

The `hyperref-generic` module

A generic driver for `hyperref`

The L^AT_EX Project*

Version 0.96x, released 2025-11-12

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

2 Bookmarks / outlines

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

3 “Metadata”

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

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

The current handling of the metadata is problematic:

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

To resolve some of this problem the driver will

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

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

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

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

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

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

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

- `pdflang` is deprecated. Instead 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`.

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

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

```
test-ascii
```

test grüßpdf.pdf
 test grüße.txt

6.2 Splits

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

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

For this reason three new commands have been create:

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

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

6.3 Options

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

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

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

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

The default is `phelype`.

7.3 Public interfaces

```
\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}[2025-11-12 v0.96x %
4   generic Hyperref driver for the LaTeX PDF management bundle]
5 </headertestphase>
6 <*header>
7 \ProvidesFile{hgeneric.def}[2025-11-12 v0.96x %
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:nn {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 }
434 \legacy_if:nTF { Hy@bookmarks }
435 {
436   \AddToHook{begindocument/before}[hyperref/bookmark]
437   {
438     \RequirePackage{bookmark}
439   }
440 }

```

empty hook chunk to ensure that the chunk exists.

```

441 {
442   \AddToHook{begindocument/before}[hyperref/bookmark]{}
443 }
444 \legacy_if:nT {Hy@draft}
445 {
446   \PassOptionsToPackage{draft}{bookmark}
447 }

```

6 Reference and label commands

This uses the in-built property module.

`__hyp_property_record:nn`

```

448 %
449 \cs_new_protected:Npn \__hyp_property_record:nn #1 #2 %label/attributes
450 {
451   \@bsphack
452   \property_record:nn{#1}{#2}
453   \@esphack
454 }

```

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.

```

455 \cs_generate_variant:Nn \__hyp_property_record:nn {ee}

```

(End of definition for __hyp_property_record:nn.)

7 Variables

7.1 Private temporary variables

At first a few generic tmp variables

```
\l__hyp_tmpa_tl
\l__hyp_tmpb_tl 456 \box_new:N \l__hyp_tmpa_box
\l__hyp_tmpa_seq 457 \tl_new:N \l__hyp_tmpa_tl
\l__hyp_tmpa_int 458 \tl_new:N \l__hyp_tmpb_tl
\l__hyp_tmpa_box 459 \seq_new:N \l__hyp_tmpa_seq
\l__hyp_tmpa_str 460 \int_new:N \l__hyp_tmpa_int
461 \str_new:N \l__hyp_tmpa_str
```

(End of definition for \l__hyp_tmpa_tl and others.)

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

```
\l__hyp_dest_name_tmpa_tl 462 \tl_new:N \l__hyp_dest_name_tmpa_tl
\l__hyp_uri_tmpa_tl 463 \tl_new:N \l__hyp_uri_tmpa_tl
\l__hyp_filename_tmpa_tl 464 \tl_new:N \l__hyp_filename_tmpa_tl
\l__hyp_text_tmpa_str 465 \tl_new:N \l__hyp_text_tmpa_str
\g__hyp_text_tmpa_str 466 \str_new:N \g__hyp_text_tmpa_str
467 \str_new:N \g__hyp_text_tmpa_str
```

TODO: document and check use!

(End of definition for \l__hyp_dest_name_tmpa_tl and others.)

7.2 Constants

\c__hyp_dest_undefined_tl This variable is used if a destination name is empty.

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

(End of definition for \c__hyp_dest_undefined_tl.)

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

```
\c__hyp_map_annot_hyp_prop 469 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
\c__hyp_map_hyp_annot_prop 470 {url,link,file,menu,run}
471 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
472 {
473   URI = url,
474   GoTo = link,
475   GoToR = file,
476   Named = menu,
477   Launch = run
478 }
479 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
480 {
481   url = URI,
482   link = GoTo,
483   file = GoToR,
484   menu = Named,
485   run = Launch
486 }
487
```

(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!

```
488 \tl_new:N \g__hyp_dest_pdfstartpage_tl
489 \tl_new:N \g__hyp_dest_pdfstartview_tl
490 \tl_new:N \l__hyp_dest_pdfremotestartview_tl
```

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

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

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

491 \tl_new:N \l__hyp_text_enc_uri_print_tl
492 \tl_new:N \l__hyp_text_enc_info_print_tl
493 \tl_new:N \l__hyp_text_enc_dest_tl
494 \tl_new:N \l__hyp_text_enc_dest_print_tl
495 \tl_new:N \l__hyp_text_enc_file_print_tl
496 \tl_new:N \l__hyp_text_enc_para_print_tl
497
498 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
499 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
500 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
501 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
502 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
503 \tl_set:Nn \l__hyp_text_enc_para_print_tl {utf8/string}
```

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

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

```
504 \tl_new:N\l__hyp_link_Contents_tl
505 \socket_new:nn {hyp/link/GoTo/Contents}{2}
506 \socket_new:nn {hyp/link/URI/Contents}{1}
507 \socket_new:nn {hyp/link/GoToR/Contents}{1}
508 \socket_new_plug:nnn {hyp/link/GoTo/Contents}{default}
509 {
510   \__hyp_text_pdfstring:eoN
511   { Go~to~destination~#1 }
512   { \l__hyp_text_enc_info_print_tl }
513   \l__hyp_link_Contents_tl
514   \pdfannot_dict_put:mne {link/GoTo}{Contents}
515   {\l__hyp_link_Contents_tl}
516 }
517 \socket_new_plug:nnn {hyp/link/GoToR/Contents}{default}
518 {
519   \__hyp_text_pdfstring:eoN
520   { Open~file~#1 }
```

```

521     { \l__hyp_text_enc_info_print_tl }
522     \l__hyp_link_Contents_tl
523     \pdfannot_dict_put:nne {link/GoToR}{Contents}
524     {\l__hyp_link_Contents_tl}
525   }
526 \socket_new_plug:nnn {hyp/link/URI/Contents}{default}
527 {
528   \__hyp_text_pdfstring:eON
529   { #1 }
530   { \l__hyp_text_enc_info_print_tl }
531   \l__hyp_link_Contents_tl
532   \pdfannot_dict_put:nne {link/URI}{Contents}
533   {\l__hyp_link_Contents_tl}
534 }
535 \socket_assign_plug:nn{hyp/link/GoTo/Contents}{default}
536 \socket_assign_plug:nn{hyp/link/GoToR/Contents}{default}
537 \socket_assign_plug:nn{hyp/link/URI/Contents}{default}

```

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

```

538 \tl_new:N \l__hyp_dest_pdfview_tl

```

(End of definition for `\l__hyp_dest_pdfview_tl`.)

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

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

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

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

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

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

```

539 \str_new:N \g__hyp_bordercolormodel_str

```

(End of definition for `\g__hyp_bordercolormodel_str`.)

7.4 Booleans

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

`\l_hyp_annot_colorurl_bool`

`\l_hyp_annot_colorfile_bool`

`\l_hyp_annot_colorryn_bool`

`\l_hyp_annot_colormenu_bool`

```

540 \seq_map_inline:Nn \c__hyp_annot_types_seq
541 {
542   \bool_new:c {l_hyp_annot_color#1_bool}
543 }

```

(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_ocgcolorryn_bool`

`\l_hyp_annot_ocgcolormenu_bool`

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

```

544 \seq_map_inline:Nn \c__hyp_annot_types_seq
545 {
546   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
547 }

```

(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` These booleans are used to disable some link types while keeping others.

```
548 \seq_map_inline:Nn \c_pdfannot_link_types_seq
549 {
550   \bool_new:c {l__hyp_annot_#1_bool}
551   \bool_set_true:c {l__hyp_annot_#1_bool}
552 }
```

(End of definition for \l__hyp_annot_GoTo_bool \l__hyp_annot_URI_bool \l__hyp_annot_GoToR_boo1 \l__hyp_annot_Named_bool \l__hyp_annot_Launch_bool.)

7.5 Boxes

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

```
553 \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.

```
554 \regex_const:Nn \c__hyp_dest_startview_regex
555 {
556   \A\ *
557   (?
558     (? :XYZ (? :\ +(?: (? :\d+|\d*\.\d+) |null)) {3} \ )
559     |
560     (? :Fit\b|FitB\b)
561     |
562     (? :(? :FitH|FitV|FitBH|FitBV) (? :\ +(?: \d+|\d*\.\d+) |\ +null) {1} )
563     |
564     (? :FitR (? :\ +\d+|\ +\d*\.\d+) {4} \ )
565   )
566 }
```

(End of definition for \c__hyp_dest_startview_regex.)

7.7 PDF dictionaries

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

```
567 \pdfdict_new:n {l__hyp_page/Trans}
568 \pdfdict_put:nnn {l__hyp_page/Trans}{Type}{/Trans}
```

(End of definition for l__hyp_page/Trans.)

8 PDF string conversion

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

hyp/text/pdfstring

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

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

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

__hyp_text_purify:nN

```
570 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
571 {
572   \str_set:Ne #2 {\text_purify:n { #1 } }
573 }
```

(End of definition for __hyp_text_purify:nN.)

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

__hyp_text_cleanup:N

```
574 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
575 {
576
577 }
```

(End of definition for __hyp_text_cleanup:N.)

The last step converts the string to a PDF encoding. As we have at least two targets (hex and literal) there is an argument. The conversion assumes utf8 input, it is based on `cspdf_string_from_unicode:nnN` in `l3pdftools`.

#2 is str variable, #1 should be one of

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

__hyp_text_string_from_unicode:nN

```
578 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
579 {
580   \pdf_string_from_unicode:nVN { #1 } #2 #2
581 }
```

(End of definition for __hyp_text_string_from_unicode:nN.)

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

__hyp_text_pdfstring:nnN

```
582 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
583 {
584   \group_begin:
585   \Hy@pdfstringtrue
```

```

586 \hook_use:n {hyp/text/pdfstring}
587 \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
588 \__hyp_text_cleanup:N \l__hyp_text_tmpa_str
589 \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
590 \str_gset_eq:NN \g__hyp_text_tmpa_str\l__hyp_text_tmpa_str
591 \group_end:
592 \str_set_eq:NN #3 \g__hyp_text_tmpa_str
593 }
594 \cs_generate_variant:Nn \__hyp_text_pdfstring:nnN {enN,onN,eoN,ooN,noN}

```

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

!!! temporary until all instances are gone

```

595 \cs_new_protected:Npn\Hy@pstringdef #1 #2
596 { \__hyp_text_pdfstring:enN {#2} {utf8/string-raw}#1 }

```

This is a special version for info keys:

`__hyp_text_pdfstring_info:nN`

```

597 \cs_new_protected:Npn \__hyp_text_pdfstring_info:nN #1 #2
598 {
599 \__hyp_text_pdfstring:noN { #1 } { \l__hyp_text_enc_info_print_tl } #2
600 }
601 \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
602 \cs_new_protected:Npn\__hyp_PageLabels_gpush:
603   {
604     \pdfmanagement_add:nne {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
605   }
606
607 \def\Hy@PutCatalog #1 {}
608
609
610 \legacy_if:nT { Hy@pdfpagelabels }
611   {
612     \cs_set_protected:Npn \HyPL@StorePageLabel #1
613       {
614         \tl_gput_right:Ne \HyPL@Labels { \the\Hy@abspage<<#1>> }
615         \__hyp_PageLabels_gpush:
616       }
617   }

```

(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
618 \int_new:N \g__hyp_linknestlevel_int

```


(End of definition for `g__hyp_linknestlevel_int`.)

```

619 \prg_new_conditional:Npnn \__hyp_if_outer_link: {TF}
620 {
621   \int_compare:nNnTF { \g__hyp_linknestlevel_int } > {1}
622     { \prg_return_false: }
623     { \prg_return_true: }
624 }
625 \cs_new:Npn \__hyp_check_link_nesting:TF #1 #2
626 {
627   \use_i:nn {#1}{#2}
628 }
629 \keys_define:nn { hyp }
630 {
631   nested-links .choice:,
632   nested-links / true .code:n =
633     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \use_i:nn },
634   nested-links / false .code:n =
635     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \__hyp_if_outer_link:TF },
636   nested-links .default:n = {true}
637 }

```

10.2 Anchors / destinations

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

```

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

```

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

```

\__hyp_destination:nn

```

```

638 \Hy@WrapperDef \__hyp_destination:nn #1 #2
639 {
640   \mode_if_horizontal:T { \@savsf\spacefactor }
641   \Hy@SaveLastskip      %defined in hyperref
642   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
643   \__hyp_text_pdfstring:eoN
644   { \HyperDestNameFilter{#1} }
645   { \l__hyp_text_enc_dest_tl }
646   \l__hyp_tmpa_tl
647   \str_if_eq:nnTF {#2} {fitrbox}
648   {
649     \exp_args:NV
650     \pdf_destination:nmmn \l__hyp_tmpa_tl
651     { \box_wd:N \l__hyp_dest_box }
652     { \box_ht:N \l__hyp_dest_box }
653     { \box_dp:N \l__hyp_dest_box }
654   }
655   {

```

```

656     \exp_args:NV
657     \pdf_destination:nf
658     { \l__hyp_tmpa_tl }
659     { #2 }
660   }
661   \Hy@RestoreLastskip   %defined in hyperref
662   \mode_if_horizontal:T { \spacefactor\@savsf }
663 }

```

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

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

```

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

```

(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:

```

687 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
688 {
689   \mode_leave_vertical:

```

```

690 \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
691 \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
692 {
693   \msg_warning:nne
694     { hyp }
695     { empty-destination-name }
696     { \c__hyp_dest_undefined_tl }
697   \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
698 }
699 {
700   \__hyp_text_pdfstring:eoN
701   { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
702   { \l__hyp_text_enc_dest_tl }
703   \l__hyp_dest_name_tmpa_tl
704 }
705 \exp_args:No
706   \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
707 }
708
709 \cs_new_protected:Npn \__hyp_link_goto_end:
710 {
711   \pdfannot_link_goto_end:
712 }

```

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

```

713
714 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
715 {
716   \bool_if:NTF \l__hyp_annot_GoTo_bool
717   {

```

```

718     \int_gincr:N\g__hyp_linknestlevel_int
719     \__hyp_check_link_nesting:TF
720     {
721         \Hy@VerboseLinkStart{#1}{#2}
722         \group_begin:
723         \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
this socket adds something to the /Contents key.
724         \socket_use:nnn{hyp/link/GoTo/Contents}{#2}{#3}
725         \hook_use:n {hyp/link/#1}
726         \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
727         \__hyp_link_goto_end:
728         \group_end:
729         \Hy@VerboseLinkStop
730     }
731     {
732         \group_begin: #3\group_end:
733     }
734     \int_gdecr:N\g__hyp_linknestlevel_int
735 }
736 {\let\protect\relax#3}}
737 }
738 \cs_new_protected:Npn \hyper@linkstart #1 #2 % #1 context, #2=destination name
739 {
740     \bool_if:NT \l__hyp_annot_GoTo_bool
741     {
742         \int_gincr:N\g__hyp_linknestlevel_int
743         \__hyp_check_link_nesting:TF
744         {
745             \Hy@VerboseLinkStart{#1}{#2}% only for debug
746             \group_begin:
747             \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
748             \socket_use:nnn{hyp/link/GoTo/Contents}{#2}{#3}
749             \hook_use:n {hyp/link/#1}
750             \__hyp_link_goto_begin:nw {#2}
751         }
752         {
753             \group_begin:
754         }
755     }
756 }
757
758 \cs_new_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:eoN
787             { #2}
788             { \l__hyp_text_enc_uri_print_tl }
789             \l__hyp_uri_tmpa_tl
790             \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
791             \bool_if:NT \l__hyp_href_url_ismap_bool
792             {
793                 \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
794             }

```

This socket adds something to the /Contents key.

```

795         \socket_use:nn{hyp/link/URI/Contents}{#2}
796         \cs_set_eq:NN \# \c_hash_str
797         \cs_set_eq:NN \% \c_percent_str
798         \Hy@safe@activetrue
799         \mode_leave_vertical:
800         \pdfannot_dict_put:nne {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
801         \pdfannot_link:nen { URI }
802         {
803         }
804         {
805             \let\protect\relax
806             #1
807             \Hy@xspace@end
808             \Hy@VerboseLinkStop %where is the start??
809         }
810         \group_end:
811     }
812     {
813         \group_begin: #1 \group_end:
814     }
815     \int_gdecr:N\g__hyp_linknestlevel_int
816 }
817 {\let\protect\relax#1}}
818 }

```

10.5 GoToR Links files

```

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

```

This socket adds something to the /Contents key.

```

851   \socket_use:nn{hyp/link/GoToR/Contents}{#2}
852   \tl_if_blank:eTF {#3}
853   {
854     \pdfdict_put:nne {l_hyp/annot/A/GoToR}{D}
855     {
856       [
857         \int_eval:n
858         { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
859         /\l__hyp_dest_pdfremotestartview_tl
860       ]
861     }
862   }
863   {
864     \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
865   }
866   \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>

```

867     \pdf_object_unnamed_write:ne{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
868     \pdfannot_dict_put:nne {link/GoToR}{A}{\pdf_object_ref_last:}
869     \pdfannot_link:nnn %expansion??
870     { GoToR }
871     {
872     }
873     {
874     \let\protect\relax
875     #1\Hy@xspace@end
876     \Hy@VerboseLinkStop %where is the start??
877     }
878     \group_end:
879     }
880     {
881     \group_begin: #1 \group_end:
882     }
883     \int_gdecr:N\g__hyp_linknestlevel_int
884     }
885     {{\let\protect\relax#1}}
886     }

```

10.6 Launch links

We define `\hyper@linklaunch` for naming consistency

```

887 \pdfdict_new:n {l_hyp/annot/A/Launch}
888 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
889 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
890
891 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
892 {
893   \bool_if:NTF \l__hyp_annot_Launch_bool
894   {
895     \int_gincr:N\g__hyp_linknestlevel_int
896     \__hyp_check_link_nesting:TF
897     {
898       \group_begin:
899       \__hyp_text_pdfstring:nnN
900       { #1 }
901       { \l__hyp_text_enc_file_print_tl }
902       \l__hyp_filename_tmpa_tl
903       \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
904       \__hyp_text_pdfstring:noN
905       { #3 }
906       { \l__hyp_text_enc_para_print_tl }
907       \l__hyp_para_tmpa_tl
908       \bool_if:nTF
909       {
910         \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}()
911         ||
912         \pdf_version_compare_p:Nn > {1.9}
913       }
914       {

```

```

915         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
916     }
917     {
918         \pdfdict_put:nne
919         {l_hyp/annot/A/Launch}
920         {Win}
921         {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
922     }
923     \mode_leave_vertical:
924     \pdfannot_dict_put:nne {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}
925     \pdfannot_link:nen
926     { Launch }
927     {
928         % /A
929         % <<
930         % \pdfdict_use:n {l_hyp/annot/A/Launch}
931         % >>
932     }
933     {
934         \let\protect\relax
935         #2\Hy@xspace@end
936         \Hy@VerboseLinkStop %where is the start??
937     }
938     \group_end:
939     }
940     { \group_begin: #2 \group_end: }
941     \int_gdecr:N\g__hyp_linknestlevel_int
942     }
943     {\let\protect\relax#2}}
944 }

```

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

```

945 \use:e
946 { % filename, anchor text, linkname
947   \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str #1 \exp_not:N \ \ #2 #3
948 }
949 {
950   \hyper@linklaunch {#1}{#2}{#3}
951 }

```

10.7 Named links (menu)

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

```

952 \pdfdict_new:n {l_hyp/annot/A/Named}
953 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
954 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
955
956 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
957 {
958   \bool_if:NTF \l__hyp_annot_Named_bool
959   {
960     \int_gincr:N\g__hyp_linknestlevel_int
961     \__hyp_check_link_nesting:TF

```



```

962     {
963     \group_begin:
964     \pdfmeta_standard_verify:nnTF {named_actions}{#1}
965     {
966     \mode_leave_vertical:
967     \pdfdict_put:nne {l_hyp/annot/A/Named}{N}
968     {\pdf_name_from_unicode_e:n{#1}}
969     \pdfannot_dict_put:nne {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}
970     \pdfannot_link:nnn { Named }
971     {
972     % /A
973     % <<
974     % \pdfdict_use:n { l_hyp/annot/A/Named }
975     % >>
976     }
977     {
978     #2
979     \Hy@xspace@end
980     \Hy@VerboseLinkStop
981     }
982     }
983     {
984     \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
985     #2
986     }
987     \group_end:
988     }
989     { \group_begin: #2 \group_end: }
990     \int_gdecr:N\g__hyp_linknestlevel_int
991     }
992     {\let\protect\relax#2}}
993 }
994

```

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.

```

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

```

```

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

```

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

```

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

```

```

996 \cs_new_protected:Npn \__hyp_color_select:n #1

```

```

997 {
998   \t1_if_head_eq_charcode:nNTF {#1}[ %]
999   {
1000     \__hyp_color_select_aux:wn #1
1001   }
1002   {
1003     \color_select:n {#1}
1004   }
1005 }
1006
1007 \cs_new_protected:Npn \__hyp_color_select_aux:wn [#1] #2
1008 {
1009   \color_select:nn {#1}{#2}
1010 }
1011
1012 \cs_generate_variant:Nn \__hyp_color_select:n {e}

```

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

```

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

```

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

```

\__hyp_color_set:nn \__hyp_color_set_aux:nwn

```

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

```

1013 \cs_new_protected:Npn \__hyp_color_set:nn #1 #2
1014 {
1015   \t1_if_head_eq_charcode:nNTF {#2}[ %]
1016   {
1017     \__hyp_color_set_aux:nwn { #1 } #2
1018   }
1019   {
1020     \color_set:nn {#1} {#2}
1021   }
1022 }
1023
1024 \cs_new_protected:Npn \__hyp_color_set_aux:nwn #1 [#2] #3
1025 {
1026   \color_set:nnn {#1}{#2}{#3}
1027 }
1028
1029 \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.

```

1030 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1031 {
1032   \hook_gput_code:nnn

```

```

1033     {pdfannot/link/#2/begin}
1034     {hyp/color}
1035     {
1036         \bool_if:cT { l_hyp_annot_color#1_bool }
1037         {
1038             \group_begin:
1039             \color_select:n { hyp/color/#1}
1040         }
1041     }
1042     \hook_gput_code:nnn
1043     {pdfannot/link/#2/end}
1044     {hyp/color}
1045     {
1046         \bool_if:cT { l_hyp_annot_color#1_bool }
1047         {
1048             \group_end:
1049         }
1050     }
1051 }

```

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

```

1052 \keys_define:nn { hyp }
1053 {
1054     ,colorlinks .choice:
1055     ,colorlinks / true .meta:n =
1056     {
1057         ,pdfborder={0~0~0}
1058         ,pdfborderstyle=
1059         ,colorurl =#1
1060         ,colorlink =#1
1061         ,colorryn =#1
1062         ,colormenu =#1
1063         ,colorfile =#1
1064     }
1065     ,colorlinks / false .meta:n =
1066     {
1067         ,colorurl =#1
1068         ,colorlink =#1
1069         ,colorryn =#1
1070         ,colormenu =#1
1071         ,colorfile =#1
1072     }
1073     ,colorlinks .default:n = {true}
1074 }

```

`colorurl` (*setup key*)

`colorlink` (*setup key*)

`colorryn` (*setup key*)

`colormenu` (*setup key*)

`colorfile` (*setup key*)

`urlcolor` (*setup key*)

`linkcolor` (*setup key*)

`runcolor` (*setup key*)

`menucolor` (*setup key*)

`filecolor` (*setup key*)

`allcolors` (*setup key*)

```

1075 \seq_map_inline:Nn \c__hyp_annot_types_seq
1076 {
1077     \keys_define:nn { hyp }
1078     {
1079         ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
1080         ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
1081     }
1082 }

```

```

1083
1084 \keys_define:nn { hyp }
1085 {
1086   ,allcolors .meta:n =
1087   {
1088     ,urlcolor=#1
1089     ,linkcolor=#1
1090     ,runcolor=#1
1091     ,filecolor=#1
1092     ,menucolor=#1
1093   }
1094   ,allcolors .value_required:n = true
1095 }

```

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*)

```

1096 \keys_define:nn { hyp }
1097 {
1098   ,bordercolormodel .choices:nn =
1099   {rgb,cmyk}
1100   { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
1101   ,bordercolormodel .initial:n ={rgb}
1102 }

1103 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1104 {
1105   \keys_define:nn { hyp }
1106   {
1107     #1bordercolor .code:n =
1108     {
1109       \tl_if_empty:nTF { ##1 }
1110       {
1111         \pdfannot_dict_remove:nn
1112         {link/#2}
1113         { C }
1114       }
1115       {
1116         \__hyp_color_set:ne {hyp/color/#1border}{##1}
1117         \color_export:nVN
1118         {hyp/color/#1border}
1119         \g__hyp_bordercolormodel_str
1120         \l__hyp_tmpa_tl
1121         \pdfannot_dict_put:nne
1122         {link/#2}
1123         { C }
1124         { [\l__hyp_tmpa_tl] }
1125       }
1126     }

```

```

1127     }
1128   }
1129
1130   \keys_define:nn { hyp }
1131   {
1132     ,allbordercolors .meta:n =
1133     {
1134       ,linkbordercolor=#1
1135       ,urlbordercolor =#1
1136       ,filebordercolor=#1
1137       ,menubordercolor=#1
1138       ,runbordercolor =#1
1139     }
1140     ,allbordercolors .value_required:n = true
1141   }
1142

```

11.3.2 Borderwidth and -arc

```

1143   \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1144   {
1145     \keys_define:nn { hyp }
1146     {
1147       #1border .code:n =
1148       {
1149         \tl_if_empty:nTF { ##1 }
1150         {
1151           \pdfannot_dict_remove:nn
1152           {link/#2}
1153           { Border }
1154         }
1155         {
1156           \pdfannot_dict_put:nnn
1157           {link/#2}
1158           { Border }
1159           { [##1] }
1160         }
1161       }
1162     }
1163   }
1164   \keys_define:nn { hyp }
1165   {
1166     ,pdfborder .code:n =
1167     {
1168       \tl_if_empty:nTF { #1 }
1169       {
1170         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1171         {
1172           \pdfannot_dict_remove:nn
1173           {link/##2}
1174           { Border }
1175         }
1176       }
1177     }

```

```

1178         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1179         {
1180             \pdfannot_dict_put:nnn
1181             {link/##2}
1182             { Border }
1183             { [#1] }
1184         }
1185     }
1186 }
1187 ,pdfborder .initial:n = {0~0~1},
1188 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1189 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1190 {
runborderstyle (setup key) 1191   \keys_define:nn { hyp }
fileborderstyle (setup key) 1192   {
menuborderstyle (setup key) 1193     #1borderstyle .code:n =
1194     {
1195         \tl_if_empty:nTF { ##1 }
1196         {
1197             \pdfannot_dict_remove:nn
1198             {link/#2}
1199             { BS }
1200         }
1201         {
1202             \pdfannot_dict_put:nnn
1203             {link/#2}
1204             { BS }
1205             { <<##1>> }
1206         }
1207     }
1208 }
1209 }
1210 \keys_define:nn { hyp }
1211 {
1212     ,pdfborderstyle .code:n =
1213     {
1214         \tl_if_empty:nTF { #1 }
1215         {
1216             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1217             {
1218                 \pdfannot_dict_remove:nn
1219                 {link/##2}
1220                 { BS }
1221             }
1222         }
1223         {
1224             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1225             {
1226                 \pdfannot_dict_put:nnn

```

```

1227         {link/##2}
1228         { BS }
1229         { <<#1>> }
1230     }
1231 }
1232 }
1233 ,pdfborderstyle .initial:n = {},
1234 }

```

11.4 ocolorlinks

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

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

```

1235 \cs_new_protected:Npn \_hyp_ocg_init:
1236 {
1237     \pdf_object_new:n { __hyp/OCG/View }
1238     \pdf_object_new:n { __hyp/OCG/Print }
1239     \pdf_object_new:n { __hyp/OCG/config }
1240     \pdf_object_new:n { __hyp/OCG/refarray }
1241     \pdf_object_write:mne { __hyp/OCG/refarray } { array }
1242     {
1243         \pdf_object_ref:n { __hyp/OCG/View }
1244         \c_space_tl
1245         \pdf_object_ref:n { __hyp/OCG/Print }
1246     }
1247     \pdf_object_write:nnn { __hyp/OCG/View } { dict }
1248     {
1249         /Type/OCG
1250         /Name(View)
1251         /Usage
1252         <<
1253         /Print <</PrintState/OFF>>~
1254         /View <</ViewState/ON >>~
1255         >>
1256     }
1257     \pdf_object_write:nnn { __hyp/OCG/Print } { dict }
1258     {
1259         /Type/OCG
1260         /Name(Print)
1261         /Usage
1262         <<
1263         /Print <</PrintState/ON>>~
1264         /View <</ViewState/OFF>>~
1265         >>
1266     }
1267     \pdfmanagement_add:mne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/View} }
1268     \pdfmanagement_add:mne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/Print} }
1269     \pdf_object_write:mne { __hyp/OCG/config } { dict }
1270     {
1271         /OFF[\pdf_object_ref:n { __hyp/OCG/Print } ]
1272         /AS[

```

```

1273         <<
1274         /Event/View
1275         /OCGs\c_space_t1 \pdf_object_ref:n { __hyp/OCG/refarray }
1276         /Category[/View]
1277         >>
1278         <<
1279         /Event/Print
1280         /OCGs\c_space_t1 \pdf_object_ref:n { __hyp/OCG/refarray }
1281         /Category[/Print]
1282         >>
1283         <<
1284         /Event/Export
1285         /OCGs\c_space_t1 \pdf_object_ref:n { __hyp/OCG/refarray }
1286         /Category[/Print]
1287         >>
1288         ]
1289     }
1290     \pdfmanagement_add:nne { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG
1291     \cs_gset:Npn \__hyp_ocg_init: {}
1292 }

```

(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.

```

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

```



```

1323         \pdf_emc:
1324         }
1325     \group_end:
1326 }
1327 }
1328 }

```

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

```

ocgcolorlink (setup key) 1329 \bool_lazy_or:nnTF
ocgcolorurl (setup key) 1330 { \pdf_version_compare_p:Nn > {1.4} }
ocgcolorfile (setup key) 1331 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
ocgcolormenu (setup key) 1332 {
ocgcolorrund (setup key) 1333     \keys_define:nn { hyp }
1334     {
1335         ,_ocgcolorlinks .meta:n =
1336         {
1337             ocgcolorlink=#1,
1338             ocgcolorurl=#1,
1339             ocgcolorfile=#1,
1340             ocgcolorrund=#1,
1341             ocgcolormenu=#1
1342         }
1343         ,_ocgcolorlinks .default:n = true
1344     }
1345 }
1346 {
1347     \keys_define:nn { hyp }
1348     {
1349         ,_ocgcolorlinks .code:n =
1350         {
1351             \msg_warning:nnee
1352             { hyp }
1353             { ignore-deprecated-or-unknown-option-in-pdf-version }
1354             { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1355         }
1356     }
1357 }
1358
1359 \keys_define:nn { hyp }
1360 {
1361     ,ocgcolorlinks .choice:
1362     ,ocgcolorlinks / true .meta:n =
1363     {
1364         pdfborder      ={{0~0~0}},
1365         pdfborderstyle ={},
1366         colorlinks     = false,
1367         _ocgcolorlinks = true
1368     }
1369     ,ocgcolorlinks / false .meta:n =
1370     {
1371         _ocgcolorlinks = false
1372     }
1373     ,ocgcolorlinks .default:n = {true}
1374 }

```

```

1375
1376 \seq_map_inline:Nn \c__hyp_annot_types_seq
1377 {
1378   \bool_lazy_or:nnTF
1379   { \pdf_version_compare_p:Nn > {1.4} }
1380   { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1381   {
1382     \keys_define:nn { hyp }
1383     {
1384       ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1385     }
1386   }
1387   {
1388     \keys_define:nn { hyp }
1389     {
1390       ,ocgcolor#1 .code:n=
1391       {
1392         \msg_warning:nnee
1393         { hyp }
1394         { ignore-deprecated-or-unknown-option-in-pdf-version }
1395         { ocgcolor#1 }
1396         { \pdf_version_major:.\pdf_version_minor: }
1397       }
1398     }
1399   }
1400 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1401 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1402 {
1403   \keys_define:nn { hyp }
1404   {
1405     ,#1highlight .choices:nn =
1406     { /I, /N, /O, /P}
1407     {
1408       \pdfannot_dict_put:nnn
1409       {link/#2}
1410       { H }
1411       { ##1 }
1412     }
1413     ,#1highlight / .code:n =
1414     {
1415       \pdfannot_dict_remove:nn
1416       {link/#2}
1417       { H }
1418     }
1419     }
1420     ,#1highlight / unknown .code:n =
1421     {
1422       \msg_warning:nneee { hyp } { unknown-choice+empty }
1423       { #1highlight }
1424     }

```

```

1425         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1426         { \exp_not:n {##1} }
1427     }
1428 }
1429 }
1430
1431
1432 \keys_define:nn { hyp }
1433 {
1434     ,pdfhighlight .choices:nn =
1435     { /I, /N, /O, /P}
1436     {
1437         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1438         {
1439             \pdfannot_dict_put:nnn
1440             {link/###2}
1441             { H }
1442             { #1 }
1443         }
1444     }
1445     ,pdfhighlight / .code:n =
1446     {
1447         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1448         {
1449             \pdfannot_dict_remove:nn
1450             {link/##2}
1451             { H }
1452         }
1453     }
1454     ,pdfhighlight .initial:n = {/I},
1455     ,pdfhighlight / unknown .code:n =
1456     {
1457         \msg_warning:neee { hyp } { unknown-choice+empty }
1458         { pdfhighlight }
1459         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1460         { \exp_not:n {#1} }
1461     }
1462 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1463 \keys_define:nn { hyp }
hideurl (setup key) 1464 {
hidefile (setup key) 1465     hidelinks .meta:n =
hiderun (setup key) 1466     {
hidemenu (setup key) 1467         ,colorlinks = false
1468         ,ocgcolorlinks = false
1469         ,pdfborder = { 0~0~0 }
1470         ,pdfborderstyle=
1471     }
1472 }

```

```

1473
1474 \seq_map_inline:Nn \c__hyp_annot_types_seq
1475 {
1476   \keys_define:nn { hyp }
1477   {
1478     hide#1 .meta:n =
1479     {
1480       ,color#1      = false
1481       ,ocgcolor#1   = false
1482       ,#1border     = { 0~0~0 }
1483       ,#1borderstyle =
1484     }
1485   }
1486 }

```

11.7 color schemes and settings

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

`colorscheme` (*setup key*)

```

1487 \keys_define:nn { hyp }
1488 {
1489   colorscheme .code:n =
1490   {
1491     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1492     {
1493       \keys_set:nn { hyp }
1494       {
1495         ##1 = ##2
1496       }
1497     }
1498   }
1499 }
1500 \keys_set:nn { hyp } {colorscheme=phelype}

```

12 Keys

12.1 Ignored keys

The following are ignored (with or without warnings)

`unicode` (*setup key*)

`pdfencoding` (*setup key*)

`pdfversion` (*setup key*)

```

1501 \keys_define:nn { hyp }
1502 {
1503   ,unicode      .code:n = {}
1504   ,pdfencoding  .code:n = {}
1505   ,pdfversion   .code:n =
1506   {
1507     \msg_warning:nn { hyp }{ pdfversion-disabled }
1508   }
1509 }
1510 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
debug (setup key) 1511 \keys_define:nn { hyp }
draft (setup key) 1512 {
final (setup key) 1513   ,verbose .legacy_if_set:n = {Hy@verbose}
1514   ,debug .legacy_if_set:n = {Hy@verbose}
1515 }
1516 \keys_define:nn { hyp }
1517 {
1518   ,draft .code:n =
1519   {
1520     \Hy@drafttrue
1521     \PassOptionsToPackage{draft}{bookmark}
1522   }
1523   ,final .code:n =
1524   {
1525     \Hy@finaltrue
1526     \PassOptionsToPackage{final}{bookmark}
1527   }
1528 }

extension (setup key)
hypertextnames (setup key) 1529 \keys_define:nn { hyp }
naturalnames (setup key) 1530 {
pageanchor (setup key) 1531   ,extension .tl_set:N = \XR@ext
linktoc (setup key) 1532   ,extension .initial:n= pdf
linktocpage (setup key) 1533   ,hypertextnames .legacy_if_set:n = {Hy@hypertextnames}
plainpages (setup key) 1534   ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
localanchorname (setup key) 1535   ,localanchorname .legacy_if_set:n = {Hy@localanchorname}
linkfileprefix (setup key) 1536   ,naturalnames .legacy_if_set:n = {Hy@naturalnames}
1537   ,pageanchor .legacy_if_set:n = {Hy@pageanchor}
1538   ,plainpages .legacy_if_set:n = {Hy@plainpages}
1539 }
1540
1541 \keys_define:nn { hyp }
1542 {
1543   ,linktoc .choices:nn = { none, section, all, page }
1544   {
1545     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1546   }
1547   ,linktoc / unknown .code:n =
1548   {
1549     \msg_warning:nnee { hyp } { unknown-choice }
1550     { linktoc }
1551     { none, section, all, page }
1552     { \exp_not:n {#1} }
1553   }
1554   ,linktocpage .choice:
1555   ,linktocpage / true .meta:n = {linktoc=page}
1556   ,linktocpage / false .meta:n = {linktoc=section}
1557   ,linktocpage .default:n = true

```

```

1558 }
1559
link (setup key) This booleans allow to disable the link types.
url (setup key) 1560 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1561 {
menu (setup key) 1562 \keys_define:nn { hyp }
run (setup key) 1563 {
1564 ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1565 }
1566 }

1567 \keys_define:nn { hyp }
1568 {
1569 ,baseurl .code:n =
1570 {
1571 \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1572 \tl_if_empty:NTF \l__hyp_tmpa_tl
1573 {
1574 \pdfmanagement_remove:nn {Catalog} { URI }
1575 }
1576 {
1577 \pdfmanagement_add:nne {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1578 }
1579 \__hyp_store_metadata:nn {baseurl}{#1}
1580 }
1581 %only false does something ...
1582 ,bookmarks .choice:
1583 ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1584 ,bookmarks / true .code:n = {}
1585 ,bookmarks .default:n = {true}
1586 ,bookmarksnumbered .legacy_if_set:n = {Hy@bookmarksnumbered}
1587 ,bookmarksopen .legacy_if_set:n = {Hy@bookmarksopen}
1588 ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1589 ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1590 ,pdfcenterwindow .choice:
1591 ,pdfcenterwindow / false .code:n =
1592 {
1593 \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1594 }
1595 ,pdfcenterwindow / true .code:n =
1596 {
1597 \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1598 }
1599 ,pdfcenterwindow / .code:n =
1600 {
1601 \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1602 }
1603 ,pdfcenterwindow / unknown .code:n =
1604 {
1605 \msg_warning:nnee { hyp } { no-bool }
1606 { pdfcenterwindow }
1607 { \exp_not:n {#1} }
1608 }

```

```

1609 ,pdfcenterwindow .default:n = true
1610 ,pdfdirection .choice:
1611 ,pdfdirection / L2R .code:n =
1612 {
1613   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{/L2R }
1614 }
1615 ,pdfdirection / R2L .code:n =
1616 {
1617   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{/R2L }
1618 }
1619 ,pdfdirection / .code:n =
1620 {
1621   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1622 }
1623 ,pdfdirection / unknown .code:n =
1624 {
1625   \msg_warning:nnee { hyp } { unknown-choice+empty }
1626   { pdfdirection }
1627   { L2R , R2L }
1628   { \exp_not:n {#1} }
1629 }
1630 ,pdfdisplaydoctitle .choice:
1631 ,pdfdisplaydoctitle / false .code:n =
1632 {
1633   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1634 }
1635 ,pdfdisplaydoctitle / true .code:n =
1636 {
1637   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1638 }
1639 ,pdfdisplaydoctitle .default:n = true
1640 ,pdfduplex .choices:nn =
1641 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1642 {
1643   \pdf_version_compare:NnTF > {1.6}
1644   {
1645     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1646     { Duplex } { /#1 }
1647   }
1648   {
1649     \msg_warning:nnee
1650     {hyp}
1651     {ignore-deprecated-or-unknown-option-in-pdf-version}
1652     {pdfduplex}
1653     {\pdf_version:}
1654   }
1655 }%
1656 ,pdfduplex / .code:n =
1657 {
1658   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Duplex }
1659 }
1660 ,pdfduplex / unknown .code:n =
1661 {
1662   \msg_warning:nnee { hyp } { unknown-choice+empty }

```

```

1663     { pdfduplex }
1664     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1665     { \exp_not:n {#1} }
1666   }
1667 ,pdffitwindow .choice:
1668 ,pdffitwindow / false .code:n =
1669   {
1670     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1671   }
1672 ,pdffitwindow / true .code:n =
1673   {
1674     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1675   }
1676 ,pdffitwindow / .code:n =
1677   {
1678     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1679   }
1680 ,pdffitwindow .default:n = true
1681 ,pdffitwindow / unknown .code:n =
1682   {
1683     \msg_warning:nnee { hyp } { no-bool }
1684     { pdffitwindow }
1685     { \exp_not:n {#1} }
1686   }
1687 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1688 ,pdflinkmargin .initial:n = {1pt}
1689 ,pdfmenubar .choice:
1690 ,pdfmenubar / true .code:n =
1691   {
1692     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1693   }
1694 ,pdfmenubar / false .code:n =
1695   {
1696     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1697     { HideMenubar } { true }
1698   }
1699 ,pdfmenubar / .code:n =
1700   {
1701     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1702   }
1703 ,pdfmenubar .default:n = true
1704 ,pdfmenubar / unknown .code:n =
1705   {
1706     \msg_warning:nnee { hyp } { no-bool }
1707     { pdfmenubar }
1708     { \exp_not:n {#1} }
1709   }
1710 ,pdfnewwindow .choice:
1711 ,pdfnewwindow / true .code:n =
1712   {
1713     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1714     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1715   }
1716 ,pdfnewwindow / false .code:n =

```



```

1717     {
1718     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1719     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1720     }
1721 ,pdfnewwindow / .code:n =
1722     {
1723     \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1724     \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1725     }
1726 ,pdfnonfullscreenpagemode .choices:nn =
1727     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1728     {
1729     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1730     { NonFullScreenPageMode } {/#1}
1731     }
1732 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1733     {
1734     \pdf_version_compare:NnTF < {1.6}
1735     {
1736     %message
1737     }
1738     {
1739     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1740     {NonFullScreenPageMode}{/UseAttachments}
1741     }
1742     }
1743 ,pdfnonfullscreenpagemode / .code:n =
1744     {
1745     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1746     }
1747 ,pdfnonfullscreenpagemode / unknown .code:n =
1748     {
1749     \msg_warning:nree { hyp } { unknown-choice+empty }
1750     { pdfnonfullscreenpagemode }
1751     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1752     { \exp_not:n {#1} }
1753     }
1754 ,pdfnumcopies .code:n =
1755     {
1756     \pdf_version_compare:NnTF > {1.6}
1757     {
1758     \tl_if_empty:nTF {#1}
1759     {
1760     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1761     }
1762     {
1763     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1764     {NumCopies}{#1}
1765     }
1766     }
1767     {
1768     \msg_warning:nnee
1769     {hyp}
1770     {ignore-deprecated-or-unknown-option-in-pdf-version}

```

```

1771         {pdfnumcopies}
1772         {\pdf_version:}
1773     }
1774 }
1775 ,pdfpagelayout .choices:nn =
1776 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1777 { \pdfmanagement_add:nne {Catalog} { PageLayout }{ /#1 } }
1778 ,pdfpagelayout / .code:n =
1779 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1780 ,pdfpagelayout / unknown .code:n =
1781 {
1782     \msg_warning:nnee { hyp } { unknown-choice+empty }
1783     { pdfpagelayout }
1784     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1785     { \exp_not:n {#1} }
1786 }
1787 ,pdfpagemode .choices:nn =
1788 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1789 { \pdfmanagement_add:nne {Catalog} { PageMode }{ /#1 } }
1790 ,pdfpagemode / UseAttachments .code:n =
1791 {
1792     \pdf_version_compare:NnTF > {1.5}
1793     {
1794         \pdfmanagement_add:nne {Catalog} { PageMode }{ /UseAttachments }
1795     }
1796     {
1797         \msg_warning:nnee
1798         {hyp}
1799         {ignore-deprecated-or-unknown-value-in-pdf-version}
1800         {UseAttachments}
1801         {\pdf_version:}
1802     }
1803 }
1804 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1805 ,pdfpagemode / unknown .code:n =
1806 {
1807     \msg_warning:nnee { hyp } { unknown-choice+empty }
1808     { pdfpagemode }
1809     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1810     { \exp_not:n {#1} }
1811 }
1812 ,pdfpagescrop .code:n =
1813 {
1814     \tl_if_empty:nTF {#1} %or blank?
1815     {
1816         \pdfmanagement_remove:nn {Pages} { CropBox }
1817     }
1818     {
1819         \pdfmanagement_add:nne {Pages} { CropBox } { [#1] }
1820     }
1821 }
1822 ,pdfpicktraybypdfsize .choice:
1823 ,pdfpicktraybypdfsize / true .code:n =
1824 {

```

```

1825     \pdf_version_compare:NnTF > {1.6}
1826     {
1827         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1828         { PickTrayByPDFSize } { true }
1829     }
1830     {
1831         \msg_warning:nnee
1832         {hyp}
1833         {ignore-deprecated-or-unknown-option-in-pdf-version}
1834         {pdfpicktraybypdfsize}
1835         {\pdf_version:}
1836     }
1837 }
1838 ,pdfpicktraybypdfsize / false .code:n =
1839 {
1840     \pdf_version_compare:NnTF > {1.6}
1841     {
1842         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1843         { PickTrayByPDFSize } { false }
1844     }
1845     {
1846         \msg_warning:nnee
1847         {hyp}
1848         {ignore-deprecated-or-unknown-option-in-pdf-version}
1849         {pdfpicktraybypdfsize}
1850         {\pdf_version:}
1851     }
1852 }
1853 ,pdfpicktraybypdfsize / .code:n =
1854 {
1855     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1856 }
1857 ,pdfpicktraybypdfsize / unknown .code:n =
1858 {
1859     \msg_warning:nnee { hyp } { no-bool }
1860     { picktraybypdfsize }
1861     { \exp_not:n {#1} }
1862 }
1863 ,pdfprintarea .choices:nn =
1864 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1865 {
1866     \pdf_version_compare:NnTF < {2.0}
1867     {
1868         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1869         { PrintArea } { /#1 }
1870     }
1871     {
1872         \msg_warning:nnee
1873         {hyp}
1874         {ignore-deprecated-or-unknown-option-in-pdf-version}
1875         {pdfprintarea}
1876         {\pdf_version:}
1877     }
1878 }%

```

```

1879 ,pdfprintarea / .code:n =
1880 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1881 ,pdfprintarea / unknown .code:n =
1882 {
1883   \msg_warning:nneee { hyp } { unknown-choice+empty }
1884   { pdfprintarea }
1885   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1886   { \exp_not:n {#1} }
1887 }
1888 ,pdfprintclip .choices:nn =
1889 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1890 {
1891   \pdf_version_compare:NnTF < {2.0}
1892   {
1893     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1894     { PrintClip } { /#1 }
1895   }
1896   {
1897     \msg_warning:nnee
1898     {hyp}
1899     {ignore-deprecated-or-unknown-option-in-pdf-version}
1900     {pdfprintclip}
1901     {\pdf_version:}
1902   }
1903 }%
1904 ,pdfprintclip / .code:n =
1905 {
1906   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1907 }
1908 ,pdfprintclip / unknown .code:n =
1909 {
1910   \msg_warning:nneee
1911   { hyp }
1912   { unknown-choice+empty }
1913   { pdfprintclip }
1914   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1915   { \exp_not:n {#1} }
1916 }
1917 ,pdfprintpagerange .code:n =
1918 {
1919   \pdf_version_compare:NnTF > {1.6}
1920   {
1921     \tl_if_empty:nTF { #1}
1922     {
1923       \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1924       { PrintPageRange }
1925     }
1926     {
1927       \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1928       {PrintPageRange}{[#1]}
1929     }
1930   }
1931   {
1932     \msg_warning:nnee

```

```

1933         {hyp}
1934         {ignore-deprecated-or-unknown-option-in-pdf-version}
1935         {pdfprintpagerange}
1936         {\pdf_version:}
1937     }
1938 }
1939 ,pdfprintscaling .choices:nn =
1940 { None, AppDefault }
1941 {
1942     \pdf_version_compare:NnTF > {1.5}
1943     {
1944         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1945         { PrintScaling } { /#1 }
1946     }
1947     {
1948         \msg_warning:nnee
1949         {hyp}
1950         {ignore-deprecated-or-unknown-option-in-pdf-version}
1951         {pdfprintscaling}
1952         {\pdf_version:}
1953     }
1954 }%
1955 ,pdfprintscaling / .code:n =
1956 {
1957     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1958 }
1959 ,pdfprintscaling / unknown .code:n =
1960 {
1961     \msg_warning:nnee { hyp } { unknown-choice+empty }
1962     { pdfprintarea }
1963     { None, AppDefault }
1964     { \exp_not:n {#1} }
1965 }
1966 ,pdfremotestartview .code:n =
1967 {
1968     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
1969     \exp_args:NNV
1970     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1971     {
1972         \tl_set:Ne \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1973     }
1974     {
1975         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1976         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1977     }
1978 }
1979 ,pdfremotestartview .initial:n = {Fit}
1980 % pdfstartpage is special as it shares code with pdfstartview
1981 ,pdfstartpage .code:n =
1982 {
1983     \tl_gset:Ne \g__hyp_dest_pdfstartpage_tl { #1 }
1984     \bool_if:nTF
1985     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1986     {

```

```

1987         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1988     }
1989     {
1990         \pdfmanagement_add:nne {Catalog} { OpenAction }
1991         {
1992             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1993         }
1994     }
1995 }
1996 ,pdfstartpage .initial:n =1
1997 ,pdfstartview .code:n =
1998 {
1999     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
2000     \exp_args:NNV
2001     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa
2002     {
2003         \tl_gset:Ne \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
2004     }
2005     {
2006         \msg_warning:nmmm {hyp}{invalid-destination-value}{#1}{pdfstartview}
2007         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
2008     }
2009     \bool_if:nTF
2010     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
2011     {
2012         \pdfmanagement_remove:nn {Catalog} { OpenAction }
2013     }
2014     {
2015         \pdfmanagement_add:nne {Catalog} { OpenAction }
2016         {
2017             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
2018         }
2019     }
2020 }
2021 ,pdfstartview .initial:n = Fit
2022 ,pdftoolbar .choice:
2023 ,pdftoolbar / true .code:n =
2024 {
2025     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
2026 }
2027 ,pdftoolbar / false .code:n =
2028 {
2029     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
2030     { HideToolbar } { true }
2031 }
2032 ,pdftoolbar / true .code:n =
2033 {
2034     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
2035 }
2036 ,pdftoolbar .default:n = true
2037 ,pdftoolbar / unknown .code:n =
2038 {
2039     \msg_warning:nnee { hyp } { no-bool }
2040     { pdftoolbar }

```

```

2041         { \exp_not:n {#1} }
2042     }
2043 % pdfview see below.
2044 ,pdfviewarea .choices:nn =
2045 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2046 {
2047     \pdf_version_compare:NnTF < {2.0}
2048     {
2049         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2050         { ViewArea } { /#1 }
2051     }
2052     {
2053         \msg_warning:nnee
2054         {hyp}
2055         {ignore-deprecated-or-unknown-option-in-pdf-version}
2056         {pdfviewarea}
2057         {\pdf_version:}
2058     }
2059 }%
2060 ,pdfviewarea / .code:n =
2061 {
2062     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
2063 }
2064 ,pdfviewarea / unknown .code:n =
2065 {
2066     \msg_warning:nneee { hyp } { unknown-choice+empty }
2067     { pdfviewarea }
2068     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2069     { \exp_not:n {#1} }
2070 }
2071 ,pdfviewclip .choices:nn =
2072 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2073 {
2074     \pdf_version_compare:NnTF < {2.0}
2075     {
2076         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2077         { ViewClip } { /#1 }
2078     }
2079     {
2080         \msg_warning:nnee
2081         {hyp}
2082         {ignore-deprecated-or-unknown-option-in-pdf-version}
2083         {pdfviewclip}
2084         {\pdf_version:}
2085     }
2086 }%
2087 ,pdfviewclip / .code:n =
2088 {
2089     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
2090 }
2091 ,pdfviewclip / unknown .code:n =
2092 {
2093     \msg_warning:nneee { hyp } { unknown-choice+empty }
2094     { pdfviewclip }

```

```

2095     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2096     { \exp_not:n {#1} }
2097   }
2098   ,pdfwindowui .choice:
2099   ,pdfwindowui / true .code:n =
2100   {
2101     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2102   }
2103   ,pdfwindowui / false .code:n =
2104   {
2105     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2106     { HideWindowUI } { true }
2107   }
2108   ,pdfwindowui / .code:n =
2109   {
2110     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2111   }
2112   ,pdfwindowui / unknown .code:n =
2113   {
2114     \msg_warning:nnee { hyp } { no-bool }
2115     { pdfwindowui }
2116     { \exp_not:n {#1} }
2117   }
2118   ,pdfwindowui .default:n = true
2119 }

```

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

```

2120 \keys_define:nn { hyp }
2121 {
2122   ,pdfview .code:n =
2123   {
2124     \seq_set_split:Nnn \l__hyp_tmpa_seq {-}{#1}
2125     \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
2126     {
2127       { xyz }
2128       {
2129         \int_compare:nNnTF { \seq_count:N \l__hyp_tmpa_seq } > { 1 }
2130         {
2131           \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2132           \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2133           {
2134             \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2135           }
2136           {
2137             \tl_set:Ne \l__hyp_dest_pdfview_tl
2138             {
2139               \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2140             }
2141           }
2142         }
2143         {
2144           \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2145         }
2146       }

```



```

2147 { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2148 { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2149 { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2150 { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2151 { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2152 { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2153 { fitr }
2154 {
2155   \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2156   {
2157     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2158   }
2159   {
2160     %ensure 4 values ...
2161     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2162     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2163     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2164     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2165     \hbox_set_to_wd:Nnn \l__hyp_dest_box
2166     {
2167       \fp_eval:n
2168       {
2169         round
2170         (
2171           abs
2172           (
2173             \seq_item:Nn\l__hyp_tmpa_seq{4}
2174             -
2175             (\seq_item:Nn\l__hyp_tmpa_seq{2})
2176           ),
2177           3
2178         )
2179       }bp
2180     }{}
2181     \box_set_dp:Nn \l__hyp_dest_box
2182     {
2183       \fp_eval:n
2184       {
2185         round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2186       }bp
2187     }
2188     \box_set_ht:Nn \l__hyp_dest_box
2189     {
2190       \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2191     }
2192   }
2193 }
2194 {
2195   \msg_warning:nmmn {hyp}{invalid-destination-value}{#1}{pdfview}
2196   \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2197 }
2198 }
2199 }
2200 ,pdfview .initial:n = {xyz}

```

```
2201 }
```

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.

```
2202 \keys_define:nn { hyp }
2203 {
2204   ,pdflang      .code:n =
2205   {
2206     \tl_if_empty:nF { #1 }
2207     {
2208       \pdfmanagement_add:nne {Catalog} { Lang } { (#1) }
2209       \AddToDocumentProperties[document]{lang}{#1}
2210     }
2211   }
2212 }
```

12.3.1 “info keys”

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

```
2213 \regex_new:N\l__hyp_optlang_regex
2214 \regex_set:Nn\l__hyp_optlang_regex {\A\{([A-Za-z\-\-]+)\}(.*)}
2215 \cs_generate_variant:Nn\clist_item:nn{on}
2216 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2217 {
2218   \keys_define:nn { hyp }
2219   {
2220     pdf#1 .code:n =
2221     {
2222       \tl_set:Ne\l__hyp_tmpa_tl {\text_expand:n{##1}}
2223       \__hyp_store_metadata:no {pdf#1}{\l__hyp_tmpa_tl}
2224       \tl_if_empty:NTF \l__hyp_tmpa_tl
2225       {
2226         \str_case:nnF { #1 }
2227         {
2228           {creator}
2229           {
2230             \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2231             \pdfmanagement_add:nne {Info}{Creator}{()}
2232           }
2233           {producer}
2234           {
2235             \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2236             \pdfmanagement_add:nne {Info}{Producer}{()}
2237           }
2238         }
2239       }

```

```

2240         \pdfmanagement_remove:nn {Info}{#2}
2241     }
2242 }
2243 {
2244     \tl_set:N\l__hyp_tmpb_tl {\clist_item:on{\l__hyp_tmpa_tl}{1}}
2245     \exp_args:NNV
2246     \regex_extract_once:NnN \l__hyp_optlang_regex \l__hyp_tmpb_tl\l__hyp_tmpa_s
2247     \seq_if_empty:NTF\l__hyp_tmpa_seq
2248     {
2249         \__hyp_text_pdfstring_info:oN {\l__hyp_tmpa_tl}\l__hyp_tmpa_str
2250     }
2251     {
2252         \__hyp_text_pdfstring_info:eN {\seq_item:Nn \l__hyp_tmpa_seq{3}}\l__hyp_t
2253     }
2254     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2255     {
2256         \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2257     }
2258 }
2259 }
2260 }
2261 \keys_define:nn { hyp / info }
2262 {
2263     #2 .code:n =
2264     {
2265         \tl_set:N\l__hyp_tmpa_tl {\text_expand:n{##1}}
2266         \__hyp_store_metadata:eo {pdf\str_lowercase:n{##1}}{\l__hyp_tmpa_tl}
2267         \tl_if_blank:nTF {##1}
2268         {
2269             \pdfmanagement_remove:nn {Info}{#2}
2270         }
2271         {
2272             \__hyp_text_pdfstring_info:oN {\l__hyp_tmpa_tl}\l__hyp_tmpa_str
2273             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2274             {
2275                 \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2276             }
2277         }
2278     }
2279     ,unknown .code:n =
2280     {
2281         \__hyp_text_pdfstring_info:eN {##1}\l__hyp_tmpa_str
2282         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2283         {
2284             \exp_args:Nno
2285             \pdfmanagement_add:nne {Info}
2286                 { \l_keys_key_str } {\l__hyp_tmpa_str}
2287         }
2288     }
2289 }
2290 }
2291 \__hyp_setup_info_key:nn {author} {Author}
2292 \__hyp_setup_info_key:nn {title} {Title}
2293 \__hyp_setup_info_key:nn {producer} {Producer}

```

```

2294 \_hyp_setup_info_key:nn {creator} {Creator}
2295 % ignored key: addtopdfcreator
2296 \_hyp_setup_info_key:nn {subject} {Subject}
2297 \_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)
2298 \cs_new_protected:Npn \_hyp_setup_info_date_key:nn #1 #2
2299 {
2300   \keys_define:nn { hyp }
2301   {
2302     pdf#1 .code:n =
2303     {
2304       \tl_if_blank:nTF {##1}
2305       {
2306         \pdfmanagement_remove:nn {Info}{#2}
2307       }
2308       {
2309         \pdfmanagement_add:nne {Info}{#2}{(##1)}
2310       }
2311       \_hyp_store_metadata:nn {pdf#1}{##1}
2312       \AddToDocumentProperties[document]{#1}{##1}
2313     }
2314   }
2315   \keys_define:nn { hyp / info }
2316   {
2317     #2 .code:n =
2318     {
2319       \tl_if_blank:nTF {##1}
2320       {
2321         \pdfmanagement_remove:nn {Info}{#2}
2322       }
2323       {
2324         \pdfmanagement_add:nne {Info}{#2}{(##1)}
2325       }
2326       \exp_args:Ne \_hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2327     }
2328   }
2329 }
2330
2331 \_hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2332 \_hyp_setup_info_date_key:nn {moddate} {ModDate}
2333 \keys_define:nn { hyp }
2334 {
2335   pdfmetadate .code:n = { \_hyp_store_metadata:nn {pdfmetadate}{#1} }
2336 }

```

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

```

2337 \keys_define:nn { hyp }
2338 {
2339   ,pdftrapped .code:n =
2340   {
2341     \exp_args:Nne
2342     \keys_set:nn { hyp } { _pdftrapped = \str_uppercase:n { #1 } }

```

```

2343     }
2344     ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2345     {
2346         \pdfmanagement_add:nne {Info}{Trapped}
2347         {/
2348             \str_uppercase:f { \str_head:n { #1 } }
2349             \str_lowercase:f { \str_tail:n { #1 } }
2350         }
2351         \__hyp_store_metadata:ne {pdftrapped}
2352         {
2353             \str_uppercase:f { \str_head:n { #1 } }
2354             \str_lowercase:f { \str_tail:n { #1 } }
2355         }
2356     }
2357     ,_pdftrapped / unknown .code:n =
2358     {
2359         \msg_warning:nneee { hyp } { unknown-choice }
2360         { pdftrapped }
2361         { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2362         { \exp_not:n {#1} }
2363     }
2364 }

```

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

```

2365 \keys_define:nn { hyp }
2366 {
2367     pdfinfo .code:n =
2368     {
2369         \keys_set:nn { hyp / info } { #1 }
2370     }
2371 }

```

Now we set some default values

```

2372 \keys_set:nn { hyp } {pdfcreator = LaTeX-with-hyperref}
2373 \keys_set:nn { hyp } {pdfauthor = }
2374 \keys_set:nn { hyp } {pdftitle = }
2375 \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:

```

2376 \clist_map_inline:nn
2377 {
2378     ,pdfcopyright
2379     ,pdftype
2380     ,pdflicenseurl
2381     ,pdfauthortitle
2382     ,pdfcaptionwriter
2383     ,pdfmetalang
2384     ,pdfsource
2385     ,pdfdocumentid
2386     ,pdfinstanceid

```

```

2387 ,pdfversionid
2388 ,pdfrendition
2389 ,pdfpublication
2390 ,pdfpubtype
2391 ,pdfbytes
2392 ,pdfnumpages
2393 ,pdfissn
2394 ,pdfeissn
2395 ,pdfisbn
2396 ,pdfbookedition
2397 ,pdfpublisher
2398 ,pdfvolumenum
2399 ,pdfissuenum
2400 ,pdfpagerange
2401 ,pdfdoi
2402 ,pdfurl
2403 ,pdfidentifier
2404 ,pdfsubtitle
2405 ,pdfpubstatus
2406 ,pdfcontactaddress
2407 ,pdfcontactcity
2408 ,pdfcontactregion
2409 ,pdfcontactpostcode
2410 ,pdfcontactcountry
2411 ,pdfcontactphone
2412 ,pdfcontactemail
2413 ,pdfcontacturl
2414 ,pdfdate
2415 }
2416 {
2417   \keys_define:nn { hyp }
2418   {
2419     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2420   }
2421 }
2422

```

12.5 Transitions

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

```

2423 \keys_define:nn { hyp }
2424 {
2425   pdfpageduration .code:n =
2426   {
2427     \tl_if_blank:nTF { #1 }
2428     {
2429       \pdfmanagement_remove:nn {Page}{Dur}
2430     }
2431     {
2432       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2433     }
2434   }
2435 }

```

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

```
2436 \keys_define:nn { hyp }
2437 {
2438   pdfpagetransition .code:n =
2439   {
2440     \tl_if_blank:nTF {#1}
2441     {
2442       \pdfmanagement_remove:nn {Page}{Trans}
2443     }
2444     {
2445       \group_begin:
2446       \keys_set:nn { hyp / trans }{style=R,#1}
2447       \pdf_object_unnamed_write:ne { dict }
2448       {
2449         \pdfdict_use:n {l__hyp_page/Trans}
2450       }
2451       \pdfmanagement_add:nne {Page}{Trans}{\pdf_object_ref_last:}
2452       \group_end:
2453     }
2454   }
2455 }
2456 \keys_define:nn { hyp / trans }
2457 {
2458   ,style .choices:nn =
2459   {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
```

```

2460     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2461 ,style / unknown .code:n =
2462     {
2463     \msg_warning:nneee { hyp } { unknown-choice }
2464     { trans / style }
2465     { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2466     { \exp_not:n {#1} }
2467     }
2468 ,duration .code:n =
2469     {
2470     \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}
2471     }
2472 ,direction .choices:nn =
2473     {H,V}
2474     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2475 ,direction .choices:nn =
2476     {0,90,180,270,315}
2477     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2478 ,direction / None .code:n =
2479     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2480 ,direction / unknown .code:n =
2481     {
2482     \msg_warning:nneee { hyp } { unknown-choice }
2483     { trans / direction }
2484     {
2485     H~(horizontal,~only~Split,~Blinds),
2486     V~(vertical,~only~Split,~Blinds),
2487     0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2488     90~(bottom~to~top,~only~Wipe),
2489     180~(right~to~left,~only~Wipe),
2490     270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2491     315~(top~left~to~bottom,~only~Glitter),
2492     None~(only~Fly)
2493     }
2494     { \exp_not:n {#1} }
2495     }
2496 ,motion .choices:nn =
2497     {I,0}
2498     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2499 ,motion / unknown .code:n =
2500     {
2501     \msg_warning:nneee { hyp } { unknown-choice }
2502     { trans / motion }
2503     { I~(inwards) , 0~(outwards) }
2504     { \exp_not:n {#1} }
2505     }
2506 ,scale .code:n =
2507     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2508 ,opaque .choices:nn = {true,false}
2509     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2510 ,opaque / unknown .code:n =
2511     {
2512     \msg_warning:nneee { hyp } { unknown-choice }
2513     { trans / B }

```



```

2514         { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2515         { \exp_not:n {#1} }
2516     }
2517     % try to set unknown keys as style
2518     ,unknown .code:n =
2519     {
2520         % warning ...
2521         \exp_args:Nne\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2522     }
2523 }

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

```

```

2565
2566 \def\HyField@AddToFields
2567 {
2568   \exp_args:N\HyField@@AddToFields
2569   {
2570     \pdfannot_box_ref_last:
2571   }
2572   \ifx\Fld@calculate@code\empty
2573   \else
2574     \begingroup
2575     \Hy@safe@activestrue
2576     \edef\Hy@temp{%
2577       \endgroup
2578       \if@filesw
2579         \write\@mainaux
2580         {
2581           \string\HyField@AuxAddToCoFields
2582           {
2583             \Fld@calculate@sortkey
2584           }
2585           {
2586             \pdfannot_box_ref_last:
2587           }
2588         }
2589       \fi
2590     }%
2591     \Hy@temp
2592   \fi
2593 }%
2594
2595 \def\HyField@@AddToFields#1{
2596 \HyField@AfterAuxOpen{%
2597   \if@filesw
2598     \write\@mainaux{%
2599       \string\HyField@AuxAddToFields{#1}%
2600     }%
2601   \fi
2602 }%
2603 }%
2604
2605 \ExplSyntaxOff
2606 \ExplSyntaxOn
2607
2608 \def\@Form[#1]
2609 {
2610   \kvsetkeys{Form}{#1}
2611   \pdf@ifdraftmode{}
2612   {
2613     \Hy@FormObjects
2614     \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2615     {
2616       \pdfmanagement_add:nne { Catalog / AcroForm } { Fields }{##1}
2617       %\pdfmanagement_show:n { Catalog / AcroForm }
2618     }

```

```

2619     \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2620     {
2621       \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2622       {
2623         \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2624       }
2625       \seq_sort:Nn \l__hyp_tmpa_seq
2626       {
2627         \str_compare:nNnTF {##1} > {##2}
2628         { \sort_return_swapped: }
2629         { \sort_return_same: }
2630       }
2631       \seq_map_inline:Nn \l__hyp_tmpa_seq
2632       {
2633         \pdfmanagement_add:nne { Catalog / AcroForm }
2634         { CO }
2635         {
2636           \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2637         }
2638       }
2639     }
2640     \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2641     {ZaDb} {\pdf_object_ref:n {\__hyp/Font/ZaDb} }
2642     \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2643     {Helv} {\pdf_object_ref:n {\__hyp/Font/Helv} }
2644     \pdfmanagement_add:nne {Catalog /AcroForm}
2645     {DA}{(/Helv~10~Tf~0~g)}
2646     \pdfmeta_standard_verify:nTF {form_no_NeedAppearance}
2647     {
2648       \legacy_if:nT { HyField@NeedAppearances }
2649       {
2650         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2651       }
2652     }
2653     {
2654       \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2655     }
2656   }
2657 }
2658 \ExplSyntaxOff
2659 \let\@endForm\@empty
2660 \let\HyAnn@AbsPageLabel\@empty
2661 \let\Fld@pageobjref\@empty
2662
2663 \ExplSyntaxOn
2664 \newcount\HyAnn@Count
2665 \HyAnn@Count=\z@
2666 \def\HyAnn@AbsPageLabel
2667 {
2668   \global\advance\HyAnn@Count by\@ne
2669   \__hyp_property_record:ee {HyAnn@\the\HyAnn@Count}{abspage}
2670   \property_ref_undefined_warn:ee {HyAnn@\the\HyAnn@Count}{abspage}
2671 }%
2672 \prg_generate_conditional_variant:Nnn \property_if_recorded:nm {ee} {T}

```

```

2673 \def\Fld@pageobjref
2674 {
2675   \property_if_recorded:eeT {HyAnn@\the\HyAnn@Count}{abspage}
2676   {
2677     /P~\pdf_pageobject_ref:e
2678     {
2679       \property_ref:ee{HyAnn@\the\HyAnn@Count}{abspage}
2680     }
2681   }
2682 }
2683 %% check if the attr should be set through
2684 %% hooks.
2685 %% check if options are missing.
2686 \def\@TextField[#1]#2{% parameters, label
2687   \def\Fld@name{#2}%
2688   \let\Fld@default\@empty
2689   \let\Fld@value\@empty
2690   \def\Fld@width{\DefaultWidthofText}%
2691   \def\Fld@height{%
2692     \ifFld@multiline
2693       \DefaultHeightofTextMultiline
2694     \else
2695       \DefaultHeightofText
2696     \fi
2697   }%
2698   \begingroup
2699   \expandafter\HyField@SetKeys\expandafter{%
2700     \DefaultOptionsofText,#1%
2701   }%
2702   \PDFForm@Name
2703   \HyField@FlagsText
2704   \ifFld@hidden\def\Fld@width{1sp}\fi
2705   \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2706   \LayoutTextField{#2}{%
2707     \leavevmode
2708     \HyAnn@AbsPageLabel
2709     \Hy@escapeform\PDFForm@Text
2710     \pdfannot_box:nnnn
2711       {\Fld@width}
2712       {\Fld@height}
2713       {Opt} %is this correct?
2714       {\PDFForm@Text}
2715     \MakeTextField{\Fld@width}{\Fld@height}
2716     \HyField@AddToFields
2717   }%
2718   \endgroup
2719 }
2720 \providecommand\@curropt{}
2721 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2722   \def\Fld@name{#2}
2723   \let\Fld@default\relax
2724   \let\Fld@value\relax
2725   \def\Fld@width{\DefaultWidthofChoiceMenu}
2726   \def\Fld@height{\DefaultHeightofChoiceMenu}

```

```

2727 \begingroup
2728   \Fld@menulength=0 %
2729   \@tempdima\z@
2730   \clist_map_variable:nNn { #3 } \@curropt
2731   %\@for\@curropt:=#3\do
2732   {%
2733     \expandafter\Fld@checkequals\@curropt==\%
2734     \Hy@StepCount\Fld@menulength
2735     \settowidth{\@tempdimb}{\@currDisplay}%
2736     \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2737   }%
2738   \advance\@tempdima by~15\p@
2739   \begingroup
2740     \HyField@SetKeys{#1}
2741   \edef\x{\endgroup
2742     \noexpand\expandafter
2743     \noexpand\HyField@SetKeys
2744     \noexpand\expandafter{%
2745       \expandafter\noexpand\csgname DefaultOptionsof%
2746       \ifFld@radio
2747         Radio%
2748       \else
2749         \ifFld@combo
2750           \ifFld@popdown
2751             PopdownBox%
2752           \else
2753             ComboBox%
2754           \fi
2755         \else
2756           ListBox%
2757         \fi
2758       \fi
2759     \endcsname
2760   }%
2761   }\x
2762   \HyField@SetKeys{#1}%
2763   \PDFForm@Name
2764   \ifFld@hidden\def\Fld@width{1sp}\fi
2765   \ifx\Fld@value\relax
2766     \let\Fld@value\Fld@default
2767   \fi
2768   \LayoutChoiceField{#2}{%
2769     \ifFld@radio
2770       \HyField@FlagsRadioButton
2771       \@@Radio{#3}%
2772     \else
2773       \begingroup
2774         \HyField@FlagsChoice
2775         \ifdim\Fld@width<\@tempdima
2776           \ifdim\@tempdima<1cm\@tempdima1cm\fi
2777           \edef\Fld@width{\the\@tempdima}%
2778         \fi
2779         \ifFld@combo
2780         \else

```

```

2781         \@tempdima=\the\Fld@menulength\Fld@charsize
2782         \advance\@tempdima by-\Fld@borderwidth bp %
2783         \advance\@tempdima by-\Fld@borderwidth bp %
2784         \edef\Fld@height{\the\@tempdima}%
2785     \fi
2786     \@ListBox{#3}%
2787 \endgroup
2788 \fi
2789 }%
2790 \endgroup
2791 }
2792
2793 \def\@@Radio#1{%
2794     \Fld@listcount=0-%
2795     \EdefEscapeName\Fld@default{\Fld@default}%
2796     \clist_map_variable:nNn { #1 } \@curropt
2797     {%
2798         \expandafter\Fld@checkequals\@curropt==\%
2799         \EdefEscapeName\@currValue{\@currValue}%
2800         \Hy@StepCount\Fld@listcount
2801         \@currDisplay\space
2802         \leavevmode
2803         \HyAnn@AbsPageLabel
2804         \Hy@escapeform\PDFForm@Radio
2805         \pdfxform_if_exist:nF { __hyp_xform_Ding }
2806         {
2807             \pdfxform_new:nnn { __hyp_xform_Ding } {}
2808             {
2809                 \group_begin:
2810                 \fontfamily{pzd}
2811                 \fontencoding{U}
2812                 \fontseries{m}
2813                 \fontshape{n}
2814                 \selectfont
2815                 \char123
2816                 \group_end:
2817             }
2818         }
2819         \pdfannot_box:nnne
2820         {\Fld@width}
2821         {\Fld@height}
2822         {0pt} %is this correct?
2823         {
2824             \PDFForm@Radio
2825             /AP
2826             <<
2827             /N
2828             <<
2829             /\@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2830             %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2831             >>
2832             >>
2833         }
2834         {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }

```

```

2835     \int_compare:nNt { \Fld@listcount } = { 1 }
2836     { \HyField@AddToFields }
2837     \c_space_tl % deliberate space between radio buttons
2838             % to do: --> should be configurable
2839 }%
2840 }
2841
2842 \newcount\Fld@listcount
2843 \def\@@ListBox#1
2844 {
2845     \HyField@PDFChoices{#1}
2846     \mode_leave_vertical:
2847     \HyAnn@AbsPageLabel
2848     \Hy@escapeform\PDFForm@List
2849     \pdfannot_box:nnnn
2850     {\Fld@width}{\Fld@height}{Opt}
2851     {\PDFForm@List}
2852     \MakeChoiceField{\Fld@width}{\Fld@height}
2853     \HyField@AddToFields
2854 }
2855
2856
2857 \def\@PushButton[#1]#2{% parameters, label
2858     \def\Fld@name{#2}%
2859     \group_begin:
2860         \exp_args:No\HyField@SetKeys
2861         {
2862             \DefaultOptionsofPushButton,#1
2863         }
2864         \PDFForm@Name
2865         \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2866         {
2867             \HyField@FlagsPushButton
2868             \legacy_if:nT {Fld@hidden}
2869             {
2870                 \def\Fld@width{1sp}
2871             }
2872             \LayoutPushButtonField
2873             {
2874                 \mode_leave_vertical:
2875                 \HyAnn@AbsPageLabel
2876                 \Hy@escapeform\PDFForm@Push
2877                 \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2878                 \pdfannot_box:nnnn
2879                 {\box_wd:N\l_tmpa_box}
2880                 {\box_ht:N\l_tmpa_box}
2881                 {\box_dp:N\l_tmpa_box} %is this correct?
2882                 {\PDFForm@Push}
2883                 {\box_use:N\l_tmpa_box}
2884                 \HyField@AddToFields
2885             }
2886         }
2887     {
2888         \msg_error:nn { hyp }{ pdfa-no-push-button }

```

```

2889         \LayoutPushButtonField
2890         {
2891             \mode_leave_vertical:
2892             \MakeButtonField{#2}
2893         }
2894     }
2895 \group_end:
2896 }
2897
2898 \def\@Submit[#1]#2
2899 {
2900     \def\Fld@width {\DefaultWidthofSubmit}
2901     \def\Fld@height{\DefaultHeightofSubmit}
2902     \group_begin:
2903         \exp_args:No\HyField@SetKeys
2904         {
2905             \DefaultOptionsofSubmit,#1
2906         }
2907     \HyField@FlagsPushButton
2908     \HyField@FlagsSubmit
2909     \legacy_if:nT { Fld@hidden }
2910     {
2911         \def\Fld@width{1sp}
2912     }
2913     \mode_leave_vertical:
2914     \HyAnn@AbsPageLabel
2915     \Hy@escapeform\PDFForm@Submit
2916     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2917     \pdfxform_if_exist:nF
2918     { __hyp_xform_Submit }
2919     {
2920         \pdfxform_new:nnn { __hyp_xform_Submit }{-}
2921         {
2922             \fbox{\color_select:n{yellow}\textsf{Submit}}
2923         }
2924         \pdfxform_new:nnn { __hyp_xform_SubmitP }{-}
2925         {
2926             \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2927         }
2928     }
2929     \pdfannot_box:nnnn
2930     {\box_wd:N\l_tmpa_box}
2931     {\box_ht:N\l_tmpa_box}
2932     {\box_dp:N\l_tmpa_box} %is this correct?
2933     {
2934         \PDFForm@Submit
2935         /AP<<
2936         /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2937         /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2938         >>
2939     }
2940     \HyField@AddToFields
2941     \box_use:N\l_tmpa_box
2942

```



```

2943     \group_end:
2944 }
2945
2946 \def\@Reset[#1]#2
2947 {
2948   \def\Fld@width {\DefaultWidthofReset}
2949   \def\Fld@height{\DefaultHeightofReset}
2950   \group_begin:
2951     \exp_args:No\HyField@SetKeys
2952     {
2953       \DefaultOptionsofReset,#1
2954     }
2955     \mode_leave_vertical:
2956     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2957     {
2958       \HyField@FlagsPushButton
2959       \legacy_if:nT { Fld@hidden }
2960       { \def\Fld@width{1sp} }
2961       \HyAnn@AbsPageLabel
2962       \Hy@escapeform\PDFForm@Reset
2963       \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2964       \pdfannot_box:nnnn
2965       {\box_wd:N\l_tmpa_box}
2966       {\box_ht:N\l_tmpa_box}
2967       {\box_dp:N\l_tmpa_box} %is this correct?
2968       { \PDFForm@Reset }
2969       \HyField@AddToFields
2970       \box_use:N \l_tmpa_box
2971     }
2972     {
2973       \msg_error:nn { hyp }{ pdfa-no-reset-button }
2974       \MakeButtonField{#2}
2975     }
2976   \group_end:
2977 }
2978
2979 \def\@CheckBox[#1]#2
2980 {% parameters, label
2981   \def\Fld@name{#2}
2982   \def\Fld@default{0}
2983   \group_begin:
2984     \def\Fld@width {\DefaultWidthofCheckBox}
2985     \def\Fld@height{\DefaultHeightofCheckBox}
2986     \exp_args:No\HyField@SetKeys
2987     {
2988       \DefaultOptionsofCheckBox,#1
2989     }
2990     \PDFForm@Name
2991     \HyField@FlagsCheckBox
2992     \legacy_if:nT { Fld@hidden }
2993     {
2994       \def\Fld@width{1sp}
2995     }
2996     \LayoutCheckField{#2}

```

```

2997     {
2998     \mode_leave_vertical:
2999     \HyAnn@AbsPageLabel
3000     \Hy@escapeform\PDFForm@Check
3001     \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
3002     {
3003         \pdfxform_new:nnn
3004         {__hyp_xform_CheckMarkYes}{-}
3005         {
3006             \group_begin:
3007             \fontfamily{pzd}
3008             \fontencoding{U}
3009             \fontseries{m}
3010             \fontshape{n}
3011             \selectfont
3012             \char51
3013             \group_end:
3014         }
3015         \pdfxform_new:nnn
3016         {__hyp_xform_CheckMarkOff}{-}
3017         {
3018             \group_begin:
3019             \fontfamily{pzd}
3020             \fontencoding{U}
3021             \fontseries{m}
3022             \fontshape{n}
3023             \selectfont
3024             \phantom{\char51} %perhaps xetex needs some small glyph ..
3025             \group_end:
3026         }
3027     }
3028     \pdfannot_box:nnnn
3029     {\Fld@width}
3030     {\Fld@height}
3031     {\Opt} %is this correct?
3032     {\PDFForm@Check}
3033     \HyField@AddToFields %check if this works with xelatex ...
3034 }
3035 \group_end:
3036 }
3037 %hm. Should a luatex driver use type1 fonts in fields????
3038 \def\Hy@FormObjects
3039 {
3040     \pdf_object_new:n {__hyp/Encoding/pdfdoc }
3041     \pdf_object_new:n {__hyp/Font/ZaDb }
3042     \pdf_object_new:n {__hyp/Font/Helv }
3043     \pdf_object_write:nne {__hyp/Encoding/pdfdoc } { dict }
3044     {
3045         /Type/Encoding
3046         /Differences[
3047             24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
3048             /ring/tilde
3049             \c_space_tl
3050             39/quotesingle

```

```

3051     \c_space_tl
3052     96/grave %
3053     \iow_newline:
3054     128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
3055     /fraction/guilsinglleft/guilsinglright/minus/perthousand
3056     /quotedblbase/quotedblleft/quotedblright/quoteleft
3057     /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
3058     /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
3059     \iow_newline:
3060     164/currency
3061     \c_space_tl
3062     166/brokenbar
3063     \c_space_tl
3064     168/dieresis/copyright/ordfeminine
3065     \c_space_tl
3066     172/logicalnot/.notdef/registered/macron/degree/plusminus
3067     /twosuperior/threesuperior/acute/mu
3068     \c_space_tl
3069     183/periodcentered/cedilla/onesuperior/ordmasculine
3070     \c_space_tl
3071     188/onequarter/onehalf/threequarters
3072     \iow_newline:
3073     192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
3074     /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
3075     /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
3076     /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
3077     /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
3078     /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
3079     /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
3080     /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
3081     /ocircumflex/otilde/odieresis/divide/oslash/ugrave
3082     /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
3083   ]
3084 }
3085 \pdf_object_write:nnn {__hyp/Font/ZaDb } { dict }
3086 {
3087   /Type/Font
3088   /Subtype/Type1
3089   /Name/ZaDb
3090   /BaseFont/ZapfDingbats
3091 }
3092 \pdf_object_write:nne {__hyp/Font/Helv } { dict }
3093 {
3094   /Type/Font
3095   /Subtype/Type1
3096   /Name/Helv
3097   /BaseFont/Helvetica
3098   /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
3099 }
3100 \global\let\Hy@FormObjects\relax
3101 }
3102 \ExplSyntaxOff
3103 \providecommand*{\Fld@pageobjref}{\}
3104 \ifcsname pdf@escapestring\endcsname

```

```

3105 \def\Hy@escapeform#1{%
3106   \ifHy@pdfescapeform
3107     \let\Hy@escapestring\pdfescapestring
3108   \else
3109     \let\Hy@escapestring\@firstofone
3110   \fi
3111 }%
3112 \Hy@escapeform{}%
3113 \else
3114 \let\Hy@escapestring\@firstofone
3115 \def\Hy@escapeform#1{%
3116   \ifHy@pdfescapeform
3117     \def\Hy@escapestring##1{%
3118       \noexpand\Hy@escapestring{\noexpand##1}%
3119     }%
3120   \edef\Hy@temp{#1}%
3121   \expandafter\Hy@@escapeform\Hy@temp\Hy@escapestring{}\@nil
3122   \def\Hy@escapestring##1{%
3123     \@ifundefined{Hy@esc@\string##1}{%
3124       ##1%
3125       \ThisShouldNotHappen
3126     }{%
3127       \csname Hy@esc@\string##1\endcsname
3128     }%
3129   }%
3130   \else
3131     \let\Hy@escapestring\@firstofone
3132   \fi
3133 }%
3134 \def\Hy@@escapeform#1\Hy@escapestring#2#3\@nil{%
3135   \ifx\#3\%
3136   \else
3137     \expandafter
3138     \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3139     \Hy@ReturnAfterFi{%
3140       \Hy@@escapeform#3\@nil
3141     }%
3142   \fi
3143 }%
3144 \fi
3145 \def\PDFForm@Name{%
3146   \PDFForm@@Name\Fld@name
3147   \ifx\Fld@altname\relax
3148   \else
3149     \PDFForm@@Name\Fld@altname
3150   \fi
3151   \ifx\Fld@mappingname\relax
3152   \else
3153     \PDFForm@@Name\Fld@mappingname
3154   \fi
3155 }
3156 \def\PDFForm@@Name#1{%
3157   \begingroup
3158   \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7

```

```

3159     \ifHy@unicode
3160     \Hy@unicodedefalse
3161     \fi
3162     \fi
3163     \pdfstringdef\Hy@gtemp#1%
3164 \endgroup
3165 \let#1\Hy@gtemp
3166 }
3167 \def\Fld@X@additionalactions{%
3168 \ifx\Fld@keystroke@code\@empty
3169 \else
3170 /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3171 \fi
3172 \ifx\Fld@format@code\@empty
3173 \else
3174 /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3175 \fi
3176 \ifx\Fld@validate@code\@empty
3177 \else
3178 /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3179 \fi
3180 \ifx\Fld@calculate@code\@empty
3181 \else
3182 /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3183 \fi
3184 \ifx\Fld@onfocus@code\@empty
3185 \else
3186 /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3187 \fi
3188 \ifx\Fld@onblur@code\@empty
3189 \else
3190 /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3191 \fi
3192 \ifx\Fld@onmousedown@code\@empty
3193 \else
3194 /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3195 \fi
3196 \ifx\Fld@onmouseup@code\@empty
3197 \else
3198 /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3199 \fi
3200 \ifx\Fld@onenter@code\@empty
3201 \else
3202 /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3203 \fi
3204 \ifx\Fld@onexit@code\@empty
3205 \else
3206 /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3207 \fi
3208 }
3209 \ExplSyntaxOn
3210 \def\Fld@additionalactions
3211 {%
3212 \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{ }

```

```

3213     {
3214         \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3215         {/AA<<\Fld@X@additionalactions>>}
3216     }
3217 }
3218 \def\Fld@annotnames{%
3219 /T(\Fld@name)%
3220 \ifx\Fld@altname\relax
3221 \else
3222 /TU(\Fld@altname)%
3223 \fi
3224 \ifx\Fld@mappingname\relax
3225 \else
3226 /TM(\Fld@mappingname)%
3227 \fi
3228 }
3229
3230 \def\PDFForm@Check
3231 {
3232 /Subtype/Widget
3233 ~\Fld@annotflags
3234 ~\Fld@pageobjref
3235 ~\Fld@annotnames
3236 /FT/Btn
3237 \Fld@flags
3238 /Q~\Fld@align
3239 /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3240 /AP
3241 <<
3242 /N
3243 <<
3244 /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3245 /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3246 >>
3247 >>
3248 /MK<<
3249 \int_compare:nNnF {\Fld@rotation}={0}
3250 {
3251 /R~\Fld@rotation
3252 }
3253 \tl_if_empty:NF\Fld@bordercolor
3254 {
3255 /BC[\Fld@bordercolor]
3256 }
3257 \tl_if_empty:NF\Fld@bcolor
3258 {
3259 /BG[\Fld@bcolor]
3260 }
3261 /CA(\Hy@escapestring{\Fld@cbsymbol})%
3262 >>
3263 /DA
3264 (
3265 /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3266 \tl_if_empty:NF \Fld@color

```

```

3267     {
3268         \c_space_tl \Fld@color
3269     }
3270 )
3271 /H/P
3272 \legacy_if:nTF {Fld@checked}
3273 {
3274     /V/Yes /AS/Yes
3275 }
3276 {
3277     /V/Off /AS/Off
3278 }
3279 \Fld@additionalactions
3280 }
3281
3282 \def\PDFForm@Push
3283 {
3284     /Subtype/Widget
3285     ~\Fld@annotflags
3286     ~\Fld@pageobjref
3287     ~\Fld@annotnames
3288     /FT/Btn
3289     ~\Fld@flags
3290     /H/P
3291     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3292     \bool_if:nT
3293     {
3294         !\int_compare_p:nNn {\Fld@rotation} = {0}
3295         ||
3296         \tl_if_exist_p:N \Fld@bordercolor
3297     }
3298     {
3299         /MK
3300         <<
3301             \int_compare:nNnF {\Fld@rotation} = {0}
3302             {
3303                 /R~\Fld@rotation
3304             }
3305             \tl_if_exist:NT \Fld@bordercolor
3306             {
3307                 /BC[\Fld@bordercolor]
3308             }
3309         >>
3310     }
3311     /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3312     \Fld@additionalactions
3313 }
3314
3315 \ExplSyntaxOff
3316 \def\PDFForm@List{%
3317     /Subtype/Widget%
3318     \Fld@annotflags
3319     \Fld@pageobjref
3320     \Fld@annotnames

```

```

3321 /FT/Ch%
3322 \Fld@flags
3323 /Q \Fld@align
3324 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3325 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3326     \ifx\Fld@bordercolor\relax\else 1\fi
3327     \ifx\Fld@bcolor\relax \else 1\fi
3328     \space
3329 \else
3330 /MK<<%
3331     \ifnum\Fld@rotation=\z@
3332     \else
3333     /R \Fld@rotation
3334     \fi
3335     \ifx\Fld@bordercolor\relax
3336     \else
3337     /BC[\Fld@bordercolor]%
3338     \fi
3339     \ifx\Fld@bcolor\relax
3340     \else
3341     /BG[\Fld@bcolor]%
3342     \fi
3343     >>%
3344 \fi
3345 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3346     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3347 \Fld@choices
3348 \Fld@additionalactions
3349 }
3350 \ExplSyntaxOn
3351 \def\PDFForm@Radio
3352 {
3353     /Subtype/Widget
3354     ~\Fld@annotflags
3355     ~\Fld@pageobjref
3356     ~\Fld@annotnames
3357     /FT/Btn
3358     \Fld@flags
3359     /H/P
3360     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3361     /MK<<
3362     \ifnum\Fld@rotation=\z@
3363     \else
3364     /R~\Fld@rotation
3365     \fi
3366     \ifx\Fld@bordercolor\relax
3367     \else
3368     /BC[\Fld@bordercolor]%
3369     \fi
3370     \ifx\Fld@bcolor\relax
3371     \else
3372     /BG[\Fld@bcolor]%
3373     \fi
3374     /CA(\Hy@escapestring{\Fld@radiosymbol})%

```



```

3375 >>
3376 /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3377 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3378 \ifx\Fld@default\@empty
3379 /V/Off%
3380 /DV/Off%
3381 \else
3382 /V/\Fld@default
3383 /DV/\Fld@default
3384 \fi
3385 \Fld@additionalactions
3386 }
3387
3388 % Does an appearance dict make sense here?
3389 \def\PDFForm@Text
3390 {
3391 /Subtype/Widget
3392 ~\Fld@annotflags
3393 ~\Fld@pageobjref
3394 ~\Fld@annotnames
3395 /FT/Tx
3396 ~\Fld@flags
3397 /Q~\Fld@align
3398 /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3399 \bool_if:nT
3400 {
3401 !\int_compare_p:nNn {\Fld@rotation} = {0}
3402 ||
3403 \tl_if_exist_p:N \Fld@bordercolor
3404 ||
3405 \tl_if_exist_p:N \Fld@bcolor
3406 }
3407 {
3408 /MK
3409 <<
3410 \int_compare:nNnF {\Fld@rotation} = {0}
3411 {
3412 /R~\Fld@rotation
3413 }
3414 \tl_if_exist:NT \Fld@bordercolor
3415 {
3416 /BC[\Fld@bordercolor]
3417 }
3418 \tl_if_exist:NT \Fld@bcolor
3419 {
3420 /BG[\Fld@bcolor]
3421 }
3422 >>
3423 }
3424 /DA
3425 (
3426 /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3427 \tl_if_empty:NF {\c_space_tl\Fld@color}
3428 )

```

```

3429 /DV(\Hy@escapestring{\Fld@default})
3430 /V(\Hy@escapestring{\Fld@value})
3431 ~\Fld@additionalactions
3432 \int_compare:nNnT { \Fld@maxlen}>{0}
3433 {
3434 /MaxLen~\Fld@maxlen
3435 }
3436 }
3437 \ExplSyntaxOff
3438
3439 \def\PDFForm@Submit{%
3440 /Subtype/Widget%
3441 \Fld@annotflags
3442 \Fld@pageobjref
3443 \Fld@annotnames
3444 /FT/Btn%
3445 \Fld@flags
3446 /H/P%
3447 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3448 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3449 \ifx\Fld@bordercolor\relax\else 1\fi
3450 \space
3451 \else
3452 /MK<<%
3453 \ifnum\Fld@rotation=\z@
3454 \else
3455 /R \Fld@rotation
3456 \fi
3457 \ifx\Fld@bordercolor\relax
3458 \else
3459 /BC[\Fld@bordercolor]%
3460 \fi
3461 >>%
3462 \fi
3463 /A<<%
3464 /S/SubmitForm%
3465 /F<<%
3466 /FS/URL%
3467 /F(\Hy@escapestring{\Form@action})%
3468 >>%
3469 \Fld@submitflags
3470 >>%
3471 \Fld@additionalactions
3472 }
3473 \ExplSyntaxOn
3474 \def\PDFForm@Reset{%
3475 /Subtype/Widget%
3476 \Fld@annotflags
3477 \Fld@pageobjref
3478 \Fld@annotnames
3479 /FT/Btn%
3480 \Fld@flags
3481 /H/P%
3482 /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%

```

```

3483     \ifcase0\ifnum\Fld@rotation=\z@   \else 1\fi
3484     \ifx\Fld@bordercolor\relax\else 1\fi
3485     \space
3486   \else
3487     /MK<<%
3488     \ifnum\Fld@rotation=\z@
3489     \else
3490     /R~\Fld@rotation
3491     \fi
3492     \ifx\Fld@bordercolor\relax
3493     \else
3494     /BC[\Fld@bordercolor]%
3495     \fi
3496     >>%
3497   \fi
3498   /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3499   /A<</S/ResetForm>>%
3500   \Fld@additionalactions
3501   }%
3502
3503
3504   %these patterns are used in hyperref checks.
3505   %it is unclear if they are really useful and if a backend support is
3506   %needed.
3507   \str_case:VnF \c_sys_backend_str
3508   {
3509     { pdfmode }
3510     {
3511       \def\HyPat@ObjRef
3512       {
3513         [0-9]*[1-9][0-9]*~0~R
3514       }
3515     }
3516     { dvipdfmx }
3517     {
3518       \def\HyPat@ObjRef
3519       {
3520         @[\~]+
3521       }
3522     }
3523     { xdvipdfmx }
3524     {
3525       \def\HyPat@ObjRef
3526       {
3527         @[\~]+
3528       }
3529     }
3530   }
3531   { %also set in hyperref sty, so probably not needed.
3532     \def\HyPat@ObjRef/{.+}
3533   }
3534
3535
3536   \ExplSyntaxOff

```

```

3537 % UF: removed Hy@writebookmark
3538 %   \Hy@currentbookmarklevel{0}
3539 %   \Hy@numberline
3540 %   \@writetorep
3541 %     counter{bookmark@seq@number}
3542 % removed \HyPsd@SanitizeForOutFile, not needed
3543 % removed \currentpdfbookmark, defined by bookmark,
3544 % should use \newcommand there
3545 % removed \subpdfbookmark, defined by bookmark,
3546 % should use \newcommand there
3547 % removed \belowpdfbookmark, defined by bookmark,
3548 % should use \newcommand there
3549 % removed \pdfbookmark, defined by bookmark,
3550 % \BOOKMARK
3551 % \@BOOKMARK
3552 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3553 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3554 %% removed \ReadBookmarks / unneeded with bookmark.
3555 %% removed \Hy@OutlineName
3556 %% removed \check@bm@number
3557 %% removed \calc@bm@number
3558
3559 \ifHy@implicit
3560 \else
3561   \expandafter\endinput
3562 \fi
3563 \newlength\Hy@SectionHShift
3564 \def\Hy@SectionAnchorHref#1{%
3565   \ifx\protect\@typeset@protect
3566     \Hy@@SectionAnchor{#1}%
3567   \fi
3568 }
3569 \DeclareRobustCommand*\Hy@@SectionAnchor}[1]{%
3570   \leavevmode
3571   \hbox to Opt{%
3572     \kern-\Hy@SectionHShift
3573     \Hy@raisedlink{%
3574       \hyper@anchorstart{#1}\hyper@anchorend
3575     }%
3576     \hss
3577   }%
3578 }
3579 \@ifundefined{hyper@nopatch@sectioning}
3580 {
3581   \let\H@old@ssect\@ssect
3582   \def\@ssect#1#2#3#4#5{%
3583     \Hy@MakeCurrentHrefAuto{section*}%
3584     \setlength{\Hy@SectionHShift}{#1}%
3585     \begingroup
3586     \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3587     \toks\tw@\expandafter{%
3588       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3589       #5%
3590     }%

```

```

3591 \edef\x{\endgroup
3592   \the\toks@{\the\toks\tw@}%
3593 } \x
3594 }
3595 \let\H@old@chapter\@chapter
3596 \def\@chapter#1{%
3597   \begingroup
3598   \let\@mkboth\@gobbletwo
3599   \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3600   \Hy@raisedlink{%
3601     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3602   }%
3603   \endgroup
3604   \H@old@chapter{#1}%
3605 }
3606 \@ifundefined{@chapter}{-}{%
3607   \let\Hy@org@chapter\@chapter
3608   \def\@chapter{%
3609     \def\Hy@next{%
3610       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3611       \Hy@raisedlink{%
3612         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3613       }%
3614     }%
3615     \ifnum\c@secnumdepth>\m@ne
3616       \ifundefined{if@mainmatter}%
3617         \iftrue{\csname if@mainmatter\endcsname}%
3618         \let\Hy@next\relax
3619       \fi
3620     \fi
3621     \Hy@next
3622     \Hy@org@chapter
3623   }%
3624 }
3625 \let\H@old@part\@part
3626 \begingroup\expandafter\expandafter\endgroup
3627 \expandafter\ifx\csname chapter\endcsname\relax
3628   \let\Hy@secnum@part\z@
3629 \else
3630   \let\Hy@secnum@part\m@ne
3631 \fi
3632 \def\@part{%
3633   \ifnum\Hy@secnum@part>\c@secnumdepth
3634     \phantomsection
3635     \fi
3636   \H@old@part
3637 }
3638 \let\H@old@spart\@spart
3639 \def\@spart#1{%
3640   \Hy@MakeCurrentHrefAuto{part*}%
3641   \Hy@raisedlink{%
3642     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3643   }%
3644   \H@old@spart{#1}%

```

```

3645 }
3646 \let\H@old@sect\@sect
3647 \def\@sect#1#2#3#4#5#6[#7]#8{%
3648   \ifnum #2>\c@secnumdepth
3649     \expandafter\@firstoftwo
3650   \else
3651     \expandafter\@secondoftwo
3652   \fi
3653   {%
3654     \Hy@MakeCurrentHrefAuto{section*}%
3655     \setlength{\Hy@SectionHShift}{#3}%
3656     \begingroup
3657     \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3658     \toks\tw@\expandafter{%
3659       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3660       #8%
3661     }%
3662     \edef\x{\endgroup
3663       \the\toks@{\the\toks\tw@}}%
3664     }\x
3665   }{%
3666     \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3667   }%
3668 }
3669 }{}
3670 \expandafter\def\csname Parent-4\endcsname{}
3671 \expandafter\def\csname Parent-3\endcsname{}
3672 \expandafter\def\csname Parent-2\endcsname{}
3673 \expandafter\def\csname Parent-1\endcsname{}
3674 \expandafter\def\csname Parent0\endcsname{}
3675 \expandafter\def\csname Parent1\endcsname{}
3676 \expandafter\def\csname Parent2\endcsname{}
3677 \expandafter\def\csname Parent3\endcsname{}
3678 \expandafter\def\csname Parent4\endcsname{}
3679 %%
3680 </package>
3681 <@@=hyp>
3682 <*colorscheme>
3683 % collected from https://tex.stackexchange.com/questions/525261/better-default-
3684 % colors-for-hyperref-links
3685 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3686 % linkcolor=
3687 %,filecolor=
3688 %,urlcolor=
3689 %,menucolor=
3690 %,runcolor=
3691 %,linkbordercolor=
3692 %,filebordercolor=
3693 %,urlbordercolor=
3694 %,menubordercolor=
3695 %,runbordercolor=
3696 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }
3697 {

```

```

3698 linkcolor      = [rgb]{1,0,0}, %red
3699 filecolor      = [rgb]{0,1,1}, %cyan
3700 urlcolor        = [rgb]{1,0,1}, %magenta
3701 menucolor       = [rgb]{1, 0, 0}, %red
3702 runcolor        = [rgb]{0,1,1}, %cyan
3703 %-----
3704 linkbordercolor = [rgb]{1, 0 ,0 },
3705 filebordercolor = [rgb]{0, .5, .5},
3706 urlbordercolor  = [rgb]{0, 1, 1},
3707 menubordercolor = [rgb]{1, 0, 0},
3708 runbordercolor  = [rgb]{0, .7, .7}
3709 }
3710
3711 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3712 {
3713 linkcolor      = [rgb]{0,0.2,0.6},
3714 filecolor      = [rgb]{0.8,0,0.8},
3715 urlcolor        = [rgb]{0.8,0,0.8},
3716 menucolor       = [rgb]{0,0.2,0.6},
3717 runcolor        = [rgb]{0.8,0,0.8},
3718 %----- %-----
3719 linkbordercolor = [rgb]{0,0.2,0.6},
3720 filebordercolor = [rgb]{0.8,0,0.8},
3721 urlbordercolor  = [rgb]{0.8,0,0.8},
3722 menubordercolor = [rgb]{0,0.2,0.6},
3723 runbordercolor  = [rgb]{0.8,0,0.8}
3724 }
3725
3726 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3727 { %two colors: intern/extern
3728 linkcolor      = [rgb]{0.79216, 0, 0.12549},
3729 filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3730 urlcolor        = [rgb]{0.01961, 0.44314, 0.6902},
3731 menucolor       = [rgb]{0.79216, 0, 0.12549 },
3732 runcolor        = [rgb]{0.01961, 0.44314, 0.6902 },
3733 %----- %-----
3734 linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3735 filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3736 urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3737 menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3738 runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3739 }
3740
3741 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3742 { %all darkgray
3743 linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3744 filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3745 urlcolor        = [rgb]{0.4 ,0.4 ,0.4 },
3746 menucolor       = [rgb]{0.4 ,0.4 ,0.4 },
3747 runcolor        = [rgb]{0.4 ,0.4 ,0.4 },
3748 %----- %-----
3749 linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3750 filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3751 urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },

```

```

3752     menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3753     runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3754 }
3755
3756 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3757 { %dvipsnam.def
3758     linkcolor        = [rgb]{0.06, 0.46, 1}, %NavyBlue
3759     filecolor        = [rgb]{1, 0, 0}, %Red
3760     urlcolor         = [rgb]{0.06, 0.46, 1}, %NavyBlue
3761     menucolor        = [rgb]{1, 0, 0}, %Red
3762     runcolor         = [rgb]{1, 0, 0}, %Red
3763 %----- %-----
3764     linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3765     filebordercolor = [rgb]{1, 0, 0}, %Red
3766     urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3767     menubordercolor = [rgb]{1, 0, 0}, %Red
3768     runbordercolor  = [rgb]{1, 0, 0} %Red
3769 }
3770
3771 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3772 { %dvipsnam.def
3773     linkcolor        = [rgb]{0.72, 0, 0}, %BrickRed
3774     filecolor        = [rgb]{0, 1, 0}, %Green
3775     urlcolor         = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3776     menucolor        = [rgb]{0.06, 0.46, 1}, %NavyBlue
3777     runcolor         = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3778 %----- %-----
3779     linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3780     filebordercolor = [rgb]{0, 1, 0}, %Green
3781     urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3782     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3783     runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3784 }
3785
3786
3787 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3788 {
3789     linkcolor        = [rgb]{0.50196, 0, 0.02353},
3790     filecolor        = [rgb]{0.07451, 0.09412, 0.46667},
3791     urlcolor         = [rgb]{0.54118, 0, 0.52941},
3792     menucolor        = [rgb]{0.44706, 0.45882, 0},
3793     runcolor         = [rgb]{0.07451, 0.46667, 0.46275},
3794 %----- %-----
3795     linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3796     filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3797     urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3798     menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3799     runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3800 }
3801
3802 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3803 {
3804     linkcolor        = [rgb]{0,0,0},
3805     filecolor        = [rgb]{0,0,0},

```



```
3806     urlcolor      = [rgb]{0,0,0},
3807     menucolor     = [rgb]{0,0,0},
3808     runcolor      = [rgb]{0,0,0},
3809     %-----      %-----
3810     linkbordercolor = [rgb]{0,0,0},
3811     filebordercolor = [rgb]{0,0,0},
3812     urlbordercolor  = [rgb]{0,0,0},
3813     menubordercolor = [rgb]{0,0,0},
3814     runbordercolor  = [rgb]{0,0,0}
3815   }
3816 </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, 796
<code>\\$</code>	271
<code>\%</code>	797
<code>\-</code>	2214
<code>\.</code>	558, 562, 564
@curropt commands:	
<code>\@curropt:</code>	2731
<code>\[</code>	2214
<code>\]</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, 947, 2733, 2798, 3135
<code>_</code>	556, 558, 562, 564
<code>\]</code>	2214
A	
<code>\A</code>	556, 2214
<code>\Acrobatmenu</code>	19, 166
<code>\addcontentsline</code>	14
<code>\AddToDocumentProperties</code>	381, 2209, 2312
<code>\AddToHook</code>	419, 436, 442
<code>\AddToHookNext</code>	191
<code>\advance</code>	2668, 2738, 2782, 2783
<code>allcolors</code> (hypersetup key)	<u>1075</u>
<code>\author</code>	2
B	
<code>\b</code>	560
<code>\begingroup</code>	202, 270, 368, 2574, 2698, 2727, 2739, 2773, 3157, 3585, 3597, 3626, 3656
<code>\belowpdfbookmark</code>	3547
<code>\bgroup</code>	267, 368
<code>\BOOKMARK</code>	3550
<code>bookmarkstype</code> (hypersetup key)	14
bool commands:	
<code>\bool_if:NTF</code>	260, 289, 348, 716, 740, 760, 780, 791, 827, 893, 958, 1036, 1046, 1299, 1310
<code>\bool_if:nTF</code>	908, 1984, 2009, 3292, 3399
<code>\bool_lazy_and:nnTF</code>	416
<code>\bool_lazy_or:nnTF</code>	1329, 1378
<code>\bool_new:N</code>	196, 197, 542, 546, 550
<code>\bool_set_true:N</code>	551
<code>bordercolormodel</code> (hypersetup key)	14, <u>1096</u>
box commands:	
<code>\box_dp:N</code>	653, 2881, 2932, 2967
<code>\box_ht:N</code>	652, 2880, 2931, 2966
<code>\box_new:N</code>	456, 553
<code>\box_set_dp:Nn</code>	2181
<code>\box_set_ht:Nn</code>	2188
<code>\box_use:N</code>	1316, 2883, 2941, 2970
<code>\box_use_drop:N</code>	1321
<code>\box_wd:N</code>	651, 2879, 2930, 2965
<code>\l_tmpa_box</code>	2877, 2879, 2880, 2881, 2883, 2916, 2930, 2931, 2932, 2941, 2963, 2965, 2966, 2967, 2970
C	
<code>\catcode</code>	271, 272
<code>\char</code>	2815, 3012, 3024
<code>\chardef</code>	151
<code>\cite</code>	34
clist commands:	
<code>\clist_item:nn</code>	2215, 2244
<code>\clist_map_function:nN</code>	117, 126
<code>\clist_map_inline:nn</code>	2376
<code>\clist_map_variable:nNn</code>	2730, 2796
color commands:	
<code>\color_export:nnN</code>	41, 427, 995, 1117
<code>\color_select:n</code>	1003, 1039, 1320, 2922, 2926
<code>\color_select:nn</code>	1009
<code>\color_set:nn</code>	2, 386, 1020
<code>\color_set:nnn</code>	2, 385, 1026
color names:	
<code>hyp/annot/file</code>	539
<code>hyp/annot/link</code>	539
<code>hyp/annot/menu</code>	539
<code>hyp/annot/run</code>	539
<code>hyp/annot/url</code>	539
<code>colorfile</code> (hypersetup key)	<u>1075</u>
<code>colorlink</code> (hypersetup key)	<u>1075</u>
<code>colorlinks</code> (hypersetup key)	<u>1052</u>
<code>colormenu</code> (hypersetup key)	<u>1075</u>
<code>colorrun</code> (hypersetup key)	<u>1075</u>
<code>colorscheme</code> (hypersetup key)	1, <u>1487</u>
<code>colorurl</code> (hypersetup key)	<u>1075</u>
cs commands:	
<code>\cs_generate_variant:Nn</code>	148, 149, 150, 383, 455, 594, 601, 824, 995, 1012, 1029, 2215
<code>\cs_gset:Npn</code>	1291
<code>\cs_gset_eq:NN</code>	402, 413
<code>\cs_if_exist_p:N</code>	417

\cs_new:Npn	110, 625	dim commands:	
\cs_new_protected:Npn	202, 298, 315, 332, 358, 378, 393, 404, 449, 570, 574, 578, 582, 595, 597, 602, 666, 674, 683, 687, 709, 714, 738, 758, 778, 825, 891, 956, 996, 1007, 1013, 1024, 1235, 2216, 2298	\dim_to_decimal_in_bp:n	12, 154
\cs_set:Nn	208, 250	\do	2731
\cs_set_eq:NN	154, 159, 160, 424, 633, 635, 796, 797, 1545	\documentclass	25
\cs_set_protected:Npn	152, 173, 612, 947	\DocumentMetadata	1, 3, 11, 15, 16, 19, 24, 33, 40
\csname	2745, 3127, 3138, 3617, 3627, 3670, 3671, 3672, 3673, 3674, 3675, 3676, 3677, 3678	draft (hypersetup key)	<u>1511</u>
\currentpdfbookmark	3543		
D			
\d	558, 562, 564		
debug (hypersetup key)	<u>1511</u>		
\DeclareRobustCommand	255, 283, 306, 323, 342, 371, 3569		
\def	12, 158, 368, 369, 371, 607, 2537, 2541, 2548, 2566, 2595, 2608, 2666, 2673, 2686, 2687, 2690, 2691, 2704, 2705, 2721, 2722, 2725, 2726, 2764, 2793, 2843, 2857, 2858, 2870, 2898, 2900, 2901, 2911, 2946, 2948, 2949, 2960, 2979, 2981, 2982, 2984, 2985, 2994, 3038, 3105, 3115, 3117, 3122, 3134, 3145, 3156, 3167, 3210, 3218, 3230, 3282, 3316, 3351, 3389, 3439, 3474, 3511, 3518, 3525, 3532, 3564, 3582, 3596, 3608, 3609, 3632, 3639, 3647, 3670, 3671, 3672, 3673, 3674, 3675, 3676, 3677, 3678		
\DefaultHeightofCheckBox	2985		
\DefaultHeightofChoiceMenu	2726		
\DefaultHeightofReset	2949		
\DefaultHeightofSubmit	2901		
\DefaultHeightofText	2695		
\DefaultHeightofTextMultiline	2693		
\DefaultOptionsofCheckBox	2988		
\DefaultOptionsofPushButton	2862		
\DefaultOptionsofReset	2953		
\DefaultOptionsofSubmit	2905		
\DefaultOptionsofText	2700		
\DefaultWidthofCheckBox	2984		
\DefaultWidthofChoiceMenu	2725		
\DefaultWidthofReset	2948		
\DefaultWidthofSubmit	2900		
\DefaultWidthofText	2690		
destlabel (hypersetup key)	14		
		\edef	2576, 2741, 2777, 2784, 3120, 3591, 3662
		\EdefEscapeName	2795, 2799
		\egroup	371
		\else	2573, 2694, 2748, 2752, 2755, 2772, 2780, 3108, 3113, 3130, 3136, 3148, 3152, 3169, 3173, 3177, 3181, 3185, 3189, 3193, 3197, 3201, 3205, 3221, 3225, 3325, 3326, 3327, 3329, 3332, 3336, 3340, 3346, 3363, 3367, 3371, 3377, 3381, 3448, 3449, 3451, 3454, 3458, 3483, 3484, 3486, 3489, 3493, 3560, 3629, 3650
		\endcsname	2759, 3104, 3127, 3138, 3617, 3627, 3670, 3671, 3672, 3673, 3674, 3675, 3676, 3677, 3678
		\endgroup	276, 279, 369, 371, 2577, 2718, 2741, 2787, 2790, 3164, 3591, 3603, 3626, 3662
		\endinput	3561
		exp commands:	
		\exp_args:Ne	834, 2326, 2568, 3212
		\exp_args:Nne	2341, 2521
		\exp_args:Nno	317, 334
		\exp_args:NnO	300, 360, 2284
		\exp_args:NNV	1969, 2000, 2245
		\exp_args:NnV	668, 677
		\exp_args:No	701, 705, 2860, 2903, 2951, 2986
		\exp_args:NV	649, 656
		\exp_not:N	947
		\exp_not:n	1426, 1460, 1552, 1607, 1628, 1665, 1685, 1708, 1752, 1785, 1810, 1861, 1886, 1915, 1964, 2041, 2069, 2096, 2116, 2362, 2466, 2494, 2504, 2515
		\expandafter	273, 2699, 2733, 2742, 2744, 2745, 2798, 3121, 3137, 3561, 3587, 3588, 3626, 3627, 3649, 3651, 3658, 3659, 3670, 3671, 3672, 3673, 3674, 3675, 3676, 3677, 3678
		\ExplSyntaxOff	15, 2605, 2658, 3102, 3315, 3437, 3536
		\ExplSyntaxOn	13, 16, 2606, 2663, 3209, 3350, 3473
		extension (hypersetup key)	14, <u>1529</u>

F	
<code>\fbox</code>	2834, 2922, 2926
<code>\fi</code>	12, 2562, 2589, 2592, 2601, 2696, 2704, 2705, 2736, 2754, 2757, 2758, 2764, 2767, 2776, 2778, 2785, 2788, 3110, 3132, 3142, 3144, 3150, 3154, 3161, 3162, 3171, 3175, 3179, 3183, 3187, 3191, 3195, 3199, 3203, 3207, 3223, 3227, 3325, 3326, 3327, 3334, 3338, 3342, 3344, 3346, 3365, 3369, 3373, 3377, 3384, 3448, 3449, 3456, 3460, 3462, 3483, 3484, 3491, 3495, 3497, 3562, 3567, 3619, 3620, 3631, 3635, 3652
file (hypersetup key)	10, 1560
file commands:	
<code>\file_input:n</code>	14
fileborderstyle (hypersetup key)	14, 1189
filecolor (hypersetup key)	1075
final (hypersetup key)	1511
<code>\fontencoding</code>	2811, 3008, 3020
<code>\fontfamily</code>	2810, 3007, 3019
<code>\fontseries</code>	2812, 3009, 3021
<code>\fontshape</code>	2813, 3010, 3022
fp commands:	
<code>\fp_eval:n</code>	2139, 2167, 2183
G	
<code>\gdef</code>	273, 274
<code>\GetDocumentProperties</code>	3
<code>\global</code>	2668, 3100
group commands:	
<code>\group_begin:</code>	258, 287, 310, 327, 346, 584, 722, 732, 746, 753, 785, 813, 832, 881, 898, 940, 963, 989, 1038, 1302, 1319, 2445, 2809, 2859, 2902, 2950, 2983, 3006, 3018
<code>\group_end:</code>	301, 318, 335, 363, 591, 728, 732, 765, 769, 810, 813, 878, 881, 938, 940, 987, 989, 1048, 1322, 1325, 2452, 2816, 2895, 2943, 2976, 3013, 3025, 3035
H	
<code>\hbox</code>	368, 3571
hbox commands:	
<code>\hbox_overlap_right:n</code>	1316
<code>\hbox_set:Nn</code>	2877, 2916, 2963
<code>\hbox_set:Nw</code>	1303
<code>\hbox_set_end:</code>	1312
<code>\hbox_set_to_wd:Nnn</code>	2165
hidefile (hypersetup key)	1463
hidelink (hypersetup key)	1463
hidelinks (hypersetup key)	1463
hidemenu (hypersetup key)	1463
hiderun (hypersetup key)	1463
hideurl (hypersetup key)	1463
hook commands:	
<code>\hook_gput_code:nnn</code>	395, 406, 1032, 1042, 1295, 1306
<code>\hook_new:n</code>	384, 569, 665
<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, 586, 671, 680, 725, 749
<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>	3576
hyp commands:	
<code>\l_hyp_annot_colorfile_bool</code> ..	8, 540
<code>\l_hyp_annot_colorlink_bool</code> ..	8, 540
<code>\l_hyp_annot_colormenu_bool</code> ..	8, 540
<code>\l_hyp_annot_colorrund_bool</code> ..	8, 540
<code>\l_hyp_annot_colorurl_bool</code> ..	8, 540
<code>\l_hyp_annot_ocgcolorfile_bool</code>	8, 544
<code>\l_hyp_annot_ocgcolorlink_bool</code>	8, 544
<code>\l_hyp_annot_ocgcolormenu_bool</code>	8, 544
<code>\l_hyp_annot_ocgcolorrund_bool</code>	8, 544
<code>\l_hyp_annot_ocgcolorurl_bool</code>	8, 544
<code>\l_hyp_current_dest_name_tl</code>	664, 723, 747
hyp internal commands:	
<code>\g__hyp_AcroForm_CoFields_prop</code> .	2532, 2550, 2619, 2621, 2636
<code>\g__hyp_AcroForm_Fields_prop</code> ...	2533, 2543, 2614
<code>\l__hyp_annot_GoTo_bool</code>	716, 740, 760
<code>\l__hyp_annot_GoTo_bool\l__-</code>	
<code>_hyp_annot_URI_bool\l__-</code>	
<code>hyp_annot_GoToR_bool\l__-</code>	
<code>hyp_annot_Named_bool\l__-</code>	
<code>hyp_annot_Launch_bool</code>	548
<code>\l__hyp_annot_GoToR_bool</code>	827
<code>\l__hyp_annot_Launch_bool</code>	893
<code>\l__hyp_annot_Named_bool</code>	958
<code>\c__hyp_annot_types_seq</code>	469, 540, 544, 1075, 1376, 1474
<code>\l__hyp_annot_URI_bool</code>	780
<code>\g__hyp_bordercolormodel_str</code> ...	429, 539, 1100, 1119
<code>__hyp_check_link_nesting:TF</code> ...	625, 633, 635, 719, 743, 762, 783, 830, 896, 961

_hyp_citebordercolor_hook_- init:	391, 404, 413	_hyp_href_url_aux:nn	296, 298
_hyp_citecolor_hook_init:	389, 393, 402	\l_hyp_href_url_encode_bool	196, 207, 245, 260, 289, 348
_hyp_clist_display:n	110, 117, 126	_hyp_href_url_format:	202, 208, 250, 361
_hyp_color_select:n	41, 996, 996, 1012	\l_hyp_href_url_ismap_bool	197, 221, 791
_hyp_color_select_aux:wn	996, 1000, 1007	\l_hyp_href_url_protocol_tl	198, 209, 248, 300, 362
_hyp_color_set:nn	41, 42, 389, 391, 426, 1013, 1013, 1029, 1080, 1116	_hyp_if_outer_link:	619
_hyp_color_set_aux:nwn	1013, 1017, 1024	_hyp_if_outer_link:TF	635
\c_hyp_colorscheme_daleif_prop	3711	\l_hyp_link_Contents_tl	504, 513, 515, 522, 524, 531, 533
\c_hyp_colorscheme_henryford_- prop	3802	_hyp_link_goto_begin:nw	687, 726, 750
\c_hyp_colorscheme_julian_prop	3726	_hyp_link_goto_end:	709, 727, 764
\c_hyp_colorscheme_phelype_prop	3787	\g_hyp_linknestlevel_int	618, 621, 718, 734, 742, 771, 782, 815, 829, 883, 895, 941, 960, 990
\c_hyp_colorscheme_szabolcsA_- prop	3756	g_hyp_linknestlevel_int	618
\c_hyp_colorscheme_szabolcsB_- prop	3771	\c_hyp_map_annot_hyp_prop	469
\c_hyp_colorscheme_tivv_prop	3741	\c_hyp_map_hyp_annot_prop	469, 1030, 1103, 1143, 1170, 1178, 1189, 1216, 1224, 1293, 1401, 1437, 1447, 1560
\l_hyp_dest_box	33, 553, 651, 652, 653, 2165, 2181, 2188	_hyp_ocg_init: 1235, 1235, 1291, 1301	
\l_hyp_dest_name_tmpa_tl	462, 690, 691, 697, 701, 703, 706, 850, 864	\l_hyp_optlang_regex 2213, 2214, 2246	
\l_hyp_dest_pdfremotestartview_- tl	488, 859, 1972, 1976	l_hyp_page/Trans	567
\g_hyp_dest_pdfstartpage_tl	488, 1983, 1985, 1992, 2010, 2017	_hyp_PageLabels_gpush: 602, 602, 615	
\g_hyp_dest_pdfstartview_tl	488, 1985, 1992, 2003, 2007, 2010, 2017	\l_hyp_para_tmpa_tl 465, 907, 910, 921	
\l_hyp_dest_pdfview_tl	538, 669, 678, 2134, 2137, 2144, 2147, 2148, 2149, 2150, 2151, 2152, 2157, 2161, 2197	\l_hyp_para_tmpa_tl\l_hyp_- text_tmpa_str\l_hyp_text_- tmpa_str	462
\c_hyp_dest_startview_regex	554, 1970, 2001	_hyp_property_record:nn	448, 449, 455, 2669
\c_hyp_dest_undefined_tl	468, 696, 697	_hyp_secondoftwowithopt:wnn	374, 375, 376, 377
_hyp_destination:nn	33, 638, 638, 669, 678	_hyp_setup_info_date_key:nn	2298, 2331, 2332
\l_hyp_filename_tmpa_tl	462, 833, 835, 840, 841, 846, 902, 903, 921	_hyp_setup_info_key:nn	2216, 2291, 2292, 2293, 2294, 2296, 2297
_hyp_href_pdf_aux:nn	312, 315	_hyp_store_metadata:nn	378, 383, 1579, 2223, 2266, 2311, 2326, 2335, 2351, 2419
\l_hyp_href_pdf_destination_tl	199, 210, 249, 317	_hyp_text_cleanup:N	574, 574, 588
\l_hyp_href_pdf_page_tl 200, 218, 858		\l_hyp_text_enc_dest_print_tl	491, 849
_hyp_href_run_aux:nn	329, 332	\l_hyp_text_enc_dest_tl	33, 491, 645, 702
\l_hyp_href_run_parameter_tl	201, 222, 334	\l_hyp_text_enc_file_print_tl	491, 901
_hyp_href_url_aux:n	355, 358	\l_hyp_text_enc_info_print_tl	491, 512, 521, 530, 599

\l__hyp_text_enc_para_print_tl	491 , 906	debug	1511
\l__hyp_text_enc_uri_print_tl	262,	destlabel	14
265, 291, 294, 350, 353, 491 , 788, 1571		draft	1511
__hyp_text_pdfstring:nnN		extension	14 , 1529
. 510, 519, 528, 582 , 582, 594, 596,		file	10 , 1560
599, 643, 700, 786, 847, 899, 904, 1571		fileborderstyle	14 , 1189
__hyp_text_pdfstring_info:nnN		filecolor	1075
597 , 597, 601, 2249, 2252, 2272, 2281		final	1511
__hyp_text_purify:nnN	570 , 570, 587	hidefile	1463
__hyp_text_string_from_unicode:nnN	578 , 578, 589	hidelink	1463
\g__hyp_text_tmpa_str	467, 590, 592	hidelinks	1463
\l__hyp_text_tmpa_str		hidemenu	1463
.	466, 587, 588, 589, 590	hiderun	1463
\l__hyp_tmpa_box 456 , 1303, 1316, 1321		hideurl	1463
\l__hyp_tmpa_int	456	hypertexnames	1529
\l__hyp_tmpa_seq		link	10 , 1560
.	456, 1970, 1972, 2001, 2003,	linkborder	14
2124, 2125, 2129, 2131, 2155, 2162,		linkborderstyle	14 , 1189
2163, 2164, 2173, 2175, 2185, 2190,		linkcolor	1075
2246, 2247, 2252, 2623, 2625, 2631		linkfileprefix	1529
\l__hyp_tmpa_str		linktoc	1529
.	456, 2249, 2252, 2254, 2256,	linktocpage	1529
2272, 2273, 2275, 2281, 2282, 2286		localanchorname	1529
\l__hyp_tmpa_tl		menu	10 , 1560
.	456, 646, 650, 658, 1120, 1124,	menuborder	14
1571, 1572, 1577, 1968, 1970, 1999,		menuborderstyle	14 , 1189
2001, 2131, 2132, 2139, 2222, 2223,		menucolor	1075
2224, 2244, 2249, 2265, 2266, 2272		naturalnames	1529
\l__hyp_tmpb_tl	456 , 2244, 2246	nested-links	10
\l__hyp_uri_tmpa_tl	462 , 789, 790	nesting	14
hyp/anchor	664	ocgcolorfile	1329
hyp/annot/file (color name)	539	ocgcolorlink	1329
hyp/annot/link (color name)	539	ocgcolorlinks	1329
hyp/annot/menu (color name)	539	ocgcolormenu	1329
hyp/annot/run (color name)	539	ocgcolorrn	1329
hyp/annot/url (color name)	539	ocgcolorurl	1329
hyp/text/pdfstring	569	pageanchor	1529
\hypercalcbp	12 , 19 , 154	pdfauthor	2213
\HyperDestNameFilter	13 , 644, 701	pdfborder	14
\hypersetup 2 , 3 , 5 , 9 , 10 , 14 , 19 , 69 , 102 , 173		pdfborderstyle	14 , 1189
\hypersetup keys:		pdfcreationdate	15 , 2298
allcolors	1075	pdfcreator	2213
bookmarkstype	14	pdfencoding	1501
bordercolormodel	14 , 1096	pdfinfo	2365
colorfile	1075	pdfkeywords	2213
colorlink	1075	pdflang	15 , 2202
colorlinks	1052	pdflinkmargin	15
colormenu	1075	pdfmetadate	15 , 2298
colorrn	1075	pdfmoddate	15 , 2298
colorscheme	1 , 1487	pdfproducer	2213
colorurl	1075	pdfremotestartview	12
		pdfstartview	12
		pdfsubject	2213
		pdftitle	2213

[pdftrapped](#) [2337](#)
[pdfversion](#) [1501](#)
[pdfview](#) [12](#), [2120](#)
[plainpages](#) [1529](#)
[run](#) [10](#), [1560](#)
[runborder](#) [14](#)
[runborderstyle](#) [14](#), [1189](#)
[runcolor](#) [1075](#)
[unicode](#) [1501](#)
[url](#) [10](#), [1560](#)
[urlborder](#) [14](#)
[urlborderstyle](#) [14](#), [1189](#)
[urlcolor](#) [1075](#)
[verbose](#) [1511](#)
hypertextnames (hypersetup key) [1529](#)

I

[\ifcase](#) [3325](#), [3448](#), [3483](#)
[\ifcsname](#) [3104](#)
[\ifdim](#) [2736](#), [2775](#), [2776](#)
[\ifnum](#) ... [3158](#), [3325](#), [3331](#), [3362](#), [3448](#),
[3453](#), [3483](#), [3488](#), [3615](#), [3633](#), [3648](#)
[\iftrue](#) [3617](#)
[\ifx](#) [2572](#), [2705](#), [2765](#),
[3135](#), [3147](#), [3151](#), [3168](#), [3172](#), [3176](#),
[3180](#), [3184](#), [3188](#), [3192](#), [3196](#), [3200](#),
[3204](#), [3220](#), [3224](#), [3326](#), [3327](#), [3335](#),
[3339](#), [3346](#), [3366](#), [3370](#), [3377](#), [3378](#),
[3449](#), [3457](#), [3484](#), [3492](#), [3565](#), [3627](#)
[\immediate](#) [2556](#), [2559](#)
int commands:
[\int_compare:nNnTF](#) ... [621](#), [2129](#),
[2155](#), [2835](#), [3249](#), [3301](#), [3410](#), [3432](#)
[\int_compare_p:nNn](#) [3294](#), [3401](#)
[\int_eval:n](#) [857](#)
[\int_gdecr:N](#) [734](#), [771](#), [815](#), [883](#), [941](#), [990](#)
[\int_gincr:N](#) [718](#), [742](#), [782](#), [829](#), [895](#), [960](#)
[\int_max:nn](#) [858](#)
[\int_new:N](#) [460](#), [618](#)
iow commands:
[\iow_newline:](#) [3053](#), [3059](#), [3072](#)

K

[\kern](#) [3572](#)
keys commands:
[\keys_define:nn](#) [179](#), [187](#), [205](#), [243](#),
[387](#), [629](#), [1052](#), [1077](#), [1084](#), [1096](#),
[1105](#), [1130](#), [1145](#), [1164](#), [1191](#), [1210](#),
[1333](#), [1347](#), [1359](#), [1382](#), [1388](#), [1403](#),
[1432](#), [1463](#), [1476](#), [1487](#), [1501](#), [1511](#),
[1516](#), [1529](#), [1541](#), [1562](#), [1567](#), [2120](#),
[2202](#), [2218](#), [2261](#), [2300](#), [2315](#), [2333](#),
[2337](#), [2365](#), [2417](#), [2423](#), [2436](#), [2456](#)
[\l_keys_key_str](#) [184](#), [2286](#), [2521](#)

[\keys_set:nn](#) [176](#),
[213](#), [232](#), [259](#), [288](#), [311](#), [328](#), [347](#),
[397](#), [408](#), [1493](#), [1500](#), [2342](#), [2369](#),
[2372](#), [2373](#), [2374](#), [2375](#), [2446](#), [2521](#)
[\keys_set_known:nn](#) [2524](#)
[\kvsetkeys](#) [175](#), [2610](#)

L

[\label](#) [9](#), [14](#)
[\LayoutCheckField](#) [2996](#)
[\LayoutChoiceField](#) [2768](#)
[\LayoutPushButtonField](#) [2872](#), [2889](#)
[\LayoutTextField](#) [2706](#)
[\leavevmode](#) [2707](#), [2802](#), [3570](#)
legacy commands:
[\legacy_if:nTF](#) [161](#), [434](#), [444](#),
[610](#), [2648](#), [2868](#), [2909](#), [2959](#), [2992](#), [3272](#)
[\let](#) [375](#),
[376](#), [377](#), [736](#), [805](#), [817](#), [874](#), [885](#),
[934](#), [943](#), [992](#), [2535](#), [2536](#), [2563](#),
[2659](#), [2660](#), [2661](#), [2688](#), [2689](#), [2723](#),
[2724](#), [2766](#), [3100](#), [3107](#), [3109](#), [3114](#),
[3131](#), [3165](#), [3581](#), [3595](#), [3598](#), [3607](#),
[3618](#), [3625](#), [3628](#), [3630](#), [3638](#), [3646](#)
[link](#) (hypersetup key) [10](#), [1560](#)
[linkborder](#) (hypersetup key) [14](#)
[linkborderstyle](#) (hypersetup key) [14](#), [1189](#)
[linkcolor](#) (hypersetup key) [1075](#)
[linkfileprefix](#) (hypersetup key) ... [1529](#)
[linktoc](#) (hypersetup key) [1529](#)
[linktocpage](#) (hypersetup key) [1529](#)
[localanchorname](#) (hypersetup key) ... [1529](#)
[\long](#) [12](#)

M

[\MakeButtonField](#)
..... [2877](#), [2892](#), [2916](#), [2963](#), [2974](#)
[\MakeChoiceField](#) [2852](#)
[\MakeFieldObject](#) [2526](#)
[\MakeRadioField](#) [2834](#)
[\MakeTextField](#) [2715](#)
[\mbox](#) [1313](#)
[menu](#) (hypersetup key) [10](#), [1560](#)
[menuborder](#) (hypersetup key) [14](#)
[menuborderstyle](#) (hypersetup key) [14](#), [1189](#)
[menucolor](#) (hypersetup key) [1075](#)
mode commands:
[\mode_if_horizontal:TF](#) [640](#), [662](#)
[\mode_leave_vertical:](#) .. [256](#), [285](#),
[308](#), [325](#), [344](#), [689](#), [799](#), [866](#), [923](#),
[966](#), [2846](#), [2874](#), [2891](#), [2913](#), [2955](#), [2998](#)
msg commands:
[\msg_error:nn](#) [418](#), [2888](#), [2973](#)
[\msg_info:mnn](#) [2230](#), [2235](#)

<code>\msg_line_context:</code>	75	<code>\pdf_object_ref_last:</code>	868, 2451
<code>\g_msg_module_name_prop</code>	17	<code>\pdf_object_unnamed_write:nn</code>	867, 2447
<code>\msg_new:nnn</code>	43, 50, 55, 59, 63, 70, 77, 84, 91, 98, 104, 111, 120, 130, 141	<code>\pdf_object_write:nnn</code>	1241, 1247, 1257, 1269, 3043, 3085, 3092
<code>\msg_new:nnnn</code>	18, 28	<code>\pdf_pageobject_ref:n</code>	150, 1992, 2017, 2677
<code>\msg_warning:nn</code>	163, 1507	<code>\pdf_string_from_unicode:nnN</code>	580
<code>\msg_warning:nnn</code>	172, 693, 984	<code>\pdf_version:</code>	1653, 1772, 1801, 1835, 1850, 1876, 1901, 1936, 1952, 2057, 2084
<code>\msg_warning:nnnn</code>	1351, 1392, 1605, 1649, 1683, 1706, 1768, 1797, 1831, 1846, 1859, 1872, 1897, 1932, 1948, 1975, 2006, 2039, 2053, 2080, 2114, 2196	<code>\pdf_version_compare:NnTF</code>	1643, 1734, 1756, 1792, 1825, 1840, 1866, 1891, 1919, 1942, 2047, 2074
<code>\msg_warning:nnnnn</code>	183, 1423, 1457, 1549, 1625, 1662, 1749, 1782, 1807, 1883, 1910, 1961, 2066, 2093, 2359, 2463, 2482, 2501, 2512	<code>\pdf_version_compare_p:Nn</code>	912, 1330, 1379
N			
naturalnames (hypersetup key)	1529	<code>\pdf_version_major:</code>	160, 1331, 1354, 1380, 1396
nested-links (hypersetup key)	10	<code>\pdf_version_minor:</code>	159, 1354, 1396
nesting (hypersetup key)	14	pdfannot commands:	
<code>\newcommand</code>	157, 3544, 3546, 3548	<code>\pdfannot_box:nnnn</code>	2710, 2819, 2849, 2878, 2929, 2964, 3028
<code>\newcount</code>	2664, 2842	<code>\pdfannot_box_ref_last:</code>	2570, 2586
<code>\NewDocumentCommand</code>	2526	<code>\pdfannot_dict_put:nnn</code>	514, 523, 532, 800, 868, 924, 969, 1121, 1156, 1180, 1202, 1226, 1408, 1439
<code>\NewExpandableDocumentCommand</code>	374	<code>\pdfannot_dict_remove:nn</code>	1111, 1151, 1172, 1197, 1218, 1416, 1449
<code>\newlength</code>	3563	<code>\pdfannot_link:nnn</code>	801, 869, 925, 970
<code>\noexpand</code>	2742, 2743, 2744, 2745, 3118	<code>\pdfannot_link_goto_begin:nw</code>	706
<code>\nolinkurl</code>	4	<code>\pdfannot_link_goto_end:</code>	711
O			
ocgcolorfile (hypersetup key)	1329	<code>\pdfannot_link_margin:n</code>	8, 1687
ocgcolorlink (hypersetup key)	1329	<code>\c_pdfannot_link_types_seq</code>	548
ocgcolorlinks (hypersetup key)	1329	pdfauthor (hypersetup key)	2213
ocgcolormenu (hypersetup key)	1329	<code>\pdfbookmark</code>	3549
ocgcolorrun (hypersetup key)	1329	pdfborder (hypersetup key)	14
ocgcolorurl (hypersetup key)	1329	<code>\pdfborderstyle</code> (hypersetup key)	14, 1189
P			
pageanchor (hypersetup key)	1529	pdfcreationdate (hypersetup key)	15, 2298
<code>\paperwidth</code>	4	pdfcreator (hypersetup key)	2213
<code>\PassOptionsToPackage</code>	446, 1521, 1526	<code>\pdfdest</code>	12
pdf commands:		pdfdict commands:	
<code>\pdf_bdcobject:nn</code>	1315, 1318	<code>\pdfdict_new:n</code>	567, 774, 820, 887, 952
<code>\pdf_destination:nn</code>	28, 148, 657	<code>\pdfdict_put:nnn</code>	230, 237, 568, 775, 776, 790, 793, 821, 822, 837, 843, 854, 864, 888, 889, 903, 918, 953, 954, 967, 1713, 1714, 1718, 1719, 2460, 2470, 2474, 2477, 2479, 2498, 2507, 2509
<code>\pdf_destination:nnnn</code>	650	<code>\pdfdict_remove:nn</code>	227, 915, 1723, 1724
<code>\pdf_emc:</code>	1317, 1323	<code>\pdfdict_use:n</code>	800, 867, 924, 930, 969, 974, 2449
<code>\pdf_name_from_unicode_e:n</code>	238, 837, 968	pdfencoding (hypersetup key)	1501
<code>\pdf_object_if_exist:nTF</code>	835	<code>\pdfescapestring</code>	3107
<code>\pdf_object_new:n</code>	1237, 1238, 1239, 1240, 3040, 3041, 3042		
<code>\pdf_object_ref:n</code>	149, 846, 1243, 1245, 1267, 1268, 1271, 1275, 1280, 1285, 1290, 2641, 2643, 3098		

pdffile commands:	prg commands:
\pdffile_embed_file:nnn 824, 838	\prg_do_nothing: 402, 413
pdfinfo (hypersetup key) 2365	\prg_generate_conditional_-
pdfkeywords (hypersetup key) 2213	variant:Nnn 2672
pdflang (hypersetup key) 15, 2202	\prg_new_conditional:Npnn 619
pdflinkmargin (hypersetup key) 15	\prg_return_false: 622
pdfmanagement commands:	\prg_return_true: 623
\pdfmanagement_add:nn 1696	prop commands:
\pdfmanagement_add:nnn	\prop_const_from_keyval:Nn
. 604, 1267, 1268, 1290, 1577, 471, 479, 3696, 3711,
1597, 1613, 1617, 1637, 1645, 1674,	3726, 3741, 3756, 3771, 3787, 3802
1729, 1739, 1763, 1777, 1789, 1794,	\prop_gput:Nnn 17, 2543, 2550
1819, 1827, 1842, 1868, 1893, 1927,	\prop_if_empty:NTF 2619
1944, 1990, 2015, 2029, 2049, 2076,	\prop_item:Nn 2636
2105, 2208, 2231, 2236, 2256, 2275,	\prop_map_inline:Nn
2285, 2309, 2324, 2346, 2432, 2451, 1030, 1103, 1143, 1170,
2616, 2633, 2640, 2642, 2644, 2650	1178, 1189, 1216, 1224, 1293, 1401,
\pdfmanagement_if_active_p: 417	1437, 1447, 1491, 1560, 2614, 2621
\pdfmanagement_remove:nn	\prop_new:N 2532, 2533
. 1574, 1593, 1601, 1621,	property commands:
1633, 1658, 1670, 1678, 1692, 1701,	\property_if_recorded:nn 2672
1745, 1760, 1779, 1816, 1855, 1880,	\property_if_recorded:nnTF 2675
1906, 1923, 1957, 1987, 2012, 2025,	\property_record:nn 452
2034, 2062, 2089, 2101, 2110, 2240,	\property_ref:nn 2679
2269, 2306, 2321, 2429, 2442, 2654	\property_ref_undefined_warn:nn 2670
\pdfmanagement_show:n 2617	\protect 736,
pdfmeta commands:	805, 817, 874, 885, 934, 943, 992, 3565
\pdfmeta_standard_verify:nnTF	\providecommand
. 964, 2865, 2956 155, 156, 2557, 2560, 2720, 3103
\pdfmeta_standard_verify:nTF	\ProvidesFile 3, 7
. 2646, 3214	
pdfmetadate (hypersetup key) 15, 2298	R
pdfmoddate (hypersetup key) 15, 2298	\ReadBookmarks 3554
pdfproducer (hypersetup key) 2213	\refstepcounter 13
pdfremotestartview (hypersetup key) 12	regex commands:
pdfstartview (hypersetup key) 12	\regex_const:Nn 554
\pdfstringdef 9, 10, 23, 29, 3163	\regex_extract_once:NnN 2246
\pdfstringdefDisableCommands	\regex_extract_once:NnNTF 1970, 2001
. 375, 376, 377	\regex_new:N 2213
pdfsubject (hypersetup key) 2213	\regex_set:Nn 2214
pdftitle (hypersetup key) 2213	\relax 736, 805, 817,
pdftrapped (hypersetup key) 2337	874, 885, 934, 943, 992, 2723, 2724,
pdfversion (hypersetup key) 1501	2765, 3100, 3147, 3151, 3220, 3224,
pdfview (hypersetup key) 12, 2120	3326, 3327, 3335, 3339, 3366, 3370,
pdfxform commands:	3449, 3457, 3484, 3492, 3618, 3627
\pdfxform_if_exist:nTF	\RemoveFromHook 1583
. 2805, 2917, 3001	\RenewDocumentCommand 166
\pdfxform_new:nnn	\RequirePackage 11, 26, 438, 3552
. 2528, 2807, 2920, 2924, 3003, 3015	run (hypersetup key) 10, 1560
\pdfxform_ref:n	runborder (hypersetup key) 14
. 2829, 2830, 2936, 2937, 3244, 3245	runborderstyle (hypersetup key) 14, 1189
\phantom 3024	runcolor (hypersetup key) 1075
\phantomsection 13, 20, 3634	
plainpages (hypersetup key) 1529	S
	\selectfont 2814, 3011, 3023

seq commands:	
\seq_const_from_clist:Nn	469
\seq_count:N	2129, 2155
\seq_get_right:NN	2131
\seq_if_empty:NTF	2247
\seq_item:Nn	1972, 2003, 2125, 2173, 2175, 2185, 2190, 2252
\seq_map_inline:Nn	540, 544, 548, 1075, 1376, 1474, 2631
\seq_new:N	459
\seq_put_right:Nn	2162, 2163, 2164, 2623
\seq_set_split:Nnn	2124
\seq_sort:Nn	2625
\setbox	368
\SetKeys	34
\setlength	3584, 3655
\setpdflinkmargin	8
\settowidth	2735
socket commands:	
\socket_assign_plug:nm	535, 536, 537
\socket_new:nm	505, 506, 507
\socket_new_plug:nmm	508, 517, 526
\socket_use:nm	795, 851
\socket_use:nmm	724, 748
sort commands:	
\sort_return_same:	2629
\sort_return_swapped:	2628
\space	2801, 3328, 3345, 3346, 3376, 3377, 3450, 3482, 3485
\spacefactor	640, 662
\stockwidth	4
str commands:	
\c_colon_str	947
\c_hash_str	796
\c_percent_str	797
\str_case:nmTF	2226, 3507
\str_case_e:nmTF	2125
\str_compare:nNmTF	2627
\str_gset:Nn	1100
\str_gset_eq:NN	590
\str_head:n	2348, 2353
\str_if_eq:nmTF	647, 2254, 2273, 2282, 3212
\str_if_eq_p:nm	910, 1331, 1380
\str_lowercase:n	2125, 2266, 2326, 2349, 2354
\str_new:N	461, 466, 467, 539
\str_set:Nn	572
\str_set_convert:Nmm	9
\str_set_eq:NN	592
\str_tail:n	2349, 2354
\str_uppercase:n	2342, 2348, 2353
\string	2557, 2560, 2581, 2599, 3123, 3127, 3138
\subpdfbookmark	3545
sys commands:	
\c_sys_backend_str	3507
T	
TeX and L ^A T _E X 2 _ε commands:	
\@@Listbox	2786, 2843
\@@Radio	2771, 2793
\@@writetorep	3540
\@BOOKMARK	3551
\@CheckBox	2979
\@ChoiceMenu	2721
\@Form	2608
\@PushButton	2857
\@Reset	2946
\@Submit	2898
\@TextField	2686
\@bookmarksopenlevel	1588
\@bsphack	451
\@chapter	3607, 3608
\@currDisplay	2735, 2801
\@currValue	2799, 2829
\@currentHref	3588, 3601, 3612, 3642, 3659
\@curropt	2720, 2730, 2733, 2796, 2798
\@empty	424, 2535, 2536, 2572, 2659, 2660, 2661, 2688, 2689, 2705, 3168, 3172, 3176, 3180, 3184, 3188, 3192, 3196, 3200, 3204, 3346, 3377, 3378
\@endForm	2659
\@esphack	453
\@firstofone	2563, 3109, 3114, 3131
\@firstoftwo	3649
\@for	2731
\@gobbletwo	3598
\@hyper@launch	40, 947
\@ifnextchar	267
\@ifundefined	3123, 3579, 3606, 3616
\@mainaux	2556, 2559, 2579, 2598
\@mkboth	3598
\@ne	2668
\@nil	3121, 3134, 3140
\@part	3625, 3632
\@pdfauthor	23
\@pdfborder	155
\@pdfborderstyle	156
\@savsf	640, 662
\@schapter	3595, 3596
\@secondoftwo	3651
\@sect	3646, 3647
\@spart	3638, 3639
\@ssect	3581, 3582

\@tempdima	. 2729, 2736, 2738, 2775, 2776, 2777, 2781, 2782, 2783, 2784	\Fld@onexit@code 3204, 3206
\@tempdimb 2735, 2736	\Fld@onfocus@code 3184, 3186
\@typeset@protect 3565	\Fld@onmousedown@code 3192, 3194
\BKM@color 424, 430	\Fld@onmouseup@code 3196, 3198
\c@secnumdepth 3615, 3633, 3648	\Fld@pageobjref 2661, 2673, 3103, 3234, 3286, 3319, 3355, 3393, 3442, 3477
\calc@bm@number 3557	\Fld@radiosymbol 3374
\check@bm@number 3556	\Fld@rotation 3249, 3251, 3294, 3301, 3303, 3325, 3331, 3333, 3362, 3364, 3401, 3410, 3412, 3448, 3453, 3455, 3483, 3488, 3490
\define@key 421	\Fld@submitflags 3469
\Fld@additionalactions	3210, 3279, 3312, 3348, 3385, 3431, 3471, 3500	\Fld@validate@code 3176, 3178
\Fld@align 3238, 3323, 3397	\Fld@value 2689, 2705, 2724, 2765, 2766, 3430
\Fld@altname	... 3147, 3149, 3220, 3222	\Fld@width 2690, 2704, 2711, 2715, 2725, 2764, 2775, 2777, 2820, 2834, 2850, 2852, 2870, 2900, 2911, 2948, 2960, 2984, 2994, 3029
\Fld@annotflags 3233, 3285, 3318, 3354, 3392, 3441, 3476	\Fld@X@additionalactions 3167, 3212, 3215
\Fld@annotnames 3218, 3235, 3287, 3320, 3356, 3394, 3443, 3478	\Form@action 3467
\Fld@bcolor 3257, 3259, 3339, 3341, 3370, 3372, 3405, 3418, 3420	\H@old@part 3625, 3636
\fld@bcolor 3327	\H@old@schapter 3595, 3604
\Fld@bordercolor 3253, 3255, 3296, 3305, 3307, 3326, 3335, 3337, 3366, 3368, 3403, 3414, 3416, 3449, 3457, 3459, 3484, 3492, 3494	\H@old@sect 3646, 3657, 3666
\Fld@borderstyle 3239, 3291, 3324, 3360, 3398, 3447, 3498	\H@old@spart 3638, 3644
\Fld@borderwidth	2782, 2783, 3239, 3291, 3324, 3360, 3398, 3447, 3498	\H@old@ssect 3581, 3586
\Fld@calculate@code	. 2572, 3180, 3182	\href@ 267, 273
\Fld@calculate@sortkey 2583	\href@split 273, 274
\Fld@cbsymbol 3261	\Hy@@escapeform 3121, 3134, 3140
\Fld@charsize 2781, 3265, 3345, 3376, 3426, 3482	\Hy@@SectionAnchor 3566, 3569
\Fld@checkequals 2733, 2798	\Hy@abspage 614
\Fld@choices 3347	\Hy@activeanchorfalse 685
\Fld@color	3266, 3268, 3346, 3377, 3427	\Hy@activeanchortrue 676
\Fld@default	2688, 2705, 2723, 2766, 2795, 2982, 3378, 3382, 3383, 3429	\Hy@AtBeginDocument 2537, 2553
\Fld@flags 3237, 3289, 3322, 3358, 3396, 3445, 3480	\Hy@bookmarkstype 1589
\Fld@format@code 3172, 3174	\Hy@chapapp 3599, 3610
\Fld@height 2691, 2712, 2715, 2726, 2784, 2821, 2834, 2850, 2852, 2901, 2949, 2985, 3030	\Hy@colorlink 35
\Fld@keystroke@code 3168, 3170	\Hy@currentbookmarklevel 3538
\Fld@listcount	. 2794, 2800, 2835, 2842	\Hy@DisableOption 165
\Fld@mappingname	3151, 3153, 3224, 3226	\Hy@drafttrue 1520
\Fld@maxlen 3432, 3434	\Hy@escapeform 2709, 2804, 2848, 2876, 2915, 2962, 3000, 3105, 3112, 3115
\Fld@menulength 2728, 2734, 2781	\Hy@escapestring 3107, 3109, 3114, 3117, 3118, 3121, 3122, 3131, 3134, 3170, 3174, 3178, 3182, 3186, 3190, 3194, 3198, 3202, 3206, 3261, 3311, 3374, 3429, 3430, 3467
\Fld@name 2687, 2722, 2858, 2981, 3146, 3219	\Hy@finaltrue 1525
\Fld@onblur@code 3188, 3190	\Hy@FormObjects 2613, 3038, 3100
\Fld@onclick@code 3311	\Hy@gttemp 3163, 3165
\Fld@onenter@code 3200, 3202	\Hy@href 267

\Hy@href@nextactionraw	231	\HyField@FlagsPushButton	2867, 2907, 2958
\Hy@href@page	219	\HyField@FlagsRadioButton	2770
\Hy@linkfileprefix	1534	\HyField@FlagsSubmit	2908
\Hy@linktoc	1545	\HyField@FlagsText	2703
\Hy@MakeCurrentHref	192	\HyField@PDFChoices	2845
\Hy@MakeCurrentHrefAuto	3583, 3599, 3610, 3640, 3654	\HyField@SetKeys	2699, 2740, 2743, 2762, 2860, 2903, 2951, 2986
\Hy@next	3609, 3618, 3621	\HyPat@ObjRef	3511, 3518, 3525, 3532
\Hy@numberline	158, 3539	\hyper@link	275
\Hy@org@chapter	3607, 3622	\hyper@anchor	664
\Hy@OutlineName	3555	\hyper@anchorend	664, 3574, 3601, 3612, 3642
\Hy@OutlineRerunCheck	3553	\hyper@anchorstart	664, 3574, 3601, 3612, 3642
\Hy@pdfmajorversion	160	\hyper@link	35, 714
\Hy@pdfminorversion	159	\hyper@linkend	35, 758
\Hy@pdfstringtrue	30, 585	\hyper@linkfile	317, 825
\Hy@pdfversion	3158	\hyper@linklaunch	39, 334, 891, 950
\Hy@pstringdef	595, 3138	\hyper@linknamed	40, 168, 956
\Hy@PutCatalog	602	\hyper@linkstart	35, 738
\Hy@raisedlink	3573, 3600, 3611, 3641	\hyper@linkurl	300, 361, 778
\Hy@RestoreLastskip	661	\hyper@normalise	267, 296, 312, 329, 355, 368
\Hy@ReturnAfterFi	12, 3139	\HyPL@Labels	604, 614
\Hy@safe@activestru	798, 2575	\HyPL@StorePageLabel	602
\Hy@SaveLastskip	641	\HyPsd@SanitizeForOutFile	3542
\Hy@secnum@part	3628, 3630, 3633	\if@filesw	2555, 2578, 2597
\Hy@SectionAnchorHref	3564, 3588, 3659	\if@fld@combo	2749, 2779
\Hy@SectionHShift	3563, 3572, 3584, 3655	\if@fld@hidden	2704, 2764
\Hy@StepCount	2734, 2800	\if@fld@multiline	2692
\Hy@temp	2576, 2591, 3120, 3121	\if@fld@popdown	2750
\Hy@unicodefalse	3160	\if@fld@radio	2746, 2769
\Hy@VerboseAnchor	642	\if@hy@implicit	3559
\Hy@VerboseLinkStart	721, 745	\if@hy@pdfescapeform	3106, 3116
\Hy@VerboseLinkStop	729, 766, 808, 876, 936, 980	\if@hy@unicode	3159
\Hy@VersionChecked	151	\kv@set@family@handler	171
\Hy@WrapperDef	638	\m@ne	3615, 3630
\Hy@xspace@end	726, 807, 875, 935, 979	\OBJ@OCG@view	157
\HyAnn@AbsPageLabel	2660, 2666, 2708, 2803, 2847, 2875, 2914, 2961, 2999	\p@	2738
\HyAnn@Count	2664, 2665, 2668, 2669, 2670, 2675, 2679	\pdf@ifdraftmode	2611
\HyField@@AddToFields	2568, 2595	\PDF@SetupDoc	152
\HyField@AddToFields	2566, 2716, 2836, 2853, 2884, 2940, 2969, 3033	\PDFForm@@@Name	3146, 3149, 3153, 3156
\HyField@afields	2535	\PDFForm@Check	3000, 3032, 3230
\HyField@AfterAuxOpen	2537, 2563, 2596	\PDFForm@List	2848, 2851, 3316
\HyField@AuxAddToCoFields	2548, 2560, 2581	\PDFForm@Name	2702, 2763, 2864, 2990, 3145
\HyField@AuxAddToFields	2541, 2557, 2599	\PDFForm@Push	2876, 2882, 3282
\HyField@cofields	2536	\PDFForm@Radio	2804, 2824, 3351
\HyField@FlagsCheckBox	2991	\PDFForm@Reset	2962, 2968, 3474
\HyField@FlagsChoice	2774	\PDFForm@Submit	2915, 2934, 3439
		\PDFForm@Text	2709, 2714, 3389
		\protected@edef	690
		\strip@pt	3265, 3345, 3376, 3426, 3482

<code>\toks@</code>	3586, 3592, 3657, 3663	
<code>\tw@</code>	3587, 3592, 3658, 3663	
<code>\url@</code>	368, 371	
<code>\Url@def</code>	369	
<code>\Url@HyperHook</code>	369	
<code>\XR@ext</code>	1531	
<code>\z@</code>	368, 2665, 2729, 3325, 3331, 3362, 3448, 3453, 3483, 3488, 3628	
<code>\texorpdfstring</code>	30	
text commands:		
<code>\text_expand:n</code>	66, 833, 2222, 2265	
<code>\text_purify:n</code>	9, 572	
<code>\textsf</code>	2922, 2926	
<code>\the</code>	614, 2669, 2670, 2675, 2679, 2777, 2781, 2784, 3592, 3663	
<code>\theH...</code>	13	
<code>\ThisShouldNotHappen</code>	3125	
<code>\title</code>	2	
tl commands:		
<code>\c_space_tl</code>	158, 1244, 1275, 1280, 1285, 2829, 2830, 2837, 3049, 3051, 3061, 3063, 3065, 3068, 3070, 3265, 3268, 3398, 3426, 3427	
<code>\tl_const:Nn</code>	468	
<code>\tl_gput_right:Nn</code>	614	
<code>\tl_gset:Nn</code>	1983, 2003, 2007	
<code>\tl_if_blank:nTF</code>	423, 852, 2267, 2304, 2319, 2427, 2440	
<code>\tl_if_empty:NnTF</code>	691, 1572, 2224, 3253, 3257, 3266, 3427	
<code>\tl_if_empty:nTF</code>	225, 1109, 1149, 1168, 1195, 1214, 1758, 1814, 1921, 2206	
<code>\tl_if_empty_p:N</code>	1985, 2010	
<code>\tl_if_eq:NnTF</code>	2132	
<code>\tl_if_exist:NnTF</code>	3305, 3414, 3418	
<code>\tl_if_exist_p:N</code>	3296, 3403, 3405	
<code>\tl_if_head_eq_charcode:nNTF</code>	998, 1015	
<code>\tl_new:N</code>	198, 199, 200, 201, 457, 458, 462, 463, 464, 465, 488, 489, 490, 491, 492, 493, 494, 495, 496, 504, 538, 664	
<code>\tl_set:Nn</code>	218, 219, 231, 262, 265, 291, 294, 350, 353, 380, 498, 499, 500, 501, 502, 503, 670, 679, 723, 747, 833, 1968, 1972, 1976, 1999, 2134, 2137, 2144, 2147, 2148, 2149, 2150, 2151, 2152, 2157, 2161, 2197, 2222, 2244, 2265	
<code>\tl_set_eq:NN</code>	697	
<code>\tl_to_str:N</code>	835, 841, 846	
<code>\tl_to_str:n</code>	24, 25, 34, 38	
token commands:		
<code>\token_to_str:N</code>	33	
<code>\toks</code>	3587, 3592, 3658, 3663	
U		
unicode (hypersetup key)	1501	
<code>\Url</code>	4, 5, 202	
url (hypersetup key)	10, 1560	
<code>\url</code>	4-6, 23, 342, 368	
urlborder (hypersetup key)	14	
urlborderstyle (hypersetup key)	14, 1189	
urllcolor (hypersetup key)	1075	
<code>\urldef</code>	5, 23, 368, 370	
<code>\Urlfont</code>	6	
<code>\UrlSpecial</code>	4	
use commands:		
<code>\use:n</code>	945	
<code>\use_i:nn</code>	627, 633	
V		
verbose (hypersetup key)	1511	
W		
<code>\write</code>	2556, 2559, 2579, 2598	
X		
<code>\x</code>	2741, 2761, 3591, 3593, 3662, 3664	