }
991

```

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.

```

992 \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`.

```

993 \cs_new_protected:Npn \__hyp_color_select:n #1
994 {
995     \tl_if_head_eq_charcode:nNTF {#1}[ %]
996     {
997         \__hyp_color_select_aux:wn #1
998     }
999     {
1000         \color_select:n {#1}
1001     }
1002 }
1003
1004 \cs_new_protected:Npn \__hyp_color_select_aux:wn [#1] #2
1005 {
1006     \color_select:nn {#1}{#2}
1007 }
1008
1009 \cs_generate_variant:Nn \__hyp_color_select:n {e}

```

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

```

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

```

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

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

```

1010 \cs_new_protected:Npn \_hyp_color_set:nn #1 #2
1011 {
1012   \tl_if_head_eq_charcode:nNTF {#2}[ %]
1013   {
1014     \_hyp_color_set_aux:nwn { #1 } #2
1015   }
1016   {
1017     \color_set:nn {#1} {#2}
1018   }
1019 }
1020
1021 \cs_new_protected:Npn \_hyp_color_set_aux:nwn #1 [#2] #3
1022 {
1023   \color_set:nnn {#1}{#2}{#3}
1024 }
1025
1026 \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.

```

1027 \prop_map_inline:Nn \c_hyp_map_hyp_annot_prop
1028 {
1029   \hook_gput_code:nnn
1030   {pdfannot/link/#2/begin}
1031   {hyp/color}
1032   {
1033     \bool_if:cT { l_hyp_annot_color#1_bool }
1034     {
1035       \group_begin:
1036       \color_select:n { hyp/color/#1}
1037     }
1038   }
1039   \hook_gput_code:nnn
1040   {pdfannot/link/#2/end}
1041   {hyp/color}
1042   {
1043     \bool_if:cT { l_hyp_annot_color#1_bool }
1044     {
1045       \group_end:
1046     }
1047   }
1048 }

```

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

```

1049 \keys_define:nn { hyp }
1050 {

```

```

1051     ,colorlinks .choice:
1052     ,colorlinks / true .meta:n =
1053     {
1054         ,pdfborder={0~0~0}
1055         ,pdfborderstyle=
1056         ,colorurl =#1
1057         ,colorlink =#1
1058         ,colorrun =#1
1059         ,colormenu =#1
1060         ,colorfile =#1
1061     }
1062     ,colorlinks / false .meta:n =
1063     {
1064         ,colorurl =#1
1065         ,colorlink =#1
1066         ,colorrun =#1
1067         ,colormenu =#1
1068         ,colorfile =#1
1069     }
1070     ,colorlinks .default:n = {true}
1071 }

```

```

colorurl (setup key)
colorlink (setup key) 1072 \seq_map_inline:Nn \c__hyp_annot_types_seq
colorrun (setup key) 1073 {
colormenu (setup key) 1074     \keys_define:nn { hyp }
colorfile (setup key) 1075     {
urlcolor (setup key) 1076         ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
linkcolor (setup key) 1077         ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
runcolor (setup key) 1078     }
menucolor (setup key) 1079 }
filecolor (setup key) 1080
allcolors (setup key) 1081 \keys_define:nn { hyp }
1082 {
1083     ,allcolors .meta:n =
1084     {
1085         ,urlcolor=#1
1086         ,linkcolor=#1
1087         ,runcolor=#1
1088         ,filecolor=#1
1089         ,menucolor=#1
1090     }
1091     ,allcolors .value_required:n = true
1092 }

```

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)

```

1093 \keys_define:nn { hyp }

```

```

1094 {
1095   ,bordercolormodel .choices:nn =
1096   {rgb,cmyk}
1097   { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
1098   ,bordercolormodel .initial:n ={rgb}
1099 }
1100 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1101 {
1102   \keys_define:nn { hyp }
1103   {
1104     #1bordercolor .code:n =
1105     {
1106       \tl_if_empty:nTF { ##1 }
1107       {
1108         \pdfannot_dict_remove:nn
1109         {link/#2}
1110         { C }
1111       }
1112       {
1113         \__hyp_color_set:ne {hyp/color/#1border}{##1}
1114         \color_export:nVN
1115         {hyp/color/#1border}
1116         \g__hyp_bordercolormodel_str
1117         \l__hyp_tmpa_tl
1118         \pdfannot_dict_put:nne
1119         {link/#2}
1120         { C }
1121         { [\l__hyp_tmpa_tl] }
1122       }
1123     }
1124   }
1125 }
1126
1127 \keys_define:nn { hyp }
1128 {
1129   ,allbordercolors .meta:n =
1130   {
1131     ,linkbordercolor=#1
1132     ,urlbordercolor =#1
1133     ,filebordercolor=#1
1134     ,menubordercolor=#1
1135     ,runbordercolor =#1
1136   }
1137   ,allbordercolors .value_required:n = true
1138 }
1139

```

11.3.2 Borderwidth and -arc

```

1140 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1141 {
1142   \keys_define:nn { hyp }
1143   {
1144     #1border .code:n =

```

```

1145     {
1146         \tl_if_empty:nTF { ##1 }
1147         {
1148             \pdfannot_dict_remove:nn
1149             {link/#2}
1150             { Border }
1151         }
1152         {
1153             \pdfannot_dict_put:nnn
1154             {link/#2}
1155             { Border }
1156             { [##1] }
1157         }
1158     }
1159 }
1160 }
1161 \keys_define:nn { hyp }
1162 {
1163     ,pdfborder .code:n =
1164     {
1165         \tl_if_empty:nTF { #1 }
1166         {
1167             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1168             {
1169                 \pdfannot_dict_remove:nn
1170                 {link/##2}
1171                 { Border }
1172             }
1173         }
1174         {
1175             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1176             {
1177                 \pdfannot_dict_put:nnn
1178                 {link/##2}
1179                 { Border }
1180                 { [#1] }
1181             }
1182         }
1183     }
1184     ,pdfborder .initial:n = {0~0~1},
1185 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1186 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1187 {
runborderstyle (setup key) 1188     \keys_define:nn { hyp }
fileborderstyle (setup key) 1189     {
menuborderstyle (setup key) 1190         #1borderstyle .code:n =
1191         {
1192             \tl_if_empty:nTF { ##1 }
1193             {

```

```

1194         \pdfannot_dict_remove:nn
1195         {link/#2}
1196         { BS }
1197     }
1198     {
1199         \pdfannot_dict_put:nnn
1200         {link/#2}
1201         { BS }
1202         { <<##1>> }
1203     }
1204 }
1205 }
1206 }
1207 \keys_define:nn { hyp }
1208 {
1209     ,pdfborderstyle .code:n =
1210     {
1211         \tl_if_empty:nTF { #1 }
1212         {
1213             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1214             {
1215                 \pdfannot_dict_remove:nn
1216                 {link/##2}
1217                 { BS }
1218             }
1219         }
1220         {
1221             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1222             {
1223                 \pdfannot_dict_put:nnn
1224                 {link/##2}
1225                 { BS }
1226                 { <<##1>> }
1227             }
1228         }
1229     }
1230     ,pdfborderstyle .initial:n = {},
1231 }

```

11.4 ocolorlinks

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

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

```

1232 \cs_new_protected:Npn \__hyp_ocg_init:
1233 {
1234     \pdf_object_new:n { __hyp/OCG/View }
1235     \pdf_object_new:n { __hyp/OCG/Print }
1236     \pdf_object_new:n { __hyp/OCG/config }
1237     \pdf_object_new:n { __hyp/OCG/refarray }
1238     \pdf_object_write:nne { __hyp/OCG/refarray } { array }
1239     {

```

```

1240         \pdf_object_ref:n { __hyp/OCG/View }
1241         \c_space_tl
1242         \pdf_object_ref:n { __hyp/OCG/Print }
1243     }
1244 \pdf_object_write:nnn { __hyp/OCG/View } { dict }
1245 {
1246     /Type/OCG
1247     /Name(View)
1248     /Usage
1249     <<
1250         /Print <</PrintState/OFF>>~
1251         /View <</ViewState/ON >>~
1252     >>
1253 }
1254 \pdf_object_write:nnn { __hyp/OCG/Print } { dict }
1255 {
1256     /Type/OCG
1257     /Name(Print)
1258     /Usage
1259     <<
1260         /Print <</PrintState/ON>>~
1261         /View <</ViewState/OFF>>~
1262     >>
1263 }
1264 \pdfmanagement_add:nne { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG
1265 \pdfmanagement_add:nne { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG
1266 \pdf_object_write:nne { __hyp/OCG/config } { dict }
1267 {
1268     /OFF[\pdf_object_ref:n { __hyp/OCG/Print } ]
1269     /AS[
1270     <<
1271         /Event/View
1272         /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1273         /Category[/View]
1274     >>
1275     <<
1276         /Event/Print
1277         /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1278         /Category[/Print]
1279     >>
1280     <<
1281         /Event/Export
1282         /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1283         /Category[/Print]
1284     >>
1285     ]
1286 }
1287 \pdfmanagement_add:nne { Catalog / OCProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG
1288 \cs_gset:Npn \__hyp_ocg_init: { }
1289 }

```

(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.

```

1290 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1291 {
1292   \hook_gput_code:nnn
1293     {pdfannot/link/#2/begin}
1294     {hyp/ocg}
1295     {
1296       \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1297       {
1298         \__hyp_ocg_init:
1299         \group_begin:
1300         \hbox_set:Nw \l__hyp_tmpa_box
1301       }
1302     }
1303   \hook_gput_code:nnn
1304     {pdfannot/link/#2/end}
1305     {hyp/ocg}
1306     {
1307       \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1308       {
1309         \hbox_set_end:
1310         \mbox
1311         {
1312           \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1313           \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1314           \pdf_emc:
1315           \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1316           \group_begin:
1317           \color_select:n { hyp/color/#1 }
1318           \box_use_drop:N \l__hyp_tmpa_box
1319           \group_end:
1320           \pdf_emc:
1321         }
1322         \group_end:
1323       }
1324     }
1325 }

```

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

```

ocgcolorlink (setup key) 1326 \bool_lazy_or:nnTF
  ocgcolorurl (setup key) 1327 { \pdf_version_compare_p:Nn > {1.4} }
ocgcolorfile (setup key) 1328 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
ocgcolormenu (setup key) 1329 {
  ocgcolorrund (setup key) 1330   \keys_define:nn { hyp }
1331   {
1332     ,_ocgcolorlinks .meta:n =
1333     {
1334       ocgcolorlink=#1,
1335       ocgcolorurl=#1,
1336       ocgcolorfile=#1,
1337       ocgcolorrund=#1,
1338       ocgcolormenu=#1
1339     }
1340     ,_ocgcolorlinks .default:n = true
1341   }

```



```

1342 }
1343 {
1344   \keys_define:nn { hyp }
1345   {
1346     ,_ocgcolorlinks .code:n =
1347     {
1348       \msg_warning:nnee
1349       { hyp }
1350       { ignore-deprecated-or-unknown-option-in-pdf-version }
1351       { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1352     }
1353   }
1354 }
1355
1356 \keys_define:nn { hyp }
1357 {
1358   ,ocgcolorlinks .choice:
1359   ,ocgcolorlinks / true .meta:n =
1360   {
1361     pdfborder      = {0~0~0},
1362     pdfborderstyle = {},
1363     colorlinks     = false,
1364     _ocgcolorlinks = true
1365   }
1366   ,ocgcolorlinks / false .meta:n =
1367   {
1368     _ocgcolorlinks = false
1369   }
1370   ,ocgcolorlinks .default:n = {true}
1371 }
1372
1373 \seq_map_inline:Nn \c__hyp_annot_types_seq
1374 {
1375   \bool_lazy_or:nnTF
1376   { \pdf_version_compare_p:Nn > {1.4} }
1377   { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1378   {
1379     \keys_define:nn { hyp }
1380     {
1381       ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1382     }
1383   }
1384   {
1385     \keys_define:nn { hyp }
1386     {
1387       ,ocgcolor#1 .code:n=
1388       {
1389         \msg_warning:nnee
1390         { hyp }
1391         { ignore-deprecated-or-unknown-option-in-pdf-version }
1392         { ocgcolor#1 }
1393         { \pdf_version_major:.\pdf_version_minor: }
1394       }
1395     }

```

```

1396     }
1397 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1398 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1399 {
1400   \keys_define:nn { hyp }
1401   {
1402     ,#1highlight .choices:nn =
1403     { /I, /N, /O, /P}
1404     {
1405       \pdfannot_dict_put:nnn
1406       {link/#2}
1407       { H }
1408       { ##1 }
1409     }
1410     ,#1highlight / .code:n =
1411     {
1412       \pdfannot_dict_remove:nn
1413       {link/#2}
1414       { H }
1415     }
1416     ,#1highlight / unknown .code:n =
1417     {
1418       \msg_warning:nnee { hyp } { unknown-choice+empty }
1419       { #1highlight }
1420       { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1421       { \exp_not:n {##1} }
1422     }
1423   }
1424 }
1425 }
1426 }
1427
1428
1429 \keys_define:nn { hyp }
1430 {
1431   ,pdfhighlight .choices:nn =
1432   { /I, /N, /O, /P}
1433   {
1434     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1435     {
1436       \pdfannot_dict_put:nnn
1437       {link/####2}
1438       { H }
1439       { #1 }
1440     }
1441   }
1442   ,pdfhighlight / .code:n =
1443   {
1444     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1445     {

```

```

1446         \pdfannot_dict_remove:nn
1447         {link/##2}
1448         { H }
1449     }
1450 }
1451 ,pdfhighlight .initial:n = {/I},
1452 ,pdfhighlight / unknown .code:n =
1453 {
1454     \msg_warning:nree { hyp } { unknown-choice+empty }
1455     { pdfhighlight }
1456     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1457     { \exp_not:n {#1} }
1458 }
1459 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1460 \keys_define:nn { hyp }
hideurl (setup key) 1461 {
hidefile (setup key) 1462     hidelinks .meta:n =
hiderun (setup key) 1463     {
hidemenu (setup key) 1464         ,colorlinks      = false
1465         ,ocgcolorlinks = false
1466         ,pdfborder      = { 0~0~0 }
1467         ,pdfborderstyle=
1468     }
1469 }
1470
1471 \seq_map_inline:Nn \c__hyp_annot_types_seq
1472 {
1473     \keys_define:nn { hyp }
1474     {
1475         hide#1 .meta:n =
1476         {
1477             ,color#1      = false
1478             ,ocgcolor#1  = false
1479             ,#1border    = { 0~0~0 }
1480             ,#1borderstyle =
1481         }
1482     }
1483 }

```

11.7 color schemes and settings

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

```

colorscheme (setup key)
1484 \keys_define:nn { hyp }
1485 {
1486     colorscheme .code:n =
1487     {

```

```

1488     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1489     {
1490         \keys_set:nn { hyp }
1491         {
1492             ##1 = ##2
1493         }
1494     }
1495 }
1496 }
1497 \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) 1498 \keys_define:nn { hyp }
pdfversion (setup key) 1499 {
1500     ,unicode .code:n = {}
1501     ,pdfencoding .code:n = {}
1502     ,pdfversion .code:n =
1503     {
1504         \msg_warning:nn { hyp }{ pdfversion-disabled }
1505     }
1506 }
1507 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
debug (setup key) 1508 \keys_define:nn { hyp }
draft (setup key) 1509 {
final (setup key) 1510     ,verbose .legacy_if_set:n = {Hy@verbose}
1511     ,debug .legacy_if_set:n = {Hy@verbose}
1512 }
1513 \keys_define:nn { hyp }
1514 {
1515     ,draft .code:n =
1516     {
1517         \Hy@draftrue
1518         \PassOptionsToPackage{draft}{bookmark}
1519     }
1520     ,final .code:n =
1521     {
1522         \Hy@finaltrue
1523         \PassOptionsToPackage{final}{bookmark}
1524     }
1525 }

```

```

extension (setup key)
hypertextnames (setup key) 1526 \keys_define:nn { hyp }
naturalnames (setup key) 1527 {
pageanchor (setup key) 1528 ,extension .tl_set:N = \XR@ext
linktoc (setup key) 1529 ,extension .initial:n= pdf
linktocpage (setup key) 1530 ,hypertextnames .legacy_if_set:n = {Hy@hypertextnames}
plainpages (setup key) 1531 ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
localanchorname (setup key) 1532 ,localanchorname .legacy_if_set:n = {Hy@localanchorname}
linkfileprefix (setup key) 1533 ,naturalnames .legacy_if_set:n = {Hy@naturalnames}
1534 ,pageanchor .legacy_if_set:n = {Hy@pageanchor}
1535 ,plainpages .legacy_if_set:n = {Hy@plainpages}
1536 }
1537
1538 \keys_define:nn { hyp }
1539 {
1540 ,linktoc .choices:nn = { none, section, all, page }
1541 {
1542 \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1543 }
1544 ,linktoc / unknown .code:n =
1545 {
1546 \msg_warning:nnee { hyp } { unknown-choice }
1547 { linktoc }
1548 { none, section, all, page }
1549 { \exp_not:n {#1} }
1550 }
1551 ,linktocpage .choice:
1552 ,linktocpage / true .meta:n = {linktoc=page}
1553 ,linktocpage / false .meta:n = {linktoc=section}
1554 ,linktocpage .default:n = true
1555 }
1556

link (setup key) This booleans allow to disable the link types.
url (setup key) 1557 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1558 {
menu (setup key) 1559 \keys_define:nn { hyp }
run (setup key) 1560 {
1561 ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1562 }
1563 }

1564 \keys_define:nn { hyp }
1565 {
1566 ,baseurl .code:n =
1567 {
1568 \__hyp_text_pdfstring:noN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1569 \tl_if_empty:NTF \l__hyp_tmpa_tl
1570 {
1571 \pdfmanagement_remove:nn {Catalog} { URI }
1572 }
1573 {
1574 \pdfmanagement_add:nne {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1575 }

```

```

1576     \__hyp_store_metadata:nn {baseurl}{#1}
1577   }
1578   %only false does something ...
1579   ,bookmarks .choice:
1580   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1581   ,bookmarks / true .code:n = {}
1582   ,bookmarks .default:n = {true}
1583   ,bookmarksnumbered .legacy_if_set:n = {Hy@bookmarksnumbered}
1584   ,bookmarksopen .legacy_if_set:n = {Hy@bookmarksopen}
1585   ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1586   ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1587   ,pdfcenterwindow .choice:
1588   ,pdfcenterwindow / false .code:n =
1589     {
1590       \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1591     }
1592   ,pdfcenterwindow / true .code:n =
1593     {
1594       \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1595     }
1596   ,pdfcenterwindow / .code:n =
1597     {
1598       \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1599     }
1600   ,pdfcenterwindow / unknown .code:n =
1601     {
1602       \msg_warning:nnee { hyp } { no-bool }
1603       { pdfcenterwindow }
1604       { \exp_not:n {#1} }
1605     }
1606   ,pdfcenterwindow .default:n = true
1607   ,pdfdirection .choice:
1608   ,pdfdirection / L2R .code:n =
1609     {
1610       \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1611     }
1612   ,pdfdirection / R2L .code:n =
1613     {
1614       \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1615     }
1616   ,pdfdirection / .code:n =
1617     {
1618       \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1619     }
1620   ,pdfdirection / unknown .code:n =
1621     {
1622       \msg_warning:nnee { hyp } { unknown-choice+empty }
1623       { pdfdirection }
1624       { L2R , R2L }
1625       { \exp_not:n {#1} }
1626     }
1627   ,pdfdisplaydoctitle .choice:
1628   ,pdfdisplaydoctitle / false .code:n =
1629     {

```

```

1630     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1631   }
1632 ,pdffitwindow / true .code:n =
1633   {
1634     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1635   }
1636 ,pdffitwindow .default:n = true
1637 ,pdfduplex .choices:nn =
1638   {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1639   {
1640     \pdf_version_compare:NnTF > {1.6}
1641     {
1642       \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1643         { Duplex } { /#1 }
1644     }
1645     {
1646       \msg_warning:nnee
1647         {hyp}
1648         {ignore-deprecated-or-unknown-option-in-pdf-version}
1649         {pdfduplex}
1650         {\pdf_version:}
1651     }
1652   }%
1653 ,pdfduplex / .code:n =
1654   {
1655     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Duplex }
1656   }
1657 ,pdfduplex / unknown .code:n =
1658   {
1659     \msg_warning:nneee { hyp } { unknown-choice+empty }
1660     { pdfduplex }
1661     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1662     { \exp_not:n {#1} }
1663   }
1664 ,pdffitwindow .choice:
1665 ,pdffitwindow / false .code:n =
1666   {
1667     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1668   }
1669 ,pdffitwindow / true .code:n =
1670   {
1671     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1672   }
1673 ,pdffitwindow / .code:n =
1674   {
1675     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1676   }
1677 ,pdffitwindow .default:n = true
1678 ,pdffitwindow / unknown .code:n =
1679   {
1680     \msg_warning:nnee { hyp } { no-bool }
1681     { pdffitwindow }
1682     { \exp_not:n {#1} }
1683   }

```

```

1684 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1685 ,pdflinkmargin .initial:n = {1pt}
1686 ,pdfmenubar .choice:
1687 ,pdfmenubar / true .code:n =
1688 {
1689   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1690 }
1691 ,pdfmenubar / false .code:n =
1692 {
1693   \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1694   { HideMenubar } { true }
1695 }
1696 ,pdfmenubar / .code:n =
1697 {
1698   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1699 }
1700 ,pdfmenubar .default:n = true
1701 ,pdfmenubar / unknown .code:n =
1702 {
1703   \msg_warning:nnee { hyp } { no-bool }
1704   { pdfmenubar }
1705   { \exp_not:n {#1} }
1706 }
1707 ,pdfnewwindow .choice:
1708 ,pdfnewwindow / true .code:n =
1709 {
1710   \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1711   \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1712 }
1713 ,pdfnewwindow / false .code:n =
1714 {
1715   \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1716   \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1717 }
1718 ,pdfnewwindow / .code:n =
1719 {
1720   \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1721   \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1722 }
1723 ,pdfnonfullscreenpagemode .choices:nn =
1724 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1725 {
1726   \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1727   { NonFullScreenPageMode } {/#1}
1728 }
1729 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1730 {
1731   \pdf_version_compare:NnTF < {1.6}
1732   {
1733     %message
1734   }
1735   {
1736     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1737     {NonFullScreenPageMode}{/UseAttachments}

```



```

1738     }
1739   }
1740   ,pdfnonfullscreenpagemode / .code:n =
1741   {
1742     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1743   }
1744   ,pdfnonfullscreenpagemode / unknown .code:n =
1745   {
1746     \msg_warning:nnee { hyp } { unknown-choice+empty }
1747     { pdfnonfullscreenpagemode }
1748     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1749     { \exp_not:n {#1} }
1750   }
1751   ,pdfnumcopies .code:n =
1752   {
1753     \pdf_version_compare:NnTF > {1.6}
1754     {
1755       \tl_if_empty:nTF {#1}
1756       {
1757         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1758       }
1759       {
1760         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1761         {NumCopies}{#1}
1762       }
1763     }
1764     {
1765       \msg_warning:nnee
1766       {hyp}
1767       {ignore-deprecated-or-unknown-option-in-pdf-version}
1768       {pdfnumcopies}
1769       {\pdf_version:}
1770     }
1771   }
1772   ,pdfpagelayout .choices:nn =
1773   { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1774   { \pdfmanagement_add:nne {Catalog} { PageLayout }{ /#1 } }
1775   ,pdfpagelayout / .code:n =
1776   { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1777   ,pdfpagelayout / unknown .code:n =
1778   {
1779     \msg_warning:nnee { hyp } { unknown-choice+empty }
1780     { pdfpagelayout }
1781     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1782     { \exp_not:n {#1} }
1783   }
1784   ,pdfpagemode .choices:nn =
1785   { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1786   { \pdfmanagement_add:nne {Catalog} { PageMode }{ /#1 } }
1787   ,pdfpagemode / UseAttachments .code:n =
1788   {
1789     \pdf_version_compare:NnTF > {1.5}
1790     {
1791       \pdfmanagement_add:nne {Catalog} { PageMode }{ /UseAttachments }

```

```

1792     }
1793     {
1794         \msg_warning:nnee
1795         {hyp}
1796         {ignore-deprecated-or-unknown-value-in-pdf-version}
1797         {UseAttachments}
1798         {\pdf_version:}
1799     }
1800 }
1801 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1802 ,pdfpagemode / unknown .code:n =
1803 {
1804     \msg_warning:nneee { hyp } { unknown-choice+empty }
1805     { pdfpagemode }
1806     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1807     { \exp_not:n {#1} }
1808 }
1809 ,pdfpagescrop .code:n =
1810 {
1811     \tl_if_empty:nTF {#1} %or blank?
1812     {
1813         \pdfmanagement_remove:nn {Pages} { CropBox }
1814     }
1815     {
1816         \pdfmanagement_add:nne {Pages} { CropBox } { [#1] }
1817     }
1818 }
1819 ,pdfpicktraybypdfsize .choice:
1820 ,pdfpicktraybypdfsize / true .code:n =
1821 {
1822     \pdf_version_compare:NnTF > {1.6}
1823     {
1824         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1825         { PickTrayByPDFSize } { true }
1826     }
1827     {
1828         \msg_warning:nnee
1829         {hyp}
1830         {ignore-deprecated-or-unknown-option-in-pdf-version}
1831         {pdfpicktraybypdfsize}
1832         {\pdf_version:}
1833     }
1834 }
1835 ,pdfpicktraybypdfsize / false .code:n =
1836 {
1837     \pdf_version_compare:NnTF > {1.6}
1838     {
1839         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1840         { PickTrayByPDFSize } { false }
1841     }
1842     {
1843         \msg_warning:nnee
1844         {hyp}
1845         {ignore-deprecated-or-unknown-option-in-pdf-version}

```

```

1846         {pdfpicktraybypdfsize}
1847         {\pdf_version:}
1848     }
1849 }
1850 ,pdfpicktraybypdfsize / .code:n =
1851 {
1852     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1853 }
1854 ,pdfpicktraybypdfsize / unknown .code:n =
1855 {
1856     \msg_warning:nnee { hyp } { no-bool }
1857     { picktraybypdfsize }
1858     { \exp_not:n {#1} }
1859 }
1860 ,pdfprintarea .choices:nn =
1861 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1862 {
1863     \pdf_version_compare:NnTF < {2.0}
1864     {
1865         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1866         { PrintArea } { /#1 }
1867     }
1868     {
1869         \msg_warning:nnee
1870         {hyp}
1871         {ignore-deprecated-or-unknown-option-in-pdf-version}
1872         {pdfprintarea}
1873         {\pdf_version:}
1874     }
1875 }%
1876 ,pdfprintarea / .code:n =
1877 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1878 ,pdfprintarea / unknown .code:n =
1879 {
1880     \msg_warning:nnee { hyp } { unknown-choice+empty }
1881     { pdfprintarea }
1882     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1883     { \exp_not:n {#1} }
1884 }
1885 ,pdfprintclip .choices:nn =
1886 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1887 {
1888     \pdf_version_compare:NnTF < {2.0}
1889     {
1890         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1891         { PrintClip } { /#1 }
1892     }
1893     {
1894         \msg_warning:nnee
1895         {hyp}
1896         {ignore-deprecated-or-unknown-option-in-pdf-version}
1897         {pdfprintclip}
1898         {\pdf_version:}
1899     }

```

```

1900     }%
1901 ,pdfprintclip / .code:n =
1902     {
1903         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1904     }
1905 ,pdfprintclip / unknown .code:n =
1906     {
1907         \msg_warning:nnee
1908         { hyp }
1909         { unknown-choice+empty }
1910         { pdfprintclip }
1911         { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1912         { \exp_not:n {#1} }
1913     }
1914 ,pdfprintpagerange .code:n =
1915     {
1916         \pdf_version_compare:NnTF > {1.6}
1917         {
1918             \tl_if_empty:nTF { #1}
1919             {
1920                 \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1921                 { PrintPageRange }
1922             }
1923             {
1924                 \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1925                 {PrintPageRange}{[#1]}
1926             }
1927         }
1928         {
1929             \msg_warning:nnee
1930             {hyp}
1931             {ignore-deprecated-or-unknown-option-in-pdf-version}
1932             {pdfprintpagerange}
1933             {\pdf_version:}
1934         }
1935     }
1936 ,pdfprintscaling .choices:nn =
1937     { None, AppDefault }
1938     {
1939         \pdf_version_compare:NnTF > {1.5}
1940         {
1941             \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1942             { PrintScaling } { /#1 }
1943         }
1944         {
1945             \msg_warning:nnee
1946             {hyp}
1947             {ignore-deprecated-or-unknown-option-in-pdf-version}
1948             {pdfprintscaling}
1949             {\pdf_version:}
1950         }
1951     }%
1952 ,pdfprintscaling / .code:n =
1953     {

```

```

1954     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1955   }
1956 ,pdfprintscaling / unknown .code:n =
1957   {
1958     \msg_warning:nneee { hyp } { unknown-choice+empty }
1959     { pdfprintarea }
1960     { None, AppDefault }
1961     { \exp_not:n {#1} }
1962   }
1963 ,pdfremotestartview .code:n =
1964   {
1965     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
1966     \exp_args:NNV
1967     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1968     {
1969       \tl_set:Ne \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1970     }
1971     {
1972       \msg_warning:nmmn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1973       \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1974     }
1975   }
1976 ,pdfremotestartview .initial:n = {Fit}
1977 % pdfstartpage is special as it shares code with pdfstartview
1978 ,pdfstartpage .code:n =
1979   {
1980     \tl_gset:Ne \g__hyp_dest_pdfstartpage_tl { #1 }
1981     \bool_if:nTF
1982     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1983     {
1984       \pdfmanagement_remove:nn {Catalog} { OpenAction }
1985     }
1986     {
1987       \pdfmanagement_add:nne {Catalog} { OpenAction }
1988       {
1989         [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1990       }
1991     }
1992   }
1993 ,pdfstartpage .initial:n =1
1994 ,pdfstartview .code:n =
1995   {
1996     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
1997     \exp_args:NNV
1998     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1999     {
2000       \tl_gset:Ne \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
2001     }
2002     {
2003       \msg_warning:nmmn {hyp}{invalid-destination-value}{#1}{pdfstartview}
2004       \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
2005     }
2006     \bool_if:nTF
2007     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd

```

```

2008     {
2009     \pdfmanagement_remove:nn {Catalog} { OpenAction }
2010     }
2011     {
2012     \pdfmanagement_add:nne {Catalog} { OpenAction }
2013     {
2014     [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
2015     }
2016     }
2017     }
2018 ,pdfstartview .initial:n = Fit
2019 ,pdftoolbar .choice:
2020 ,pdftoolbar / true .code:n =
2021 {
2022 \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
2023 }
2024 ,pdftoolbar / false .code:n =
2025 {
2026 \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2027 { HideToolbar } { true }
2028 }
2029 ,pdftoolbar / true .code:n =
2030 {
2031 \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
2032 }
2033 ,pdftoolbar .default:n = true
2034 ,pdftoolbar / unknown .code:n =
2035 {
2036 \msg_warning:nnee { hyp } { no-bool }
2037 { pdftoolbar }
2038 { \exp_not:n {#1} }
2039 }
2040 % pdfview see below.
2041 ,pdfviewarea .choices:nn =
2042 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2043 {
2044 \pdf_version_compare:NnTF < {2.0}
2045 {
2046 \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2047 { ViewArea } { /#1 }
2048 }
2049 {
2050 \msg_warning:nnee
2051 {hyp}
2052 {ignore-deprecated-or-unknown-option-in-pdf-version}
2053 {pdfviewarea}
2054 {\pdf_version:}
2055 }
2056 }%
2057 ,pdfviewarea / .code:n =
2058 {
2059 \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
2060 }
2061 ,pdfviewarea / unknown .code:n =

```

```

2062     {
2063     \msg_warning:nneee { hyp } { unknown-choice+empty }
2064     { pdfviewarea }
2065     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2066     { \exp_not:n {#1} }
2067     }
2068 ,pdfviewclip .choices:nn =
2069 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2070 {
2071 \pdf_version_compare:NnTF < {2.0}
2072 {
2073 \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2074 { ViewClip } { /#1 }
2075 }
2076 {
2077 \msg_warning:nnee
2078 {hyp}
2079 {ignore-deprecated-or-unknown-option-in-pdf-version}
2080 {pdfviewclip}
2081 {\pdf_version:}
2082 }
2083 }%
2084 ,pdfviewclip / .code:n =
2085 {
2086 \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
2087 }
2088 ,pdfviewclip / unknown .code:n =
2089 {
2090 \msg_warning:nneee { hyp } { unknown-choice+empty }
2091 { pdfviewclip }
2092 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2093 { \exp_not:n {#1} }
2094 }
2095 ,pdfwindowui .choice:
2096 ,pdfwindowui / true .code:n =
2097 {
2098 \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2099 }
2100 ,pdfwindowui / false .code:n =
2101 {
2102 \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2103 { HideWindowUI } { true }
2104 }
2105 ,pdfwindowui / .code:n =
2106 {
2107 \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2108 }
2109 ,pdfwindowui / unknown .code:n =
2110 {
2111 \msg_warning:nnee { hyp } { no-bool }
2112 { pdfwindowui }
2113 { \exp_not:n {#1} }
2114 }
2115 ,pdfwindowui .default:n = true

```

2116 }

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

```
2117 \keys_define:nn { hyp }
2118 {
2119   ,pdfview .code:n =
2120   {
2121     \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
2122     \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
2123     {
2124       { xyz }
2125       {
2126         \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
2127         {
2128           \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2129           \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2130           {
2131             \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2132           }
2133           {
2134             \tl_set:Ne \l__hyp_dest_pdfview_tl
2135             {
2136               \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2137             }
2138           }
2139         }
2140         {
2141           \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2142         }
2143       }
2144       { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2145       { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2146       { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2147       { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2148       { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2149       { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2150       { fitr }
2151       {
2152         \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2153         {
2154           \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2155         }
2156         {
2157           %ensure 4 values ...
2158           \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2159           \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2160           \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2161           \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2162           \hbox_set_to_wd:Nnn \l__hyp_dest_box
2163           {
2164             \fp_eval:n
2165             {
2166               round
2167               (
```



```

2168         abs
2169         (
2170         \seq_item:Nn\l__hyp_tmpa_seq{4}
2171         -
2172         (\seq_item:Nn\l__hyp_tmpa_seq{2})
2173         ),
2174         3
2175     )
2176 }bp
2177 }{}
2178 \box_set_dp:Nn \l__hyp_dest_box
2179 {
2180     \fp_eval:n
2181     {
2182         round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2183     }bp
2184 }
2185 \box_set_ht:Nn \l__hyp_dest_box
2186 {
2187     \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2188 }
2189 }
2190 }
2191 }
2192 {
2193     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2194     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2195 }
2196 }
2197 ,pdfview .initial:n = {xyz}
2198 }

```

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.

```

2199 \keys_define:nn { hyp }
2200 {
2201     ,pdflang .code:n =
2202     {
2203         \tl_if_empty:nF { #1 }
2204         {
2205             \pdfmanagement_add:nne {Catalog} { Lang } { (#1) }
2206             \AddToDocumentProperties[document]{lang}{#1}
2207         }
2208     }
2209 }

```

12.3.1 “info keys”

`pdfauthor` (*setup key*) The keys store their value also in the metadata container, so that hyperxmp can use them.
`pdftitle` (*setup key*) Creator and Producer can’t be removed with the pdfmanagement, but we allow to set an
`pdfcreator` (*setup key*)
`pdfsubject` (*setup key*)
`pdfproducer` (*setup key*)
`pdfkeywords` (*setup key*)

empty value. If the value begin with an optional argument, we assume a multilanguage
clist and use only the first value. The values are expanded with `\text_expand:n`

```

2210 \regex_new:N\l__hyp_optlang_regex
2211 \regex_set:Nn\l__hyp_optlang_regex {\A\[[A-Za-z\-\+]\](.*)}
2212 \cs_generate_variant:Nn\clist_item:nn{on}
2213 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2214 {
2215   \keys_define:nn { hyp }
2216   {
2217     pdf#1 .code:n =
2218     {
2219       \tl_set:Nel__hyp_tmpa_tl {\text_expand:n{##1}}
2220       \__hyp_store_metadata:no {pdf#1}{\l__hyp_tmpa_tl}
2221       \tl_if_empty:NTF \l__hyp_tmpa_tl
2222       {
2223         \str_case:nnF { #1 }
2224         {
2225           {creator}
2226           {
2227             \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2228             \pdfmanagement_add:nne {Info}{Creator}{()}
2229           }
2230           {producer}
2231           {
2232             \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2233             \pdfmanagement_add:nne {Info}{Producer}{()}
2234           }
2235         }
2236         {
2237           \pdfmanagement_remove:nn {Info}{#2}
2238         }
2239       }
2240     {
2241       \tl_set:Nel__hyp_tmpb_tl {\clist_item:on{\l__hyp_tmpa_tl}{1}}
2242       \exp_args:NNV
2243       \regex_extract_once:NnN \l__hyp_optlang_regex \l__hyp_tmpb_tl\l__hyp_tmpa_s
2244       \seq_if_empty:NTF\l__hyp_tmpa_seq
2245       {
2246         \__hyp_text_pdfstring_info:oN {\l__hyp_tmpa_tl}\l__hyp_tmpa_str
2247       }
2248       {
2249         \__hyp_text_pdfstring_info:eN {\seq_item:Nn \l__hyp_tmpa_seq{3}}\l__hyp_t
2250       }
2251       \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2252       {
2253         \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2254       }
2255     }
2256   }
2257 }
2258 \keys_define:nn { hyp / info }
2259 {
2260   #2 .code:n =
2261   {

```

```

2262 \tl_set:N\l__hyp_tmpa_tl {\text_expand:n{##1}}
2263 \__hyp_store_metadata:eo {pdf\str_lowercase:n{##1}}{\l__hyp_tmpa_tl}
2264 \tl_if_blank:nTF {##1}
2265 {
2266 \pdfmanagement_remove:nn {Info}{#2}
2267 }
2268 {
2269 \__hyp_text_pdfstring_info:oN {\l__hyp_tmpa_tl}\l__hyp_tmpa_str
2270 \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2271 {
2272 \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2273 }
2274 }
2275 }
2276 ,unknown .code:n =
2277 {
2278 \__hyp_text_pdfstring_info:eN {##1}\l__hyp_tmpa_str
2279 \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2280 {
2281 \exp_args:Nno
2282 \pdfmanagement_add:nne {Info}
2283 { \l_keys_key_str } {\l__hyp_tmpa_str}
2284 }
2285 }
2286 }
2287 }
2288 \__hyp_setup_info_key:nn {author} {Author}
2289 \__hyp_setup_info_key:nn {title} {Title}
2290 \__hyp_setup_info_key:nn {producer} {Producer}
2291 \__hyp_setup_info_key:nn {creator} {Creator}
2292 % ignored key: addtopdfcreator
2293 \__hyp_setup_info_key:nn {subject} {Subject}
2294 \__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)
2295 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2296 {
2297 \keys_define:nn { hyp }
2298 {
2299 pdf#1 .code:n =
2300 {
2301 \tl_if_blank:nTF {##1}
2302 {
2303 \pdfmanagement_remove:nn {Info}{#2}
2304 }
2305 {
2306 \pdfmanagement_add:nne {Info}{#2}{(##1)}
2307 }
2308 \__hyp_store_metadata:nn {pdf#1}{##1}
2309 \AddToDocumentProperties[document]{#1}{##1}
2310 }
2311 }
2312 \keys_define:nn { hyp / info }

```

```

2313     {
2314     #2 .code:n =
2315     {
2316     \tl_if_blank:nTF {##1}
2317     {
2318     \pdfmanagement_remove:nn {Info}{#2}
2319     }
2320     {
2321     \pdfmanagement_add:nne {Info}{#2}{(##1)}
2322     }
2323     \exp_args:Ne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2324     }
2325     }
2326   }
2327
2328   \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2329   \__hyp_setup_info_date_key:nn {moddate} {ModDate}
2330   \keys_define:nn { hyp }
2331   {
2332     pdfmetadate .code:n = { \__hyp_store_metadata:nn {pdfmetadate}{#1} }
2333   }

```

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

```

2334   \keys_define:nn { hyp }
2335   {
2336     ,pdftrapped .code:n =
2337     {
2338     \exp_args:Nne
2339     \keys_set:nn { hyp } { _pdftrapped = \str_uppercase:n { #1 } }
2340     }
2341     ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2342     {
2343     \pdfmanagement_add:nne {Info}{Trapped}
2344     {/
2345     \str_uppercase:f { \str_head:n { #1 } }
2346     \str_lowercase:f { \str_tail:n { #1 } }
2347     }
2348     \__hyp_store_metadata:ne {pdftrapped}
2349     {
2350     \str_uppercase:f { \str_head:n { #1 } }
2351     \str_lowercase:f { \str_tail:n { #1 } }
2352     }
2353     }
2354     ,_pdftrapped / unknown .code:n =
2355     {
2356     \msg_warning:nnee { hyp } { unknown-choice }
2357     { pdftrapped }
2358     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2359     { \exp_not:n {#1} }
2360     }
2361   }

```

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

```

2362   \keys_define:nn { hyp }

```

```

2363 {
2364   pdfinfo .code:n =
2365   {
2366     \keys_set:nn { hyp / info } { #1 }
2367   }
2368 }

```

Now we set some default values

```

2369 \keys_set:nn { hyp} {pdfcreator = LaTeX-with-hyperref}
2370 \keys_set:nn { hyp} {pdfauthor  = }
2371 \keys_set:nn { hyp} {pdftitle   = }
2372 \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:

```

2373 \clist_map_inline:nn
2374 {
2375   ,pdfcopyright
2376   ,pdftype
2377   ,pdflicenseurl
2378   ,pdfauthorortitle
2379   ,pdfcaptionwriter
2380   ,pdfmetalang
2381   ,pdfsource
2382   ,pdfdocumentid
2383   ,pdfinstanceid
2384   ,pdfversionid
2385   ,pdfrendition
2386   ,pdfpublication
2387   ,pdfpubtype
2388   ,pdfbytes
2389   ,pdfnumpages
2390   ,pdfissn
2391   ,pdfeissn
2392   ,pdfisbn
2393   ,pdfbookedition
2394   ,pdfpublisher
2395   ,pdfvolumenum
2396   ,pdfissuenum
2397   ,pdfpagerange
2398   ,pdfdoi
2399   ,pdfurl
2400   ,pdfidentifier
2401   ,pdfsubtitle
2402   ,pdfpubstatus
2403   ,pdfcontactaddress
2404   ,pdfcontactcity
2405   ,pdfcontactregion
2406   ,pdfcontactpostcode
2407   ,pdfcontactcountry
2408   ,pdfcontactphone

```

```

2409 ,pdfcontactemail
2410 ,pdfcontacturl
2411 ,pdfdate

```

these are not hyperxmp but new keys

```

2412 ,pdftdmreservation
2413 ,pdftdmpolicy
2414 }
2415 {
2416 \keys_define:nn { hyp }
2417 {
2418 #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2419 }
2420 }
2421

```

12.5 Transitions

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

```

2422 \keys_define:nn { hyp }
2423 {
2424 pdfpageduration .code:n =
2425 {
2426 \tl_if_blank:nTF { #1 }
2427 {
2428 \pdfmanagement_remove:nn {Page}{Dur}
2429 }
2430 {
2431 \pdfmanagement_add:nnn {Page}{Dur}{#1}
2432 }
2433 }
2434 }

```

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

```

2435 \keys_define:nn { hyp }
2436 {
2437   pdfpagetransition .code:n =
2438   {
2439     \tl_if_blank:nTF {#1}
2440     {
2441       \pdfmanagement_remove:nn {Page}{Trans}
2442     }
2443     {
2444       \group_begin:
2445       \keys_set:nn { hyp / trans }{style=R,#1}
2446       \pdf_object_unnamed_write:ne { dict }
2447       {
2448         \pdfdict_use:n {l__hyp_page/Trans}
2449       }
2450       \pdfmanagement_add:nne {Page}{Trans}{\pdf_object_ref_last:}
2451       \group_end:
2452     }
2453   }
2454 }
2455 \keys_define:nn { hyp / trans }
2456 {
2457   ,style .choices:nn =
2458   {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2459   { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2460   ,style / unknown .code:n =
2461   {
2462     \msg_warning:nnee { hyp } { unknown-choice }
2463     { trans / style }
2464     { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2465     { \exp_not:n {#1} }
2466   }
2467   ,duration .code:n =
2468   {
2469     \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}
  
```

```

2470     }
2471 ,direction .choices:nn =
2472     {H,V}
2473     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2474 ,direction .choices:nn =
2475     {0,90,180,270,315}
2476     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2477 ,direction / None .code:n =
2478     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2479 ,direction / unknown .code:n =
2480     {
2481         \msg_warning:nneee { hyp } { unknown-choice }
2482         { trans / direction }
2483         {
2484             H~(horizontal,~only~Split,~Blinds),
2485             V~(vertical,~only~Split,~Blinds),
2486             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2487             90~(bottom~to~top,~only~Wipe),
2488             180~(right~to~left,~only~Wipe),
2489             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2490             315~(top~left~to~bottom,~only~Glitter),
2491             None~(only~Fly)
2492         }
2493         { \exp_not:n {#1} }
2494     }
2495 ,motion .choices:nn =
2496     {I,0}
2497     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2498 ,motion / unknown .code:n =
2499     {
2500         \msg_warning:nneee { hyp } { unknown-choice }
2501         { trans / motion }
2502         { I~(inwards) , 0~(outwards) }
2503         { \exp_not:n {#1} }
2504     }
2505 ,scale .code:n =
2506     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2507 ,opaque .choices:nn = {true,false}
2508     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2509 ,opaque / unknown .code:n =
2510     {
2511         \msg_warning:nneee { hyp } { unknown-choice }
2512         { trans / B }
2513         { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2514         { \exp_not:n {#1} }
2515     }
2516 % try to set unknown keys as style
2517 ,unknown .code:n =
2518     {
2519         % warning ...
2520         \exp_args:Nne\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2521     }
2522 }

```


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

```

2523 \keys_set_known:nv{ hyp }{opt@hyperref.sty}
    Unfinished, from here on @@ notation is disabled Form field code
2524 <@=@>
2525 \NewDocumentCommand \MakeFieldObject { m m }
2526 {
2527   \pdfxform_new:nnn { #2 }{ } { #1 }
2528 }
2529
2530
2531 \prop_new:N \g__hyp_AcroForm_CoFields_prop
2532 \prop_new:N \g__hyp_AcroForm_Fields_prop
2533
2534 \let\HyField@afields\@empty
2535 \let\HyField@cofields\@empty
2536 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2537
2538 % the value doesn't matter, but with a prop we avoid duplicates and it is
2539 % clearly faster than removing them from a sequence
2540 \def\HyField@AuxAddToFields#1
2541 {
2542   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2543 }%
2544
2545 %fields with empty key get a value too -- lets hope that
2546 %this give the expected behaviour
2547 \def\HyField@AuxAddToCoFields #1 #2
2548 {
2549   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2550 }
2551
2552 \Hy@AtBeginDocument
2553 {
2554   \if@filesw
2555     \immediate\write\@mainaux{%
2556       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2557     }%
2558     \immediate\write\@mainaux{%
2559       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2560     }%
2561   \fi
2562   \let\HyField@AfterAuxOpen\@firstofone
2563 }%
2564
2565 \def\HyField@AddToFields
2566 {
2567   \exp_args:Ne\HyField@@AddToFields
2568   {
2569     \pdfannot_box_ref_last:
2570   }
2571   \ifx\Fld@calculate@code\@empty
2572   \else
2573     \begingroup

```

```

2574     \Hy@safe@activestru
2575     \edef\Hy@temp{%
2576         \endgroup
2577         \if@filesw
2578             \write\@mainaux
2579             {
2580                 \string\HyField@AuxAddToCoFields
2581                 {
2582                     \Fld@calculate@sortkey
2583                 }
2584                 {
2585                     \pdfannot_box_ref_last:
2586                 }
2587             }
2588         \fi
2589     }%
2590     \Hy@temp
2591 \fi
2592 }%
2593
2594 \def\HyField@@AddToFields#1{
2595     \HyField@AfterAuxOpen{%
2596         \if@filesw
2597             \write\@mainaux{%
2598                 \string\HyField@AuxAddToFields{#1}%
2599             }%
2600         \fi
2601     }%
2602 }%
2603
2604 \ExplSyntaxOff
2605 \ExplSyntaxOn
2606
2607 \def\@Form[#1]
2608 {
2609     \kvsetkeys{Form}{#1}
2610     \pdf@ifdraftmode{}
2611     {
2612         \Hy@FormObjects
2613         \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2614         {
2615             \pdfmanagement_add:nne { Catalog / AcroForm } { Fields }{##1}
2616             %\pdfmanagement_show:n { Catalog / AcroForm }
2617         }
2618         \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2619         {
2620             \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2621             {
2622                 \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2623             }
2624             \seq_sort:Nn \l__hyp_tmpa_seq
2625             {
2626                 \str_compare:nNnTF {##1} > {##2}
2627                 { \sort_return_swapped: }

```

```

2628         { \sort_return_same: }
2629     }
2630     \seq_map_inline:Nn \l__hyp_tmpa_seq
2631     {
2632         \pdfmanagement_add:nne { Catalog / AcroForm }
2633         { CO }
2634         {
2635             \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2636         }
2637     }
2638 }
2639 \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2640 {ZaDb} {\pdf_object_ref:n {__hyp/Font/ZaDb} }
2641 \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2642 {Helv} {\pdf_object_ref:n {__hyp/Font/Helv} }
2643 \pdfmanagement_add:nne {Catalog /AcroForm}
2644 {DA}{(/Helv~10~Tf~0~g)}
2645 \pdfmeta_standard_verify:nTF {form_no_NeedAppearance}
2646 {
2647     \legacy_if:nT { HyField@NeedAppearances }
2648     {
2649         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2650     }
2651 }
2652 {
2653     \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2654 }
2655 }
2656 }
2657 \ExplSyntaxOff
2658 \let\@endForm\@empty
2659 \let\HyAnn@AbsPageLabel\@empty
2660 \let\Fld@pageobjref\@empty
2661
2662 \ExplSyntaxOn
2663 \newcount\HyAnn@Count
2664 \HyAnn@Count=\z@
2665 \def\HyAnn@AbsPageLabel
2666 {
2667     \global\advance\HyAnn@Count by\@ne
2668     \__hyp_property_record:ee {HyAnn@\the\HyAnn@Count}{abspage}
2669     \property_ref_undefined_warn:ee {HyAnn@\the\HyAnn@Count}{abspage}
2670 }%
2671 \prg_generate_conditional_variant:Nnn \property_if_recorded:nn {ee} {T}
2672 \def\Fld@pageobjref
2673 {
2674     \property_if_recorded:eeT {HyAnn@\the\HyAnn@Count}{abspage}
2675     {
2676         /P~\pdf_pageobject_ref:e
2677         {
2678             \property_ref:ee{HyAnn@\the\HyAnn@Count}{abspage}
2679         }
2680     }
2681 }

```

```

2682 %% check if the attr should be set through
2683 %% hooks.
2684 %% check if options are missing.
2685 \def\@TextField[#1]#2{% parameters, label
2686   \def\Fld@name{#2}%
2687   \let\Fld@default\@empty
2688   \let\Fld@value\@empty
2689   \def\Fld@width{\DefaultWidthofText}%
2690   \def\Fld@height{%
2691     \ifFld@multiline
2692       \DefaultHeightofTextMultiline
2693     \else
2694       \DefaultHeightofText
2695     \fi
2696   }%
2697   \begingroup
2698     \expandafter\HyField@SetKeys\expandafter{%
2699       \DefaultOptionsofText,#1%
2700     }%
2701     \PDFForm@Name
2702     \HyField@FlagsText
2703     \ifFld@hidden\def\Fld@width{1sp}\fi
2704     \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2705     \LayoutTextField{#2}{%
2706       \leavevmode
2707       \HyAnn@AbsPageLabel
2708       \Hy@escapeform\PDFForm@Text
2709       \pdfannot_box:nnnn
2710         {\Fld@width}
2711         {\Fld@height}
2712         {0pt} %is this correct?
2713         {\PDFForm@Text}
2714       \MakeTextField{\Fld@width}{\Fld@height}
2715       \HyField@AddToFields
2716     }%
2717   \endgroup
2718 }
2719 \providecommand\@curropt{}
2720 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2721   \def\Fld@name{#2}
2722   \let\Fld@default\relax
2723   \let\Fld@value\relax
2724   \def\Fld@width{\DefaultWidthofChoiceMenu}
2725   \def\Fld@height{\DefaultHeightofChoiceMenu}
2726   \begingroup
2727     \Fld@menulength=0 %
2728     \@tempdima\z@
2729     \clist_map_variable:nNn { #3 } \@curropt
2730     %\@for\@curropt:=#3\do
2731     {%
2732       \expandafter\Fld@checkequals\@curropt==\%
2733       \Hy@StepCount\Fld@menulength
2734       \settowidth{\@tempdimb}{\@currDisplay}%
2735       \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi

```

```

2736     }%
2737     \advance\@tempdima by~15\p@
2738     \begingroup
2739     \HyField@SetKeys{#1}
2740     \edef\x{\endgroup
2741     \noexpand\expandafter
2742     \noexpand\HyField@SetKeys
2743     \noexpand\expandafter{%-
2744     \expandafter\noexpand\cname DefaultOptionsof%
2745     \ifFld@radio
2746     Radio%
2747     \else
2748     \ifFld@combo
2749     \ifFld@popdown
2750     PopdownBox%
2751     \else
2752     ComboBox%
2753     \fi
2754     \else
2755     ListBox%
2756     \fi
2757     \fi
2758     \endcsname
2759     }%
2760     }\x
2761     \HyField@SetKeys{#1}%
2762     \PDFForm@Name
2763     \ifFld@hidden\def\Fld@width{1sp}\fi
2764     \ifx\Fld@value\relax
2765     \let\Fld@value\Fld@default
2766     \fi
2767     \LayoutChoiceField{#2}{%-
2768     \ifFld@radio
2769     \HyField@FlagsRadioButton
2770     \@Radio{#3}%
2771     \else
2772     \begingroup
2773     \HyField@FlagsChoice
2774     \ifdim\Fld@width<\@tempdima
2775     \ifdim\@tempdima<1cm\@tempdima1cm\fi
2776     \edef\Fld@width{\the\@tempdima}%
2777     \fi
2778     \ifFld@combo
2779     \else
2780     \@tempdima=\the\Fld@menulength\Fld@charsize
2781     \advance\@tempdima by~\Fld@borderwidth bp %
2782     \advance\@tempdima by~\Fld@borderwidth bp %
2783     \edef\Fld@height{\the\@tempdima}%
2784     \fi
2785     \@ListBox{#3}%
2786     \endgroup
2787     \fi
2788     }%
2789     \endgroup

```

```

2790 }
2791
2792 \def\@@Radio#1{%
2793   \Fld@listcount=0-%
2794   \EdefEscapeName\Fld@default{\Fld@default}%
2795   \clist_map_variable:nNn { #1 } \@curropt
2796   {%
2797     \expandafter\Fld@checkequals\@curropt==\%
2798     \EdefEscapeName\@currValue{\@currValue}%
2799     \Hy@StepCount\Fld@listcount
2800     \@currDisplay\space
2801     \leavevmode
2802     \HyAnn@AbsPageLabel
2803     \Hy@escapeform\PDFForm@Radio
2804     \pdfxform_if_exist:nF { __hyp_xform_Ding }
2805     {
2806       \pdfxform_new:nnn { __hyp_xform_Ding } {}
2807       {
2808         \group_begin:
2809         \fontfamily{pzd}
2810         \fontencoding{U}
2811         \fontseries{m}
2812         \fontshape{n}
2813         \selectfont
2814         \char123
2815         \group_end:
2816       }
2817     }
2818     \pdfannot_box:nnne
2819     {\Fld@width}
2820     {\Fld@height}
2821     {Opt} %is this correct?
2822     {
2823       \PDFForm@Radio
2824       /AP
2825       <<
2826       /N
2827       <<
2828       /\@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2829       %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2830       >>
2831     }
2832   }
2833   {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2834   \int_compare:nNnT { \Fld@listcount } = { 1 }
2835   { \HyField@AddToFields }
2836   \c_space_tl % deliberate space between radio buttons
2837   % to do: --> should be configurable
2838 }%
2839 }
2840
2841 \newcount\Fld@listcount
2842 \def\@@ListBox#1
2843 {

```

```

2844 \HyField@PDFChoices{#1}
2845 \mode_leave_vertical:
2846 \HyAnn@AbsPageLabel
2847 \Hy@escapeform\PDFForm@List
2848 \pdfannot_box:nnnn
2849 {\Fld@width}{\Fld@height}{0pt}
2850 {\PDFForm@List}
2851 \MakeChoiceField{\Fld@width}{\Fld@height}
2852 \HyField@AddToFields
2853 }
2854
2855
2856 \def\@PushButton[#1]#2{% parameters, label
2857 \def\Fld@name{#2}%
2858 \group_begin:
2859 \exp_args:No\HyField@SetKeys
2860 {
2861 \DefaultOptionsofPushButton,#1
2862 }
2863 \PDFForm@Name
2864 \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2865 {
2866 \HyField@FlagsPushButton
2867 \legacy_if:nT {Fld@hidden}
2868 {
2869 \def\Fld@width{1sp}
2870 }
2871 \LayoutPushButtonField
2872 {
2873 \mode_leave_vertical:
2874 \HyAnn@AbsPageLabel
2875 \Hy@escapeform\PDFForm@Push
2876 \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2877 \pdfannot_box:nnnn
2878 {\box_wd:N\l_tmpa_box}
2879 {\box_ht:N\l_tmpa_box}
2880 {\box_dp:N\l_tmpa_box} %is this correct?
2881 {\PDFForm@Push}
2882 {\box_use:N\l_tmpa_box}
2883 \HyField@AddToFields
2884 }
2885 }
2886 {
2887 \msg_error:nn { hyp }{ pdfa-no-push-button }
2888 \LayoutPushButtonField
2889 {
2890 \mode_leave_vertical:
2891 \MakeButtonField{#2}
2892 }
2893 }
2894 \group_end:
2895 }
2896
2897 \def\@Submit[#1]#2

```

```

2898 {
2899   \def\Fld@width {\DefaultWidthofSubmit}
2900   \def\Fld@height{\DefaultHeightofSubmit}
2901   \group_begin:
2902     \exp_args:No\HyField@SetKeys
2903     {
2904       \DefaultOptionsofSubmit,#1
2905     }
2906   \HyField@FlagsPushButton
2907   \HyField@FlagsSubmit
2908   \legacy_if:nT { Fld@hidden }
2909   {
2910     \def\Fld@width{1sp}
2911   }
2912   \mode_leave_vertical:
2913   \HyAnn@AbsPageLabel
2914   \Hy@escapeform\PDFForm@Submit
2915   \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2916   \pdfxform_if_exist:nF
2917   { __hyp_xform_Submit }
2918   {
2919     \pdfxform_new:nnn { __hyp_xform_Submit }{ }
2920     {
2921       \fbox{\color_select:n{yellow}\textsf{Submit}}
2922     }
2923     \pdfxform_new:nnn { __hyp_xform_SubmitP }{ }
2924     {
2925       \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2926     }
2927   }
2928   \pdfannot_box:nnnn
2929   {\box_wd:N\l_tmpa_box}
2930   {\box_ht:N\l_tmpa_box}
2931   {\box_dp:N\l_tmpa_box} %is this correct?
2932   {
2933     \PDFForm@Submit
2934     /AP<<
2935     /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2936     /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2937     >>
2938   }
2939   \HyField@AddToFields
2940   \box_use:N\l_tmpa_box
2941
2942   \group_end:
2943 }
2944
2945 \def\@Reset[#1]#2
2946 {
2947   \def\Fld@width {\DefaultWidthofReset}
2948   \def\Fld@height{\DefaultHeightofReset}
2949   \group_begin:
2950     \exp_args:No\HyField@SetKeys
2951     {

```



```

2952     \DefaultOptionsofReset,#1
2953   }
2954   \mode_leave_vertical:
2955   \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2956   {
2957     \HyField@FlagsPushButton
2958     \legacy_if:nT { Fld@hidden }
2959     { \def\Fld@width{1sp} }
2960     \HyAnn@AbsPageLabel
2961     \Hy@escapeform\PDFForm@Reset
2962     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2963     \pdfannot_box:nnnn
2964     {\box_wd:N\l_tmpa_box}
2965     {\box_ht:N\l_tmpa_box}
2966     {\box_dp:N\l_tmpa_box} %is this correct?
2967     { \PDFForm@Reset }
2968     \HyField@AddToFields
2969     \box_use:N \l_tmpa_box
2970   }
2971   {
2972     \msg_error:nn { hyp }{ pdfa-no-reset-button }
2973     \MakeButtonField{#2}
2974   }
2975   \group_end:
2976 }
2977
2978 \def\@CheckBox[#1]#2
2979 {% parameters, label
2980   \def\Fld@name{#2}
2981   \def\Fld@default{0}
2982   \group_begin:
2983     \def\Fld@width {\DefaultWidthofCheckBox}
2984     \def\Fld@height{\DefaultHeightofCheckBox}
2985     \exp_args:No\HyField@SetKeys
2986     {
2987       \DefaultOptionsofCheckBox,#1
2988     }
2989     \PDFForm@Name
2990     \HyField@FlagsCheckBox
2991     \legacy_if:nT { Fld@hidden }
2992     {
2993       \def\Fld@width{1sp}
2994     }
2995     \LayoutCheckField{#2}
2996     {
2997       \mode_leave_vertical:
2998       \HyAnn@AbsPageLabel
2999       \Hy@escapeform\PDFForm@Check
3000       \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
3001       {
3002         \pdfxform_new:nnn
3003         {__hyp_xform_CheckMarkYes}{-}
3004         {
3005           \group_begin:

```

```

3006         \fontfamily{pzd}
3007         \fontencoding{U}
3008         \fontseries{m}
3009         \fontshape{n}
3010         \selectfont
3011         \char51
3012         \group_end:
3013     }
3014     \pdfxform_new:nnn
3015     {\_hyp_xform_CheckMark0ff}{ }
3016     {
3017         \group_begin:
3018         \fontfamily{pzd}
3019         \fontencoding{U}
3020         \fontseries{m}
3021         \fontshape{n}
3022         \selectfont
3023         \phantom{\char51} %perhaps xetex needs some small glyph ..
3024         \group_end:
3025     }
3026 }
3027 \pdfannot_box:nnnn
3028   {\Fld@width}
3029   {\Fld@height}
3030   {Opt} %is this correct?
3031   {\PDFForm@Check}
3032   \HyField@AddToFields %check if this works with xelatex ...
3033 }
3034 \group_end:
3035 }
3036 %hm. Should a luatex driver use type1 fonts in fields????
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/ograde/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@@escapeform\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@@escapeform#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@@escapeform#3\@nil
3140     }%
3141   \fi
3142 }%
3143 \fi
3144 \def\PDFForm@Name{%
3145   \PDFForm@@Name\Fld@name
3146   \ifx\Fld@altname\relax
3147   \else
3148     \PDFForm@@Name\Fld@altname
3149   \fi
3150   \ifx\Fld@mappingname\relax
3151   \else
3152     \PDFForm@@Name\Fld@mappingname
3153   \fi
3154 }
3155 \def\PDFForm@@Name#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:Ne\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 \def\Fld@annotnames{%
3218 /T(\Fld@name)%
3219 \ifx\Fld@altname\relax
3220 \else
3221 /TU(\Fld@altname)%

```

```

3222 \fi
3223 \ifx\Fld@mappingname\relax
3224 \else
3225 /TM(\Fld@mappingname)%
3226 \fi
3227 }
3228
3229 \def\PDFForm@Check
3230 {
3231 /Subtype/Widget
3232 ~\Fld@annotflags
3233 ~\Fld@pageobjref
3234 ~\Fld@annotnames
3235 /FT/Btn
3236 \Fld@flags
3237 /Q~\Fld@align
3238 /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3239 /AP
3240 <<
3241 /N
3242 <<
3243 /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3244 /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3245 >>
3246 >>
3247 /MK<<
3248 \int_compare:nNnF {\Fld@rotation}={0}
3249 {
3250 /R~\Fld@rotation
3251 }
3252 \tl_if_empty:NF\Fld@bordercolor
3253 {
3254 /BC[\Fld@bordercolor]
3255 }
3256 \tl_if_empty:NF\Fld@bcolor
3257 {
3258 /BG[\Fld@bcolor]
3259 }
3260 /CA(\Hy@escapestring{\Fld@cbsymbol})%
3261 >>
3262 /DA
3263 (
3264 /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3265 \tl_if_empty:NF \Fld@color
3266 {
3267 \c_space_tl \Fld@color
3268 }
3269 )
3270 /H/P
3271 \legacy_if:nTF {Fld@checked}
3272 {
3273 /V/Yes /AS/Yes
3274 }
3275 {

```

```

3276     /V/Off /AS/Off
3277   }
3278   \Fld@additionalactions
3279 }
3280
3281 \def\PDFForm@Push
3282 {
3283   /Subtype/Widget
3284   ~\Fld@annotflags
3285   ~\Fld@pageobjref
3286   ~\Fld@annotnames
3287   /FT/Btn
3288   ~\Fld@flags
3289   /H/P
3290   /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3291   \bool_if:nT
3292   {
3293     !\int_compare_p:nNn {\Fld@rotation} = {0}
3294     ||
3295     \tl_if_exist_p:N \Fld@bordercolor
3296   }
3297   {
3298     /MK
3299     <<
3300     \int_compare:nNnF {\Fld@rotation} = {0}
3301     {
3302       /R~\Fld@rotation
3303     }
3304     \tl_if_exist:NT \Fld@bordercolor
3305     {
3306       /BC[\Fld@bordercolor]
3307     }
3308     >>
3309   }
3310   /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3311   \Fld@additionalactions
3312 }
3313
3314 \ExplSyntaxOff
3315 \def\PDFForm@List{%
3316   /Subtype/Widget%
3317   \Fld@annotflags
3318   \Fld@pageobjref
3319   \Fld@annotnames
3320   /FT/Ch%
3321   \Fld@flags
3322   /Q \Fld@align
3323   /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3324   \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3325     \ifx\Fld@bordercolor\relax\else 1\fi
3326     \ifx\Fld@bcolor\relax \else 1\fi
3327     \space
3328   \else
3329     /MK<<%

```

```

3330     \ifnum\Fld@rotation=\z@
3331     \else
3332         /R \Fld@rotation
3333     \fi
3334     \ifx\Fld@bordercolor\relax
3335     \else
3336         /BC[\Fld@bordercolor]%
3337     \fi
3338     \ifx\Fld@bcolor\relax
3339     \else
3340         /BG[\Fld@bcolor]%
3341     \fi
3342     >>%
3343 \fi
3344 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3345     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3346 \Fld@choices
3347 \Fld@additionalactions
3348 }
3349 \ExplSyntaxOn
3350 \def\PDFForm@Radio
3351 {
3352     /Subtype/Widget
3353     ~\Fld@annotflags
3354     ~\Fld@pageobjref
3355     ~\Fld@annotnames
3356     /FT/Btn
3357     \Fld@flags
3358     /H/P
3359     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3360     /MK<<
3361     \ifnum\Fld@rotation=\z@
3362     \else
3363         /R~\Fld@rotation
3364     \fi
3365     \ifx\Fld@bordercolor\relax
3366     \else
3367         /BC[\Fld@bordercolor]%
3368     \fi
3369     \ifx\Fld@bcolor\relax
3370     \else
3371         /BG[\Fld@bcolor]%
3372     \fi
3373     /CA(\Hy@escapestring{\Fld@radiosymbol})%
3374     >>
3375     /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3376         \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3377     \ifx\Fld@default\@empty
3378         /V/Off%
3379         /DV/Off%
3380     \else
3381         /V/\Fld@default
3382         /DV/\Fld@default
3383     \fi

```



```

3384   \Fld@additionalactions
3385 }
3386
3387 % Does an appearance dict make sense here?
3388 \def\PDFForm@Text
3389 {
3390   /Subtype/Widget
3391   ~\Fld@annotflags
3392   ~\Fld@pageobjref
3393   ~\Fld@annotnames
3394   /FT/Tx
3395   ~\Fld@flags
3396   /Q~\Fld@align
3397   /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3398   \bool_if:nT
3399   {
3400     !\int_compare_p:nNn {\Fld@rotation} = {0}
3401     ||
3402     \tl_if_exist_p:N \Fld@bordercolor
3403     ||
3404     \tl_if_exist_p:N \Fld@bcolor
3405   }
3406   {
3407     /MK
3408     <<
3409     \int_compare:nNnF {\Fld@rotation} = {0}
3410     {
3411       /R~\Fld@rotation
3412     }
3413     \tl_if_exist:NT \Fld@bordercolor
3414     {
3415       /BC[\Fld@bordercolor]
3416     }
3417     \tl_if_exist:NT \Fld@bcolor
3418     {
3419       /BG[\Fld@bcolor]
3420     }
3421     >>
3422   }
3423   /DA
3424   (
3425     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3426     \tl_if_empty:NF {\c_space_tl\Fld@color}
3427   )
3428   /DV(\Hy@escapestring{\Fld@default})
3429   /V(\Hy@escapestring{\Fld@value})
3430   ~\Fld@additionalactions
3431   \int_compare:nNnT { \Fld@maxlen}>{0}
3432   {
3433     /MaxLen~\Fld@maxlen
3434   }
3435 }
3436 \ExplSyntaxOff
3437

```

```

3438 \def\PDFForm@Submit{%
3439   /Subtype/Widget%
3440   \Fld@annotflags
3441   \Fld@pageobjref
3442   \Fld@annotnames
3443   /FT/Btn%
3444   \Fld@flags
3445   /H/P%
3446   /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3447   \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3448     \ifx\Fld@bordercolor\relax\else 1\fi
3449     \space
3450   \else
3451     /MK<<%
3452     \ifnum\Fld@rotation=\z@
3453     \else
3454       /R \Fld@rotation
3455     \fi
3456     \ifx\Fld@bordercolor\relax
3457     \else
3458       /BC[\Fld@bordercolor]%
3459     \fi
3460     >>%
3461   \fi
3462   /A<<%
3463   /S/SubmitForm%
3464   /F<<%
3465   /FS/URL%
3466   /F(\Hy@escapestring{\Form@action})%
3467   >>%
3468   \Fld@submitflags
3469   >>%
3470   \Fld@additionalactions
3471 }
3472 \ExplSyntaxOn
3473   \def\PDFForm@Reset{%
3474     /Subtype/Widget%
3475     \Fld@annotflags
3476     \Fld@pageobjref
3477     \Fld@annotnames
3478     /FT/Btn%
3479     \Fld@flags
3480     /H/P%
3481     /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3482     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3483       \ifx\Fld@bordercolor\relax\else 1\fi
3484       \space
3485     \else
3486       /MK<<%
3487       \ifnum\Fld@rotation=\z@
3488       \else
3489         /R~\Fld@rotation
3490       \fi
3491       \ifx\Fld@bordercolor\relax

```

```

3492         \else
3493             /BC[\Fld@bordercolor]%
3494         \fi
3495     >>%
3496     \fi
3497     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3498     /A<</S/ResetForm>>%
3499     \Fld@additionalactions
3500 }%
3501
3502
3503 %these patterns are used in hyperref checks.
3504 %it is unclear if they are really useful and if a backend support is
3505 %needed.
3506 \str_case:VnF \c_sys_backend_str
3507 {
3508     { pdfmode }
3509     {
3510         \def\HyPat@ObjRef
3511         {
3512             [0-9]*[1-9][0-9]*~0~R
3513         }
3514     }
3515     { dvipdfmx }
3516     {
3517         \def\HyPat@ObjRef
3518         {
3519             @[\~]+
3520         }
3521     }
3522     { xdvipdfmx }
3523     {
3524         \def\HyPat@ObjRef
3525         {
3526             @[\~]+
3527         }
3528     }
3529 }
3530 { %also set in hyperref sty, so probably not needed.
3531     \def\HyPat@ObjRef/{.+}
3532 }
3533
3534
3535 \ExplSyntaxOff
3536 % UF: removed Hy@writebookmark
3537 %     \Hy@currentbookmarklevel{0}
3538 %     \Hy@numberline
3539 %     \@@writetorep
3540 %     counter{bookmark@seq@number}
3541 % removed \HyPsd@SanitizeForOutFile, not needed
3542 % removed \currentpdfbookmark, defined by bookmark,
3543 % should use \newcommand there
3544 % removed \subpdfbookmark, defined by bookmark,
3545 % should use \newcommand there

```

```

3546 % removed \belowpdfbookmark, defined by bookmark,
3547 % should use \newcommand there
3548 % removed \pdfbookmark, defined by bookmark,
3549 % \BOOKMARK
3550 % \@BOOKMARK
3551 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3552 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3553 %% removed \ReadBookmarks / unneeded with bookmark.
3554 %% removed \Hy@OutlineName
3555 %% removed \check@bm@number
3556 %% removed \calc@bm@number
3557
3558 \ifHy@implicit
3559 \else
3560   \expandafter\endinput
3561 \fi
3562 \newlength\Hy@SectionHShift
3563 \def\Hy@SectionAnchorHref#1{%
3564   \ifx\protect\@typeset@protect
3565     \Hy@@SectionAnchor{#1}%
3566   \fi
3567 }
3568 \DeclareRobustCommand*\Hy@@SectionAnchor}[1]{%
3569   \leavevmode
3570   \hbox to 0pt{%
3571     \kern-\Hy@SectionHShift
3572     \Hy@raisedlink{%
3573       \hyper@anchorstart{#1}\hyper@anchorend
3574     }%
3575     \hss
3576   }%
3577 }
3578 \@ifundefined{hyper@nopatch@sectioning}
3579 {
3580   \let\H@old@ssect\@ssect
3581   \def\@ssect#1#2#3#4#5{%
3582     \Hy@MakeCurrentHrefAuto{section*}%
3583     \setlength{\Hy@SectionHShift}{#1}%
3584     \begingroup
3585       \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3586       \toks\tw@\expandafter{%
3587         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3588         #5%
3589       }%
3590     \edef\x{\endgroup
3591       \the\toks@\the\toks\tw@}%
3592   }\x
3593 }
3594 \let\H@old@schapter\@schapter
3595 \def\@schapter#1{%
3596   \begingroup
3597     \let\mkboth@gobbletwo
3598     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3599     \Hy@raisedlink{%

```

```

3600     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3601     }%
3602 \endgroup
3603 \H@old@schapter{#1}%
3604 }
3605 \@ifundefined{@chapter}{}{%
3606 \let\Hy@org@chapter\@chapter
3607 \def\@chapter{%
3608   \def\Hy@next{%
3609     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3610     \Hy@raisedlink{%
3611       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3612     }%
3613   }%
3614   \ifnum\c@secnumdepth>\m@ne
3615     \@ifundefined{if@mainmatter}%
3616     \iftrue{\csname if@mainmatter\endcsname}%
3617     \let\Hy@next\relax
3618   \fi
3619   \fi
3620   \Hy@next
3621   \Hy@org@chapter
3622 }%
3623 }
3624 \let\H@old@part\@part
3625 \begingroup\expandafter\expandafter\expandafter\endgroup
3626 \expandafter\ifx\csname chapter\endcsname\relax
3627   \let\Hy@secnum@part\z@
3628 \else
3629   \let\Hy@secnum@part\m@ne
3630 \fi
3631 \def\@part{%
3632   \ifnum\Hy@secnum@part>\c@secnumdepth
3633     \phantomsection
3634     \fi
3635   \H@old@part
3636 }
3637 \let\H@old@spart\@spart
3638 \def\@spart#1{%
3639   \Hy@MakeCurrentHrefAuto{part*}%
3640   \Hy@raisedlink{%
3641     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3642   }%
3643   \H@old@spart{#1}%
3644 }
3645 \let\H@old@sect\@sect
3646 \def\@sect#1#2#3#4#5#6[#7]#8{%
3647   \ifnum #2>\c@secnumdepth
3648     \expandafter\@firstoftwo
3649   \else
3650     \expandafter\@secondoftwo
3651   \fi
3652   {%
3653     \Hy@MakeCurrentHrefAuto{section*}%

```

```

3654 \setlength{\Hy@SectionHShift}{#3}%
3655 \begingroup
3656 \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3657 \toks\tw@\expandafter{%
3658 \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3659 #8%
3660 }%
3661 \edef\x{\endgroup
3662 \the\toks@{\the\toks\tw@}%
3663 }\x
3664 }{%
3665 \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3666 }%
3667 }
3668 }{}
3669 \expandafter\def\csname Parent-4\endcsname{}
3670 \expandafter\def\csname Parent-3\endcsname{}
3671 \expandafter\def\csname Parent-2\endcsname{}
3672 \expandafter\def\csname Parent-1\endcsname{}
3673 \expandafter\def\csname Parent0\endcsname{}
3674 \expandafter\def\csname Parent1\endcsname{}
3675 \expandafter\def\csname Parent2\endcsname{}
3676 \expandafter\def\csname Parent3\endcsname{}
3677 \expandafter\def\csname Parent4\endcsname{}
3678 %%
3679 \endpackage
3680 <@=hyp>
3681 <*colorscheme>
3682 % collected from https://tex.stackexchange.com/questions/525261/better-default-
3683 % colors-for-hyperref-links
3684 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3685 % linkcolor=
3686 %,filecolor=
3687 %,urlcolor=
3688 %,menucolor=
3689 %,runcolor=
3690 %,linkbordercolor=
3691 %,filebordercolor=
3692 %,urlbordercolor=
3693 %,menubordercolor=
3694 %,runbordercolor=
3695 \prop_const_from_keyval:cn { c_hyp_colorscheme_primary-colors_prop }
3696 {
3697 linkcolor = [rgb]{1,0,0}, %red
3698 filecolor = [rgb]{0,1,1}, %cyan
3699 urlcolor = [rgb]{1,0,1}, %magenta
3700 menucolor = [rgb]{1, 0, 0}, %red
3701 runcolor = [rgb]{0,1,1}, %cyan
3702 %-----
3703 linkbordercolor = [rgb]{1, 0 ,0 },
3704 filebordercolor = [rgb]{0, .5, .5},
3705 urlbordercolor = [rgb]{0, 1, 1},
3706 menubordercolor = [rgb]{1, 0, 0},

```

```

3707   runbordercolor = [rgb]{0, .7, .7}
3708 }
3709
3710 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3711 {
3712   linkcolor      = [rgb]{0,0.2,0.6},
3713   filecolor      = [rgb]{0.8,0,0.8},
3714   urlcolor       = [rgb]{0.8,0,0.8},
3715   menucolor      = [rgb]{0,0.2,0.6},
3716   runcolor       = [rgb]{0.8,0,0.8},
3717   %-----      %-----
3718   linkbordercolor = [rgb]{0,0.2,0.6},
3719   filebordercolor = [rgb]{0.8,0,0.8},
3720   urlbordercolor  = [rgb]{0.8,0,0.8},
3721   menubordercolor = [rgb]{0,0.2,0.6},
3722   runbordercolor  = [rgb]{0.8,0,0.8}
3723 }
3724
3725 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3726 { %two colors: intern/extern
3727   linkcolor      = [rgb]{0.79216, 0, 0.12549},
3728   filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3729   urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3730   menucolor      = [rgb]{0.79216, 0, 0.12549 },
3731   runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3732   %-----      %-----
3733   linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3734   filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3735   urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3736   menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3737   runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3738 }
3739
3740 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3741 { %all darkgray
3742   linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3743   filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3744   urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3745   menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3746   runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3747   %-----      %-----
3748   linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3749   filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3750   urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3751   menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3752   runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3753 }
3754
3755 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3756 { %dvipsnam.def
3757   linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3758   filecolor      = [rgb]{1, 0, 0}, %Red
3759   urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3760   menucolor      = [rgb]{1, 0, 0}, %Red

```

```

3761   runcolor          = [rgb]{1, 0, 0}, %Red
3762 %-----          %-----
3763   linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3764   filebordercolor = [rgb]{1, 0, 0}, %Red
3765   urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3766   menubordercolor = [rgb]{1, 0, 0}, %Red
3767   runbordercolor  = [rgb]{1, 0, 0} %Red
3768 }
3769
3770 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3771 { %dvipsnam.def
3772   linkcolor        = [rgb]{0.72, 0, 0}, %BrickRed
3773   filecolor        = [rgb]{0, 1, 0}, %Green
3774   urlcolor         = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3775   menucolor        = [rgb]{0.06, 0.46, 1}, %NavyBlue
3776   runcolor         = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3777 %-----          %-----
3778   linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3779   filebordercolor = [rgb]{0, 1, 0}, %Green
3780   urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3781   menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3782   runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3783 }
3784
3785
3786 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3787 {
3788   linkcolor        = [rgb]{0.50196, 0, 0.02353},
3789   filecolor        = [rgb]{0.07451, 0.09412, 0.46667},
3790   urlcolor         = [rgb]{0.54118, 0, 0.52941},
3791   menucolor        = [rgb]{0.44706, 0.45882, 0},
3792   runcolor         = [rgb]{0.07451, 0.46667, 0.46275},
3793 %-----          %-----
3794   linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3795   filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3796   urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3797   menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3798   runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3799 }
3800
3801 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3802 {
3803   linkcolor        = [rgb]{0,0,0},
3804   filecolor        = [rgb]{0,0,0},
3805   urlcolor         = [rgb]{0,0,0},
3806   menucolor        = [rgb]{0,0,0},
3807   runcolor         = [rgb]{0,0,0},
3808 %-----          %-----
3809   linkbordercolor = [rgb]{0,0,0},
3810   filebordercolor = [rgb]{0,0,0},
3811   urlbordercolor  = [rgb]{0,0,0},
3812   menubordercolor = [rgb]{0,0,0},
3813   runbordercolor  = [rgb]{0,0,0}
3814 }

```


3815 </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>	272, 793
<code>\\$</code>	271
<code>\%</code>	794
<code>\-</code>	2211
<code>\.</code>	575, 579, 581
@curropt commands:	
<code>\@curropt:</code>	2730
<code>\[</code>	2211
<code>\]</code>	23, 24, 25, 32, 33, 37, 47, 67, 74, 81, 88, 95, 110, 115, 116, 124, 125, 134, 135, 136, 137, 138, 145, 153, 273, 274, 944, 2732, 2797, 3134
<code>_</code>	573, 575, 579, 581
<code>\]</code>	2211
A	
<code>\A</code>	573, 2211
<code>\Acrobatmenu</code>	19, 166
<code>\addcontentsline</code>	14
<code>\AddToDocumentProperties</code>	381, 2206, 2309
<code>\AddToHook</code>	419, 444, 450
<code>\AddToHookNext</code>	191
<code>\advance</code>	2667, 2737, 2781, 2782
<code>allcolors</code> (hypersetup key)	<u>1072</u>
<code>\author</code>	2
B	
<code>\b</code>	577
<code>\begingroup</code>	202, 270, 368, 2573, 2697, 2726, 2738, 2772, 3156, 3584, 3596, 3625, 3655
<code>\belowpdfbookmark</code>	3546
<code>\bgroup</code>	267, 368
<code>\BOOKMARK</code>	3549
<code>\bookmarksetup</code>	436, 438, 439
<code>bookmarkstype</code> (hypersetup key)	14
bool commands:	
<code>\bool_if:N</code>	260, 289, 348, 717, 741, 760, 780, 788, 824, 890, 955, 1033, 1043, 1296, 1307
<code>\bool_if:nTF</code>	905, 1981, 2006, 3291, 3398
<code>\bool_lazy_and:nnTF</code>	416
<code>\bool_lazy_or:nnTF</code>	1326, 1375
<code>\bool_new:N</code>	196, 197, 559, 563, 567
<code>\bool_set_true:N</code>	568
<code>bordercolormodel</code> (hypersetup key)	14, <u>1093</u>
box commands:	
<code>\box_dp:N</code>	655, 2880, 2931, 2966
<code>\box_ht:N</code>	654, 2879, 2930, 2965
<code>\box_new:N</code>	464, 570
<code>\box_set_dp:Nn</code>	2178
<code>\box_set_ht:Nn</code>	2185
<code>\box_use:N</code>	1313, 2882, 2940, 2969
<code>\box_use_drop:N</code>	1318
<code>\box_wd:N</code>	653, 2878, 2929, 2964
<code>\l_tmpa_box</code>	2876, 2878, 2879, 2880, 2882, 2915, 2929, 2930, 2931, 2940, 2962, 2964, 2965, 2966, 2969
C	
<code>\catcode</code>	271, 272
<code>\char</code>	2814, 3011, 3023
<code>\chardef</code>	151
<code>\cite</code>	34
clist commands:	
<code>\clist_item:nn</code>	2212, 2241
<code>\clist_map_function:nN</code>	117, 126
<code>\clist_map_inline:nn</code>	2373
<code>\clist_map_variable:nNn</code> ..	2729, 2795
color commands:	
<code>\color_export:nnN</code> ..	41, 427, 992, 1114
<code>\color_select:n</code>	1000, 1036, 1317, 2921, 2925
<code>\color_select:nn</code>	1006
<code>\color_set:nn</code>	2, 386, 1017
<code>\color_set:nnn</code>	2, 385, 1023
color names:	
<code>hyp/annot/file</code>	<u>556</u>
<code>hyp/annot/link</code>	<u>556</u>
<code>hyp/annot/menu</code>	<u>556</u>
<code>hyp/annot/run</code>	<u>556</u>
<code>hyp/annot/url</code>	<u>556</u>
<code>colorfile</code> (hypersetup key)	<u>1072</u>
<code>colorlink</code> (hypersetup key)	<u>1072</u>
<code>colorlinks</code> (hypersetup key)	<u>1049</u>
<code>colormenu</code> (hypersetup key)	<u>1072</u>
<code>colorrun</code> (hypersetup key)	<u>1072</u>
<code>colorscheme</code> (hypersetup key)	1, <u>1484</u>
<code>colorurl</code> (hypersetup key)	<u>1072</u>
cs commands:	
<code>\cs_generate_variant:Nn</code>	148, 149, 150, 383, 463, 595, 603, 821, 992, 1009, 1026, 2212
<code>\cs_gset:Npn</code>	1288

<code>\cs_gset_eq:NN</code>	402, 413	dim commands:	
<code>\cs_if_exist_p:N</code>	417	<code>\dim_to_decimal_in_bp:n</code>	12, 154
<code>\cs_new:Npn</code>	110, 627	<code>\do</code>	2730
<code>\cs_new_protected:Npn</code>	202, 298, 315, 332, 358, 378, 393, 404, 457, 586, 590, 597, 599, 604, 668, 676, 685, 689, 711, 778, 822, 888, 953, 993, 1004, 1010, 1021, 1232, 2213, 2295	<code>\documentclass</code>	25
<code>\cs_set:Nn</code>	208, 250	<code>\DocumentMetadata</code>	1, 3, 11, 15, 16, 19, 24, 33, 40
<code>\cs_set_eq:NN</code>	154, 159, 160, 424, 635, 637, 793, 794, 1542	<code>\draft</code> (hypersetup key)	1508
<code>\cs_set_protected:Npn</code>	152, 173, 614, 715, 739, 758, 944		
<code>\csname</code>	2744, 3126, 3137, 3616, 3626, 3669, 3670, 3671, 3672, 3673, 3674, 3675, 3676, 3677		
<code>\currentpdfbookmark</code>	3542		
	D		
<code>\d</code>	575, 579, 581	<code>\edef</code> 2575, 2740, 2776, 2783, 3119, 3590, 3661	
<code>debug</code> (hypersetup key)	1508	<code>\EdefEscapeName</code>	2794, 2798
<code>\DeclareRobustCommand</code>	255, 283, 306, 323, 342, 371, 3568	<code>\egroup</code>	371
<code>\def</code>	12, 158, 368, 369, 371, 609, 2536, 2540, 2547, 2565, 2594, 2607, 2665, 2672, 2685, 2686, 2689, 2690, 2703, 2704, 2720, 2721, 2724, 2725, 2763, 2792, 2842, 2856, 2857, 2869, 2897, 2899, 2900, 2910, 2945, 2947, 2948, 2959, 2978, 2980, 2981, 2983, 2984, 2993, 3037, 3104, 3114, 3116, 3121, 3133, 3144, 3155, 3166, 3209, 3217, 3229, 3281, 3315, 3350, 3388, 3438, 3473, 3510, 3517, 3524, 3531, 3563, 3581, 3595, 3607, 3608, 3631, 3638, 3646, 3669, 3670, 3671, 3672, 3673, 3674, 3675, 3676, 3677	<code>\else</code>	2572, 2693, 2747, 2751, 2754, 2771, 2779, 3107, 3112, 3129, 3135, 3147, 3151, 3168, 3172, 3176, 3180, 3184, 3188, 3192, 3196, 3200, 3204, 3220, 3224, 3324, 3325, 3326, 3328, 3331, 3335, 3339, 3345, 3362, 3366, 3370, 3376, 3380, 3447, 3448, 3450, 3453, 3457, 3482, 3483, 3485, 3488, 3492, 3559, 3628, 3649
<code>\DefaultHeightofCheckBox</code>	2984	<code>\endcsname</code>	2758, 3103, 3126, 3137, 3616, 3626, 3669, 3670, 3671, 3672, 3673, 3674, 3675, 3676, 3677
<code>\DefaultHeightofChoiceMenu</code>	2725	<code>\endgroup</code>	276, 279, 369, 371, 2576, 2717, 2740, 2786, 2789, 3163, 3590, 3602, 3625, 3661
<code>\DefaultHeightofReset</code>	2948	<code>\endinput</code>	3560
<code>\DefaultHeightofSubmit</code>	2900	exp commands:	
<code>\DefaultHeightofText</code>	2694	<code>\exp_args:Ne</code>	831, 2323, 2567, 3211
<code>\DefaultHeightofTextMultiline</code>	2692	<code>\exp_args:Nne</code>	2338, 2520
<code>\DefaultOptionsofCheckBox</code>	2987	<code>\exp_args:Nno</code>	317, 334
<code>\DefaultOptionsofPushButton</code>	2861	<code>\exp_args:Nno</code>	300, 360, 2281
<code>\DefaultOptionsofReset</code>	2952	<code>\exp_args:NNV</code>	1966, 1997, 2242
<code>\DefaultOptionsofSubmit</code>	2904	<code>\exp_args:NnV</code>	670, 679
<code>\DefaultOptionsofText</code>	2699	<code>\exp_args:No</code>	703, 707, 2859, 2902, 2950, 2985
<code>\DefaultWidthofCheckBox</code>	2983	<code>\exp_args:NV</code>	651, 658
<code>\DefaultWidthofChoiceMenu</code>	2724	<code>\exp_not:N</code>	944
<code>\DefaultWidthofReset</code>	2947	<code>\exp_not:n</code>	1423, 1457, 1549, 1604, 1625, 1662, 1682, 1705, 1749, 1782, 1807, 1858, 1883, 1912, 1961, 2038, 2066, 2093, 2113, 2359, 2465, 2493, 2503, 2514
<code>\DefaultWidthofSubmit</code>	2899	<code>\expandafter</code>	273, 2698, 2732, 2741, 2743, 2744, 2797, 3120, 3136, 3560, 3586, 3587, 3625, 3626, 3648, 3650, 3657, 3658, 3669, 3670, 3671, 3672, 3673, 3674, 3675, 3676, 3677
<code>\DefaultWidthofText</code>	2689	<code>\ExplSyntaxOff</code>	15, 2604, 2657, 3101, 3314, 3436, 3535
<code>destlabel</code> (hypersetup key)	14	<code>\ExplSyntaxOn</code>	13, 16, 2605, 2662, 3208, 3349, 3472
		<code>extension</code> (hypersetup key)	14, 1526

F	
<code>\fbox</code>	2833, 2921, 2925
<code>\fi</code>	12, 2561, 2588, 2591, 2600, 2695, 2703, 2704, 2735, 2753, 2756, 2757, 2763, 2766, 2775, 2777, 2784, 2787, 3109, 3131, 3141, 3143, 3149, 3153, 3160, 3161, 3170, 3174, 3178, 3182, 3186, 3190, 3194, 3198, 3202, 3206, 3222, 3226, 3324, 3325, 3326, 3333, 3337, 3341, 3343, 3345, 3364, 3368, 3372, 3376, 3383, 3447, 3448, 3455, 3459, 3461, 3482, 3483, 3490, 3494, 3496, 3561, 3566, 3618, 3619, 3630, 3634, 3651
file (hypersetup key)	10, 1557
file commands:	
<code>\file_input:n</code>	14
fileborderstyle (hypersetup key)	14, 1186
filecolor (hypersetup key)	1072
final (hypersetup key)	1508
<code>\fontencoding</code>	2810, 3007, 3019
<code>\fontfamily</code>	2809, 3006, 3018
<code>\fontseries</code>	2811, 3008, 3020
<code>\fontshape</code>	2812, 3009, 3021
fp commands:	
<code>\fp_eval:n</code>	2136, 2164, 2180
G	
<code>\gdef</code>	273, 274
<code>\GetDocumentProperties</code>	3
<code>\global</code>	2667, 3099
group commands:	
<code>\group_begin:</code>	258, 287, 310, 327, 346, 723, 733, 747, 754, 785, 810, 829, 878, 895, 937, 960, 986, 1035, 1299, 1316, 2444, 2808, 2858, 2901, 2949, 2982, 3005, 3017
<code>\group_end:</code>	301, 318, 335, 363, 729, 733, 765, 769, 807, 810, 875, 878, 935, 937, 984, 986, 1045, 1319, 1322, 2451, 2815, 2894, 2942, 2975, 3012, 3024, 3034
H	
<code>\hbox</code>	368, 3570
hbox commands:	
<code>\hbox_overlap_right:n</code>	1313
<code>\hbox_set:Nn</code>	2876, 2915, 2962
<code>\hbox_set:Nw</code>	1300
<code>\hbox_set_end:</code>	1309
<code>\hbox_set_to_wd:Nnn</code>	2162
hidefile (hypersetup key)	1460
hidelink (hypersetup key)	1460
hidelinks (hypersetup key)	1460
hidemenu (hypersetup key)	1460
hiderun (hypersetup key)	1460
hideurl (hypersetup key)	1460
hook commands:	
<code>\hook_gput_code:nnn</code>	395, 406, 1029, 1039, 1292, 1303
<code>\hook_new:n</code>	384, 667
<code>\hook_new_pair:nn</code>	253, 281, 305, 322, 340
<code>\hook_use:n</code>	257, 277, 286, 302, 309, 319, 326, 336, 345, 364, 673, 682, 726, 750
<code>\href</code>	4, 5, 20, 255
<code>\hreflaunch</code>	5
<code>\hrefpdf</code>	5, 20, 306, 376
<code>\hrefrun</code>	5, 323, 377
<code>\hrefurl</code>	5, 20, 283, 375
<code>\hss</code>	3575
hyp commands:	
<code>\l_hyp_annot_colorfile_bool</code> ..	8, 557
<code>\l_hyp_annot_colorlink_bool</code> ..	8, 557
<code>\l_hyp_annot_colormenu_bool</code> ..	8, 557
<code>\l_hyp_annot_colorrund_bool</code> ..	8, 557
<code>\l_hyp_annot_colorurl_bool</code> ..	8, 557
<code>\l_hyp_annot_ocgcolorfile_bool</code>	8, 561
<code>\l_hyp_annot_ocgcolorlink_bool</code>	8, 561
<code>\l_hyp_annot_ocgcolormenu_bool</code>	8, 561
<code>\l_hyp_annot_ocgcolorrund_bool</code>	8, 561
<code>\l_hyp_annot_ocgcolorurl_bool</code>	8, 561
<code>\l_hyp_current_dest_name_tl</code>	666 , 724 , 748
hyp internal commands:	
<code>\g__hyp_AcroForm_CoFields_prop</code> .	2531 , 2549 , 2618 , 2620 , 2635
<code>\g__hyp_AcroForm_Fields_prop</code> ...	2532 , 2542 , 2613
<code>\l_hyp_annot_GoTo_bool</code>	717, 741, 760
<code>\l_hyp_annot_GoTo_bool_uuuu\l_-</code>	<code>_hyp_annot_URI_bool_uuuu\l_-</code>
<code>hyp_annot_GoToR_bool_uuuu\l_-</code>	<code>hyp_annot_Named_bool_uuuu\l_-</code>
<code>hyp_annot_Launch_bool</code>	565
<code>\l_hyp_annot_GoToR_bool</code>	824
<code>\l_hyp_annot_Launch_bool</code>	890
<code>\l_hyp_annot_Named_bool</code>	955
<code>\c__hyp_annot_types_seq</code>	477 , 557 , 561 , 1072 , 1373 , 1471
<code>\l_hyp_annot_URI_bool</code>	780
<code>\g__hyp_bordercolormodel_str</code> ...	429 , 556 , 1097 , 1116
<code>__hyp_check_link_nesting:TF</code> ...	627 , 635 , 637 , 720 , 744 , 762 , 783 , 827 , 893 , 958

_hyp_citebordercolor_hook_- init:	391, 404, 413	_hyp_href_url_aux:nn	296, 298
_hyp_citecolor_hook_init:		\l_hyp_href_url_encode_bool	
.	389, 393, 402	196, 207, 245, 260, 289, 348
_hyp_clist_display:n	110, 117, 126	_hyp_href_url_format:	
_hyp_color_select:n	202, 208, 250, 361
.	41, 993, 993, 1009	\l_hyp_href_url_ismap_bool	
_hyp_color_select_aux:wn	197, 221, 788
.	993, 997, 1004	\l_hyp_href_url_protocol_tl	
_hyp_color_set:nn	41, 389,	198, 209, 248, 300, 362
391, 426, <u>1010</u> , 1010, 1026, 1077, 1113		_hyp_if_outer_link:	621
_hyp_color_set_aux:nwn		_hyp_if_outer_link:TF	637
.	<u>1010</u> , 1014, 1021	\l_hyp_link_Contents_tl	
\c_hyp_colorscheme_daleif_prop	3710	512, 524, 526, 533, 535, 542, 544
\c_hyp_colorscheme_henryford_- prop	3801	_hyp_link_goto_begin:nw	
\c_hyp_colorscheme_julian_prop	3725	689, 727, 751
\c_hyp_colorscheme_phelype_prop	3786	_hyp_link_goto_end:	711, 728, 764
\c_hyp_colorscheme_szabolcsA_- prop	3755	\g_hyp_linknestlevel_int	
\c_hyp_colorscheme_szabolcsB_- prop	3770	620, 623, 719, 735, 743, 771,
\c_hyp_colorscheme_tivv_prop	3740	782, 812, 826, 880, 892, 938, 957, 987
\l_hyp_dest_box	33,	g_hyp_linknestlevel_int	<u>620</u>
<u>570</u> , 653, 654, 655, 2162, 2178, 2185		\c_hyp_map_annot_hyp_prop	<u>477</u>
\l_hyp_dest_name_tmpa_tl	470,	\c_hyp_map_hyp_annot_prop	
692, 693, 699, 703, 705, 708, 847, 861		<u>477</u> , 1027,
\l_hyp_dest_pdfremotestartview_- tl	496, 856, 1969, 1973	1100, 1140, 1167, 1175, 1186, 1213,
\g_hyp_dest_pdfstartpage_tl	1221, 1290, 1398, 1434, 1444, 1557
.	496, 1980, 1982, 1989, 2007, 2014	_hyp_ocg_init: <u>1232</u> , 1232, 1288, 1298	
\g_hyp_dest_pdfstartview_tl		\l_hyp_optlang_regex	2210, 2211, 2243
.	<u>496</u> , 1982, 1989, 2000, 2004, 2007, 2014	l_hyp_page/Trans	<u>584</u>
\l_hyp_dest_pdfview_tl		_hyp_PageLabels_gpust: <u>604</u> , 604, 617	
.	555, 671, 680,	\l_hyp_para_tmpa_tl	473, 904, 907, 918
2131, 2134, 2141, 2144, 2145, 2146,		\l_hyp_para_tmpa_tl_uuu\l_hyp_- text_tmpa_str_uuu\g_hyp_text_- tmpa_str	<u>470</u>
2147, 2148, 2149, 2154, 2158, 2194		_hyp_property_record:nn	
\c_hyp_dest_startview_regex	<u>456</u> , 457, 463, 2668
.	571, 1967, 1998	_hyp_secondoftwoithopt:wnn	
\c_hyp_dest_undefined_tl	374, 375, 376, 377
.	476, 698, 699	_hyp_setup_info_date_key:nn	
_hyp_destination:nn	2295, 2328, 2329
.	33, <u>640</u> , 640, 671, 680	_hyp_setup_info_key:nn	2213,
\l_hyp_filename_tmpa_tl	470,	2288, 2289, 2290, 2291, 2293, 2294
830, 832, 837, 838, 843, 899, 900, 918		_hyp_store_metadata:nn	
_hyp_href_pdf_aux:nn	312, 315	378, 383, 1576, 2220,
\l_hyp_href_pdf_destination_tl	2263, 2308, 2323, 2332, 2348, 2418
.	199, 210, 249, 317	\l_hyp_text_enc_dest_print_tl	
\l_hyp_href_pdf_page_tl	200, 218, 855	<u>499</u> , 846
_hyp_href_run_aux:nn	329, 332	\l_hyp_text_enc_dest_tl	
\l_hyp_href_run_parameter_tl	33, <u>499</u> , 647, 704
.	201, 222, 334	\l_hyp_text_enc_file_print_tl	
_hyp_href_url_aux:n	355, 358	<u>499</u> , 898
		\l_hyp_text_enc_info_print_tl	
		<u>499</u> , 523, 532, 541, 601
		\l_hyp_text_enc_para_print_tl	
		<u>499</u> , 903

\l__hyp_text_enc_uri_print_tl ..	fileborderstyle	14, 1186
..... 262,	filecolor	1072
265, 291, 294, 350, 353, 499, 786, 1568	final	1508
__hyp_text_pdfstring:nnN	hidefile	1460
. 521, 530, 539, 590, 590, 595, 598,	hidelink	1460
601, 645, 702, 786, 844, 896, 901, 1568	hidelinks	1460
__hyp_text_pdfstring_info:nN ..	hidemenu	1460
599, 599, 603, 2246, 2249, 2269, 2278	hiderun	1460
__hyp_text_string_from_unicode:nN	hideurl	1460
..... 586, 586	hypertextnames	1526
\g__hyp_text_tmpa_str	link	10, 1557
475	linkborder	14
\l__hyp_text_tmpa_str	linkborderstyle	14, 1186
474	linkcolor	1072
\l__hyp_tmpa_box 464, 1300, 1313, 1318	linkfileprefix	1526
\l__hyp_tmpa_int	linktoc	1526
464	linktocpage	1526
\l__hyp_tmpa_seq	localanchorname	1526
.....	menu	10, 1557
.... 464, 1967, 1969, 1998, 2000,	menuborder	14
2121, 2122, 2126, 2128, 2152, 2159,	menuborderstyle	14, 1186
2160, 2161, 2170, 2172, 2182, 2187,	menucolor	1072
2243, 2244, 2249, 2622, 2624, 2630	naturalnames	1526
\l__hyp_tmpa_str	nested-links	10
.....	nesting	14
.... 464, 2246, 2249, 2251, 2253,	ocgcolorfile	1326
2269, 2270, 2272, 2278, 2279, 2283	ocgcolorlink	1326
\l__hyp_tmpa_tl	ocgcolorlinks	1326
.....	ocgcolormenu	1326
.. 464, 648, 652, 660, 1117, 1121,	ocgcolorrun	1326
1568, 1569, 1574, 1965, 1967, 1996,	ocgcolorurl	1326
1998, 2128, 2129, 2136, 2219, 2220,	pageanchor	1526
2221, 2241, 2246, 2262, 2263, 2269	pdfauthor	2210
\l__hyp_tmpb_tl	pdfborder	14
464, 2241, 2243	pdfborderstyle	14, 1186
\l__hyp_uri_tmpa_tl	pdfcreationdate	15, 2295
470, 786, 787	pdfcreator	2210
hyp/anchor	pdfencoding	1498
666	pdfinfo	2362
hyp/annot/file (color name)	pdfkeywords	2210
556	pdflang	15, 2199
hyp/annot/link (color name)	pdflinkmargin	15
556	pdfmetadate	15, 2295
hyp/annot/menu (color name)	pdfmoddate	15, 2295
556	pdfproducer	2210
hyp/annot/run (color name)	pdfremotestartview	12
556	pdfstartview	12
hyp/annot/url (color name)	pdfsubject	2210
556	pdftitle	2210
\hypercalcbp	pdftrapped	2334
12, 19, 154	pdfversion	1498
\HyperDestNameFilter	pdfview	12, 2117
13, 646, 703	plainpages	1526
\hypersetup 2, 3, 6, 9, 10, 14, 19, 69, 102, 173	run	10, 1557
\hypersetup keys:		
allcolors		1072
bookmarkstyle		14
bordercolormodel		14, 1093
colorfile		1072
colorlink		1072
colorlinks		1049
colormenu		1072
colorrun		1072
colorscheme		1, 1484
colorurl		1072
debug		1508
destlabel		14
draft		1508
extension		14, 1526
file		10, 1557

runborder	14	\keys_set_known:nn	2523
runborderstyle	14, 1186	\kvsetkeys	175, 2609
runcolor	1072		
unicode	1498		
url	10, 1557	L	
urlborder	14	\label	9, 14
urlborderstyle	14, 1186	\LayoutCheckField	2995
urlcolor	1072	\LayoutChoiceField	2767
verbose	1508	\LayoutPushButtonField	2871, 2888
hypertextnames (hypersetup key)	1526	\LayoutTextField	2705
		\leavevmode	2706, 2801, 3569
		legacy commands:	
		\legacy_if:nTF	161, 442, 452, 612, 2647, 2867, 2908, 2958, 2991, 3271
		\let	375, 376, 377, 737, 802, 814, 871, 882, 931, 940, 989, 2534, 2535, 2562, 2658, 2659, 2660, 2687, 2688, 2722, 2723, 2765, 3099, 3106, 3108, 3113, 3130, 3164, 3580, 3594, 3597, 3606, 3617, 3624, 3627, 3629, 3637, 3645
		link (hypersetup key)	10, 1557
		linkborder (hypersetup key)	14
		linkborderstyle (hypersetup key)	14, 1186
		linkcolor (hypersetup key)	1072
		linkfileprefix (hypersetup key)	1526
		linktoc (hypersetup key)	1526
		linktocpage (hypersetup key)	1526
		localanchorname (hypersetup key)	1526
		\long	12
		M	
		\MakeButtonField	2876, 2891, 2915, 2962, 2973
		\MakeChoiceField	2851
		\MakeFieldObject	2525
		\MakeRadioField	2833
		\MakeTextField	2714
		\mbox	1310
		menu (hypersetup key)	10, 1557
		menuborder (hypersetup key)	14
		menuborderstyle (hypersetup key)	14, 1186
		menucolor (hypersetup key)	1072
		mode commands:	
		\mode_if_horizontal:TF	642, 664
		\mode_leave_vertical:	256, 285, 308, 325, 344, 691, 796, 863, 920, 963, 2845, 2873, 2890, 2912, 2954, 2997
		msg commands:	
		\msg_error:nn	418, 2887, 2972
		\msg_info:nnn	2227, 2232
		\msg_line_context:	75
		\g_msg_module_name_prop	17
		\msg_new:nnn	43, 50, 55, 59, 63, 70, 77, 84, 91, 98, 104, 111, 120, 130, 141
I			
\ifcase	3324, 3447, 3482		
\ifcsname	3103		
\ifdim	2735, 2774, 2775		
\ifnum	3157, 3324, 3330, 3361, 3447, 3452, 3482, 3487, 3614, 3632, 3647		
\iftrue	3616		
\ifx	2571, 2704, 2764, 3134, 3146, 3150, 3167, 3171, 3175, 3179, 3183, 3187, 3191, 3195, 3199, 3203, 3219, 3223, 3325, 3326, 3334, 3338, 3345, 3365, 3369, 3376, 3377, 3448, 3456, 3483, 3491, 3564, 3626		
\immediate	2555, 2558		
int commands:			
\int_compare:nNnTF	623, 2126, 2152, 2834, 3248, 3300, 3409, 3431		
\int_compare_p:nNn	3293, 3400		
\int_eval:n	854		
\int_gdecr:N	735, 771, 812, 880, 938, 987		
\int_gincr:N	719, 743, 782, 826, 892, 957		
\int_max:nn	855		
\int_new:N	468, 620		
iow commands:			
\iow_newline:	3052, 3058, 3071		
K			
\kern	3571		
keys commands:			
\keys_define:nn	179, 187, 205, 243, 387, 433, 631, 1049, 1074, 1081, 1093, 1102, 1127, 1142, 1161, 1188, 1207, 1330, 1344, 1356, 1379, 1385, 1400, 1429, 1460, 1473, 1484, 1498, 1508, 1513, 1526, 1538, 1559, 1564, 2117, 2199, 2215, 2258, 2297, 2312, 2330, 2334, 2362, 2416, 2422, 2435, 2455		
\l_keys_key_str	184, 2283, 2520		
\keys_set:nn	176, 213, 232, 259, 288, 311, 328, 347, 397, 408, 1490, 1497, 2339, 2366, 2369, 2370, 2371, 2372, 2445, 2520		

<code>\msg_new:nmnn</code>	18, 28	<code>\pdf_pageobject_ref:n</code>	150, 1989, 2014, 2676
<code>\msg_warning:nn</code>	163, 1504	<code>\pdf_purify:nN</code>	30, 592
<code>\msg_warning:nnn</code>	172, 695, 981	<code>\pdf_string_from_unicode:nnN</code>	588, 593
<code>\msg_warning:nmnn</code>	1348, 1389, 1602, 1646, 1680, 1703, 1765, 1794, 1828, 1843, 1856, 1869, 1894, 1929, 1945, 1972, 2003, 2036, 2050, 2077, 2111, 2193	<code>\pdf_version:</code>	1650, 1769, 1798, 1832, 1847, 1873, 1898, 1933, 1949, 2054, 2081
<code>\msg_warning:nmnnn</code>	183, 1420, 1454, 1546, 1622, 1659, 1746, 1779, 1804, 1880, 1907, 1958, 2063, 2090, 2356, 2462, 2481, 2500, 2511	<code>\pdf_version_compare:NnTF</code>	1640, 1731, 1753, 1789, 1822, 1837, 1863, 1888, 1916, 1939, 2044, 2071
N		<code>\pdf_version_compare_p:Nn</code>	909, 1327, 1376
<code>naturalnames</code> (hypersetup key)	1526	<code>\pdf_version_major:</code>	160, 1328, 1351, 1377, 1393
<code>nested-links</code> (hypersetup key)	10	<code>\pdf_version_minor:</code> ..	159, 1351, 1393
<code>nesting</code> (hypersetup key)	14	pdfannot commands:	
<code>\newcommand</code>	157, 3543, 3545, 3547	<code>\pdfannot_box:nmnn</code>	2709, 2818, 2848, 2877, 2928, 2963, 3027
<code>\newcount</code>	2663, 2841	<code>\pdfannot_box_ref_last:</code> ..	2569, 2585
<code>\NewDocumentCommand</code>	2525	<code>\pdfannot_dict_put:nnn</code>	525, 534, 543, 797, 865, 921, 966, 1118, 1153, 1177, 1199, 1223, 1405, 1436
<code>\NewExpandableDocumentCommand</code>	374	<code>\pdfannot_dict_remove:nn</code> ..	1108, 1148, 1169, 1194, 1215, 1413, 1446
<code>\newlength</code>	3562	<code>\pdfannot_link:nnn</code> ..	798, 866, 922, 967
<code>\noexpand</code> ...	2741 , 2742 , 2743 , 2744 , 3117	<code>\pdfannot_link_goto_begin:nw</code> ...	708
<code>\nolinkurl</code>	4	<code>\pdfannot_link_goto_end:</code>	713
O		<code>\pdfannot_link_margin:n</code>	8, 1684
<code>ocgcolorfile</code> (hypersetup key)	1326	<code>\c_pdfannot_link_types_seq</code>	565
<code>ocgcolorlink</code> (hypersetup key)	1326	pdfauthor (hypersetup key)	2210
<code>ocgcolorlinks</code> (hypersetup key)	1326	<code>\pdfbookmark</code>	3548
<code>ocgcolormenu</code> (hypersetup key)	1326	pdfborder (hypersetup key)	14
<code>ocgcolorrun</code> (hypersetup key)	1326	<code>\pdfborderstyle</code> (hypersetup key)	14 , 1186
<code>ocgcolorurl</code> (hypersetup key)	1326	<code>\pdfcreationdate</code> (hypersetup key)	15 , 2295
P		<code>\pdfcreator</code> (hypersetup key)	2210
<code>pageanchor</code> (hypersetup key)	1526	<code>\pdfdest</code>	12
<code>\paperwidth</code>	4	pdfdict commands:	
<code>\PassOptionsToPackage</code> ...	454 , 1518 , 1523	<code>\pdfdict_new:n</code> .	584, 774, 817, 884, 949
pdf commands:		<code>\pdfdict_put:nmnn</code>	230, 237, 585, 775, 776, 787, 790, 818, 819, 834, 840, 851, 861, 885, 886, 900, 915, 950, 951, 964, 1710, 1711, 1715, 1716, 2459, 2469, 2473, 2476, 2478, 2497, 2506, 2508
<code>\pdf_bdcobject:nn</code>	1312, 1315	<code>\pdfdict_remove:nn</code>	227, 912, 1720, 1721
<code>\pdf_destination:nn</code>	28, 148, 659	<code>\pdfdict_use:n</code>	797, 864, 921, 927, 966, 971, 2448
<code>\pdf_destination:nmnn</code>	652	pdfencoding (hypersetup key)	1498
<code>\pdf_emc:</code>	1314, 1320	<code>\pdfescapestring</code>	3106
<code>\pdf_name_from_unicode_e:n</code>	238, 834, 965	pdffile commands:	
<code>\pdf_object_if_exist:nTF</code>	832	<code>\pdffile_embed_file:nmnn</code> ...	821, 835
<code>\pdf_object_new:n</code>	1234, 1235, 1236, 1237, 3039, 3040, 3041	pdfinfo (hypersetup key)	2362
<code>\pdf_object_ref:n</code>	149, 843, 1240, 1242, 1264, 1265, 1268, 1272, 1277, 1282, 1287, 2640, 2642, 3097	pdfkeywords (hypersetup key)	2210
<code>\pdf_object_ref_last:</code>	865, 2450	pdflang (hypersetup key)	15 , 2199
<code>\pdf_object_unnamed_write:nn</code> ...	864, 2446		
<code>\pdf_object_write:nmnn</code>	1238, 1244, 1254, 1266, 3042, 3084, 3091		

pdflinkmargin (hypersetup key)	15	\prg_new_conditional:Npnn	621
pdfmanagement commands:		\prg_return_false:	624
\pdfmanagement_add:nn	1693	\prg_return_true:	625
\pdfmanagement_add:nnn		prop commands:	
.... 606, 1264, 1265, 1287, 1574, 1594, 1610, 1614, 1634, 1642, 1671, 1726, 1736, 1760, 1774, 1786, 1791, 1816, 1824, 1839, 1865, 1890, 1924, 1941, 1987, 2012, 2026, 2046, 2073, 2102, 2205, 2228, 2233, 2253, 2272, 2282, 2306, 2321, 2343, 2431, 2450, 2615, 2632, 2639, 2641, 2643, 2649		\prop_const_from_keyval:Nn	479, 487, 3695, 3710, 3725, 3740, 3755, 3770, 3786, 3801
\pdfmanagement_if_active_p:	417	\prop_gput:Nnn	17, 2542, 2549
\pdfmanagement_remove:nn		\prop_if_empty:NTF	2618
..... 1571, 1590, 1598, 1618, 1630, 1655, 1667, 1675, 1689, 1698, 1742, 1757, 1776, 1813, 1852, 1877, 1903, 1920, 1954, 1984, 2009, 2022, 2031, 2059, 2086, 2098, 2107, 2237, 2266, 2303, 2318, 2428, 2441, 2653		\prop_item:Nn	2635
\pdfmanagement_show:n	2616	\prop_map_inline:Nn	1027, 1100, 1140, 1167, 1175, 1186, 1213, 1221, 1290, 1398, 1434, 1444, 1488, 1557, 2613, 2620
pdfmeta commands:		\prop_new:N	2531, 2532
\pdfmeta_standard_family:nn	546	property commands:	
\pdfmeta_standard_verify:nnTF	961, 2864, 2955	\property_if_recorded:nn	2671
\pdfmeta_standard_verify:nTF	548, 2645, 3213	\property_if_recorded:nnTF	2674
pdfmetadate (hypersetup key)	15, 2295	\property_record:nn	460
pdfmoddate (hypersetup key)	15, 2295	\property_ref:nn	2678
pdfproducer (hypersetup key)	2210	\property_ref_undefined_warn:nn	2669
pdfremotestartview (hypersetup key)	12	\protect	737, 802, 814, 871, 882, 931, 940, 989, 3564
pdfstartview (hypersetup key)	12	\providecommand	155, 156, 2556, 2559, 2719, 3102
\pdfstringdef	9, 10, 23, 3162	\ProvidesFile	3, 7
\pdfstringdefDisableCommands	375, 376, 377		
pdfsubject (hypersetup key)	2210		
pdftitle (hypersetup key)	2210	R	
pdftrapped (hypersetup key)	2334	\ReadBookmarks	3553
pdfversion (hypersetup key)	1498	\refstepcounter	13
pdfview (hypersetup key)	12, 2117	regex commands:	
pdfxform commands:		\regex_const:Nn	571
\pdfxform_if_exist:nTF	2804, 2916, 3000	\regex_extract_once:NnN	2243
\pdfxform_new:nnn	2527, 2806, 2919, 2923, 3002, 3014	\regex_extract_once:NnNTF	1967, 1998
\pdfxform_ref:n	2828, 2829, 2935, 2936, 3243, 3244	\regex_new:N	2210
\phantom	3023	\regex_set:Nn	2211
\phantomsection	13, 20, 3633	\relax	737, 802, 814, 871, 882, 931, 940, 989, 2722, 2723, 2764, 3099, 3146, 3150, 3219, 3223, 3325, 3326, 3334, 3338, 3365, 3369, 3448, 3456, 3483, 3491, 3617, 3626
plainpages (hypersetup key)	1526	\RemoveFromHook	1580
prg commands:		\RenewDocumentCommand	166
\prg_do_nothing:	402, 413	\RequirePackage	11, 26, 446, 3551
\prg_generate_conditional_variant:Nnn	2671	run (hypersetup key)	10, 1557
		runborder (hypersetup key)	14
		runborderstyle (hypersetup key)	14, 1186
		runcolor (hypersetup key)	1072
		S	
		\selectfont	2813, 3010, 3022
		seq commands:	
		\seq_const_from_clist:Nn	477
		\seq_count:N	2126, 2152

<code>\calc@bm@number</code>	3556	<code>\Fld@pageobjref</code>	2660, 2672, 3102, 3233,
<code>\check@bm@number</code>	3555		3285, 3318, 3354, 3392, 3441, 3476
<code>\define@key</code>	421	<code>\Fld@radiosymbol</code>	3373
<code>\Fld@additionalactions</code>	3209, 3278,	<code>\Fld@rotation</code>	3248,
	3311, 3347, 3384, 3430, 3470, 3499		3250, 3293, 3300, 3302, 3324, 3330,
<code>\Fld@align</code>	3237, 3322, 3396		3332, 3361, 3363, 3400, 3409, 3411,
<code>\Fld@altname</code> ...	3146, 3148, 3219, 3221		3447, 3452, 3454, 3482, 3487, 3489
<code>\Fld@annotflags</code>	3232,	<code>\Fld@submitflags</code>	3468
	3284, 3317, 3353, 3391, 3440, 3475	<code>\Fld@validate@code</code>	3175, 3177
<code>\Fld@annotnames</code>	3217, 3234,	<code>\Fld@value</code>
	3286, 3319, 3355, 3393, 3442, 3477		2688, 2704, 2723, 2764, 2765, 3429
<code>\Fld@bcolor</code>	3256, 3258, 3338,	<code>\Fld@width</code>	2689, 2703,
	3340, 3369, 3371, 3404, 3417, 3419		2710, 2714, 2724, 2763, 2774, 2776,
<code>\fld@bcolor</code>	3326		2819, 2833, 2849, 2851, 2869, 2899,
<code>\Fld@bordercolor</code>	3252,		2910, 2947, 2959, 2983, 2993, 3028
	3254, 3295, 3304, 3306, 3325, 3334,	<code>\Fld@X@additionalactions</code>
	3336, 3365, 3367, 3402, 3413, 3415,		3166, 3211, 3214
	3448, 3456, 3458, 3483, 3491, 3493	<code>\Form@action</code>	3466
<code>\Fld@borderstyle</code>	3238,	<code>\H@old@part</code>	3624, 3635
	3290, 3323, 3359, 3397, 3446, 3497	<code>\H@old@chapter</code>	3594, 3603
<code>\Fld@borderwidth</code>	2781, 2782, 3238,	<code>\H@old@sect</code>	3645, 3656, 3665
	3290, 3323, 3359, 3397, 3446, 3497	<code>\H@old@spart</code>	3637, 3643
<code>\Fld@calculate@code</code> .	2571, 3179, 3181	<code>\H@old@ssect</code>	3580, 3585
<code>\Fld@calculate@sortkey</code>	2582	<code>\href@</code>	267, 273
<code>\Fld@cbsymbol</code>	3260	<code>\href@split</code>	273, 274
<code>\Fld@charsize</code>	<code>\Hy@@escapeform</code>	3120, 3133, 3139
	2780, 3264, 3344, 3375, 3425, 3481	<code>\Hy@@SectionAnchor</code>	3565, 3568
<code>\Fld@checkequals</code>	2732, 2797	<code>\Hy@abspage</code>	616
<code>\Fld@choices</code>	3346	<code>\Hy@activeanchorfalse</code>	687
<code>\Fld@color</code>	3265, 3267, 3345, 3376, 3426	<code>\Hy@activeanchortrue</code>	678
<code>\Fld@default</code>	2687, 2704, 2722, 2765,	<code>\Hy@AtBeginDocument</code>	2536, 2552
	2794, 2981, 3377, 3381, 3382, 3428	<code>\Hy@bookmarkstype</code>	1586
<code>\Fld@flags</code>	3236,	<code>\Hy@chapapp</code>	3598, 3609
	3288, 3321, 3357, 3395, 3444, 3479	<code>\Hy@colorlink</code>	35
<code>\Fld@format@code</code>	3171, 3173	<code>\Hy@currentbookmarklevel</code>	3537
<code>\Fld@height</code>	2690,	<code>\Hy@DisableOption</code>	165
	2711, 2714, 2725, 2783, 2820, 2833,	<code>\Hy@drafttrue</code>	1517
	2849, 2851, 2900, 2948, 2984, 3029	<code>\Hy@escapeform</code>
<code>\Fld@keystroke@code</code>	3167, 3169		2708, 2803, 2847, 2875,
<code>\Fld@listcount</code> .	2793, 2799, 2834, 2841		2914, 2961, 2999, 3104, 3111, 3114
<code>\Fld@mappingname</code>	3150, 3152, 3223, 3225	<code>\Hy@escapestring</code>	3106,
<code>\Fld@maxlen</code>	3431, 3433		3108, 3113, 3116, 3117, 3120, 3121,
<code>\Fld@menulength</code>	2727, 2733, 2780		3130, 3133, 3169, 3173, 3177, 3181,
<code>\Fld@name</code>		3185, 3189, 3193, 3197, 3201, 3205,
	2686, 2721, 2857, 2980, 3145, 3218		3260, 3310, 3373, 3428, 3429, 3466
<code>\Fld@onblur@code</code>	3187, 3189	<code>\Hy@finaltrue</code>	1522
<code>\Fld@onclick@code</code>	3310	<code>\Hy@FormObjects</code>	2612, 3037, 3099
<code>\Fld@onenter@code</code>	3199, 3201	<code>\Hy@gttemp</code>	3162, 3164
<code>\Fld@onexit@code</code>	3203, 3205	<code>\Hy@href</code>	267
<code>\Fld@onfocus@code</code>	3183, 3185	<code>\Hy@href@nextactionraw</code>	231
<code>\Fld@onmousedown@code</code>	3191, 3193	<code>\Hy@href@page</code>	219
<code>\Fld@onmouseup@code</code>	3195, 3197	<code>\Hy@linkfileprefix</code>	1531
		<code>\Hy@linktoc</code>	1542

\Hy@MakeCurrentHref	192	\HyField@PDFChoices	2844
\Hy@MakeCurrentHrefAuto	3582, 3598, 3609, 3639, 3653	\HyField@SetKeys	2698, 2739, 2742, 2761, 2859, 2902, 2950, 2985
\Hy@next	3608, 3617, 3620	\HyPat@ObjRef	3510, 3517, 3524, 3531
\Hy@numberline	158, 3538	\hyper@link	275
\Hy@org@chapter	3606, 3621	\hyper@anchor	666
\Hy@OutlineName	3554	\hyper@anchorend	666, 3573, 3600, 3611, 3641
\Hy@OutlineRerunCheck	3552	\hyper@anchorstart	666, 3573, 3600, 3611, 3641
\Hy@pdfmajorversion	160	\hyper@link	35, 715
\Hy@pdfminorversion	159	\hyper@linkend	35, 758
\Hy@pdfversion	3157	\hyper@linkfile	317, 822
\Hy@pstringdef	597, 3137	\hyper@linklaunch	39, 334, 888, 947
\Hy@PutCatalog	604	\hyper@linknamed	40, 168, 953
\Hy@raisedlink	3572, 3599, 3610, 3640	\hyper@linkstart	35, 739
\Hy@RestoreLastskip	663	\hyper@linkurl	300, 361, 778
\Hy@ReturnAfterFi	12, 3138	\hyper@normalise	267, 296, 312, 329, 355, 368
\Hy@safe@activetrue	795, 2574	\HyPL@Labels	606, 616
\Hy@SaveLastskip	643	\HyPL@storePageLabel	604
\Hy@secnum@part	3627, 3629, 3632	\HyPsd@SanitizeForOutFile	3541
\Hy@SectionAnchorHref	3563, 3587, 3658	\if@filesw	2554, 2577, 2596
\Hy@SectionHShift	3562, 3571, 3583, 3654	\ifFld@combo	2748, 2778
\Hy@StepCount	2733, 2799	\ifFld@hidden	2703, 2763
\Hy@temp	2575, 2590, 3119, 3120	\ifFld@multiline	2691
\Hy@unicodedefalse	3159	\ifFld@popdown	2749
\Hy@VerboseAnchor	644	\ifFld@radio	2745, 2768
\Hy@VerboseLinkStart	722, 746	\ifHy@implicit	3558
\Hy@VerboseLinkStop	730, 766, 805, 873, 933, 977	\ifHy@pdfescapeform	3105, 3115
\Hy@VersionChecked	151	\ifHy@unicode	3158
\Hy@WrapperDef	640	\kv@set@family@handler	171
\Hy@xspace@end	727, 804, 872, 932, 976	\m@ne	3614, 3629
\HyAnn@AbsPageLabel	2659, 2665, 2707, 2802, 2846, 2874, 2913, 2960, 2998	\OBJ@OCG@view	157
\HyAnn@Count	2663, 2664, 2667, 2668, 2669, 2674, 2678	\p@	2737
\HyField@@AddToFields	2567, 2594	\pdf@ifdraftmode	2610
\HyField@AddToFields	2565, 2715, 2835, 2852, 2883, 2939, 2968, 3032	\PDF@SetupDoc	152
\HyField@afields	2534	\PDFForm@@Name	3145, 3148, 3152, 3155
\HyField@AfterAuxOpen	2536, 2562, 2595	\PDFForm@Check	2999, 3031, 3229
\HyField@AuxAddToCoFields	2547, 2559, 2580	\PDFForm@List	2847, 2850, 3315
\HyField@AuxAddToFields	2540, 2556, 2598	\PDFForm@Name	2701, 2762, 2863, 2989, 3144
\HyField@cofields	2535	\PDFForm@Push	2875, 2881, 3281
\HyField@FlagsCheckBox	2990	\PDFForm@Radio	2803, 2823, 3350
\HyField@FlagsChoice	2773	\PDFForm@Reset	2961, 2967, 3473
\HyField@FlagsPushButton	2866, 2906, 2957	\PDFForm@Submit	2914, 2933, 3438
\HyField@FlagsRadioButton	2769	\PDFForm@Text	2708, 2713, 3388
\HyField@FlagsSubmit	2907	\protected@edef	692
\HyField@FlagsText	2702	\strip@pt	3264, 3344, 3375, 3425, 3481
		\toks@	3585, 3591, 3656, 3662
		\tw@	3586, 3591, 3657, 3662
		\url@	368, 371
		\Url@def	369
		\Url@HyperHook	369

<code>\XR@ext</code>	1528	<code>\tl_set:Nn</code>	218,
<code>\z@</code>	368, 2664, 2728, 3324, 3330,		219, 231, 262, 265, 291, 294, 350,
	3361, 3447, 3452, 3482, 3487, 3627		353, 380, 506, 507, 508, 509, 510,
text commands:			511, 672, 681, 724, 748, 830, 1965,
<code>\text_expand:n</code> ...	66, 830, 2219, 2262		1969, 1973, 1996, 2131, 2134, 2141,
<code>\text_purify:n</code>	9		2144, 2145, 2146, 2147, 2148, 2149,
<code>\textsf</code>	2921, 2925		2154, 2158, 2194, 2219, 2241, 2262
<code>\the</code>	616, 2668, 2669, 2674,	<code>\tl_set_eq:NN</code>	699
	2678, 2776, 2780, 2783, 3591, 3662	<code>\tl_to_str:N</code>	832, 838, 843
<code>\theH...</code>	13	<code>\tl_to_str:n</code>	24, 25, 34, 38
<code>\ThisShouldNotHappen</code>	3124	token commands:	
<code>\title</code>	2	<code>\token_to_str:N</code>	33
tl commands:		<code>\toks</code>	3586, 3591, 3657, 3662
<code>\c_space_tl</code>	158, 1241,		
	1272, 1277, 1282, 2828, 2829, 2836,	U	
	3048, 3050, 3060, 3062, 3064, 3067,	unicode (hypersetup key)	1498
	3069, 3264, 3267, 3397, 3425, 3426	<code>\Url</code>	4, 5, 202
<code>\tl_const:Nn</code>	476	url (hypersetup key)	10, 1557
<code>\tl_gput_right:Nn</code>	616	<code>\url</code>	4, 5, 7, 23, 342, 368
<code>\tl_gset:Nn</code>	1980, 2000, 2004	urlborder (hypersetup key)	14
<code>\tl_if_blank:nTF</code>		urlborderstyle (hypersetup key)	14, 1186
	423, 849, 2264, 2301, 2316, 2426, 2439	urlcolor (hypersetup key)	1072
<code>\tl_if_empty:nTF</code>		<code>\urldef</code>	5, 23, 368, 370
	693, 1569, 2221, 3252, 3256, 3265, 3426	<code>\Urlfont</code>	7
<code>\tl_if_empty:nTF</code>		<code>\UrlSpecial</code>	4
	225, 1106, 1146, 1165,	use commands:	
	1192, 1211, 1755, 1811, 1918, 2203	<code>\use:n</code>	942
<code>\tl_if_empty_p:N</code>	1982, 2007	<code>\use_i:nn</code>	629, 635
<code>\tl_if_eq:NnTF</code>	2129		
<code>\tl_if_exist:nTF</code> ...	3304, 3413, 3417	V	
<code>\tl_if_exist_p:N</code> ...	3295, 3402, 3404	verbose (hypersetup key)	1508
<code>\tl_if_head_eq_charcode:nNTF</code> ...			
	995, 1012	W	
<code>\tl_new:N</code>		<code>\write</code>	2555, 2558, 2578, 2597
	198, 199, 200, 201, 465, 466, 470,		
	471, 472, 473, 496, 497, 498, 499,	X	
	500, 501, 502, 503, 504, 512, 555, 666	<code>\x</code>	2740, 2760, 3590, 3592, 3661, 3663