

The hyperref-generic module

A generic driver for hyperref

The L^AT_EX Project*

Version 0.95d, released 2021-05-14

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

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

The driver *requires* the new PDF management code, so documents wanting to use it should start like this

```
\RequirePackage{pdfmanagement-testphase} %loads the code
\DeclareDocumentMetadata % activates it
{
  %% options
  %% e.g. pdf version, backend:
  % pdfversion=1.7,
  % backend = dvipdfmx
}
```

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

1 Avoiding transition problems

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

- Package options are processed at the end of the driver, Class options are ignored. But not every option already works as package options, in some cases `hyperref` interferes. So it is recommended for most options —with the exception of a few mentioned below in section 9—to set them in `\hypersetup`, not as package option.

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

- This driver uses `l3color` 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`. This driver uses `l3color` for the colors, some special color models of `xcolor` won't work.
- Load a color package or `graphicx` to get the right page sizes.
- Report problems! Only known problem can be resolved.

2 Bookmarks / outlines

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

3 “Metadata”

“Metadata”, informations about the document, are stored in a PDF in two places: The `/Info` dictionary and the XMP-metadata. `hyperref` only handles the `/Info` dictionary. The XMP-metadata can be added with 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 an number of internal `hyperref` commands—which is a problem if the internal commands change (as happens with this new driver)
- `hyperref` (and also `hyperxmp`) tries to deduce some datas from document commands like `\title` or `\author`—something that worked reasonably well when only some standard classes with well-known definitions of these command existed, but gets problematic with classes and packages which define more powerful commands knowing a variety of optional arguments to set authors and affiliations and title information.

To resolve some of this problem the driver will

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

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

- 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 `\DeclareDocumentMetadata` should be used:

```
\DeclareDocumentMetadata{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'`.

One should be aware that `hyperxmp` will sometimes overwrites such settings.

5 PDF page size (mediabox)

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

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

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

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

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

6 Commands to create “external” references

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

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

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

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

(Expandable) commands are expanded and special chars can also be input by commands but beside this no conversion is done: for all input `hyperref` basically assumes that

the input is already a valid percent encoded url or a valid file name. `hyperref` also doesn't extend or add protocols.

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

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

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

6.1 Special problem: links to files

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

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

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

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

6.2 Splits

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

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

For this reason three new commands have been create:

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

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

6.3 Options

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

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

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

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

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

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

Some main changes are

- The default colors have been changed.
- `citecolor` is no longer set with `allcolors` and it is not part of the color scheme, it is only supported for compability.
- 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 `cmk`. `rgb` is the default. It does *not* change the model of text colors. Be aware that while the PDF format allows `cmk` (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 `original` (the colors as `hyperref` uses normally), `phelype`, `daleif`, `szabolcsA`, `szabolcsB`, `tivv`, `julian`, `henryford`.

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

The default is `phelype`.

7.3 Public interfaces

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

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

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

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

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

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

7.4 Changed behaviour

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

Color keys accept the following input syntax:

```
model based      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.
- `destlabels` (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 L^AT_EX internals)
- **nesting** unneeded key, see comment below in 14. At some time it will be either removed or extended (if some use can be found).
- **pagebackref**
- **pdfpagelabels** (set PDF page labels)
- **psdextra** this loads some extra definitions used by `\pdfstringdef`. The new driver uses `\pdfstringdef` only for the bookmarks, for other strings it is not relevant.

Options that can be without problems set as package options are

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

Ignored options:

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

10 Disabling links

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

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

url (*setup key*)

file (*setup key*)

run (*setup key*)

menu (*setup key*)

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 `\DeclareDocumentMetadata`

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 `\DeclareDocumentMetadata`:

```
\DeclareDocumentMetadata{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 pdf-tex and luatex the automatic calculation of the coordinates from the encompassing box. With dvips and (x)dvipdfmx it will fall back to `Fit`. `pdfstartview` and `pdfremotestartview` will fallback to `Fit` too.

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

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

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 is set earlier.

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

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

will create

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

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

`nesting` (*setup key*)

`pdfborder` (*setup key*)

`linkborder` (*setup key*)

`urlborder` (*setup key*)

`runborder` (*setup key*)

`menuborder` (*setup key*)

This key set accept as value three numbers or three numbers and an array describing a dash pattern, examples are `0 0 1` or `0 0 1 [3 2]`. The first two numbers should according to the reference set round corners, but PDF viewer seem to ignore it. The third number is the line width of the border. Settings done with `pdfborderstyle` should take precedence.

`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*)

| Key | Values | comment / example |
|--------------------|---------------------------------------|--|
| <code>/Type</code> | <code>/Border</code> | optional |
| <code>/W</code> | <code><number></code> | Width of border line |
| <code>/S</code> | <code>/S</code> | solid (default) |
| | <code>/D</code> | dash pattern can be set with <code>/D</code> |
| | <code>/B</code> | beveled |
| | <code>/I</code> | inset |
| | <code>/U</code> | underline |
| <code>/D</code> | <code><array of numbers></code> | dash pattern (lines/gaps) (default [3]) |

`pdfcreationdate` (*setup key*)

Setting these keys is normally not needed. If they are used the values are stored directly in the Info dictionary for `/Creationdate` and `/ModDate`. The values are converted to strings but not processed further, so they should have the correct PDF format without the enclosing parentheses, e.g. `D:20200202111111+01'00'`.

`pdfmoddate` (*setup key*)

`pdflinkmargin` (*setup key*)

As described in the `hyperref` manual the behaviour differs between the backends: with dvips it is possible to change links locally, pdf_latex and lua_latex work by page, with dvipdfmx the setting is global (and has to be done in the preamble).

`pdflang` (*setup key*)

The key will work, but it is recommended to the set the language in `\DeclareDocumentMetadata` instead.

File I

hyperref-generic driver implementation

```
1 <*package>
2 <@@=hyp>
3 \ProvidesFile{hgeneric-testphase.def}[2021-05-14 v0.95d %
4   generic Hyperref driver for the LaTeX PDF management testphase bundle]
5
6 \RequirePackage{etoolbox} %why?
7 \ExplSyntaxOn
```

```

8 \file_input:n {hyperref-colorschemes.def}
9 \ExplSyntaxOff

```

1 messages

Redirect the message name:

```

10 \ExplSyntaxOn
11 \prop_gput:Nnn \g_msg_module_name_prop { hyp }{ hyperref }

```

At first a message for the testing of the resource management

```

12 \msg_new:nnnn
13 { hyp }
14 { missing-resource-management }
15 { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
16 {
17   Activate~it~with \\
18   \tl_to_str:n{\RequirePackage{pdfmanagement-testphase}}\\
19   \tl_to_str:n{\DeclareDocumentMetadata{<options>}}\\
20   before~\tl_to_str:n{\documentclass}
21 }

```

The pdfversion should be set in \DeclareDocumentMetadata

```

22 \msg_new:nnnn
23 { hyp }
24 { pdfversion-disabled }
25 {
26   This~hyperref~driver~ignores~the~pdfversion~key!\\
27   Set~the~pdfversion~in~\token_to_str:N \DeclareDocumentMetadata
28 }
29 {
30   For example:\\
31   \tl_to_str:n
32   {
33     \DeclareDocumentMetadata { pdfversion=1.7 }
34   }
35 }

```

A generic message for ignored keys.

```

36 \msg_new:nnn
37 { hyp }
38 { key-dropped }
39 {
40   This~hyperref~driver~ignores~the~key~#1!\\
41   Please~check~the~documentation.
42 }

```

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

```

43 \msg_new:nnn
44 { hyp }
45 { pdfa-no-push-button }
46 { PDF/A:~Push~button~with~JavaScript~is~prohibited }
47
48 \msg_new:nnn
49 { hyp }
50 { pdfa-no-reset-button }
51 { PDF/A:~Reset~action~is~prohibited }

```

pdf/A message for not allowed Named actions

```
52 \msg_new:nnn
53 { hyp }
54 { pdfa-no-named-action }
55 { PDF/A:~Named~action~#1~is~prohibited }
```

A message if the destination name is empty.

```
56 \msg_new:nnn
57 { hyp }
58 { empty-destination-name }
59 {
60   Empty~destination~name,\\
61   using~'~#1'
62 }
```

A message if the destination check fails

```
63 \msg_new:nnn
64 { hyp }
65 { invalid-destination-value }
66 {
67   Invalid~value~'~#1'~of~'~#2'   \\
68   is~replaced~by~'Fit'~\msg_line_context:.
69 }
```

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

```
70 \msg_new:nnn
71 { hyp }
72 { ignore-deprecated-or-unknown-option-in-pdf-version }
73 {
74   Option~'~#1'~is~unknown~or~deprecated~in\\
75   pdf~version~#2.~Ignored.
76 }
77 \msg_new:nnn
78 { hyp }
79 { ignore-deprecated-or-unknown-value-in-pdf-version }
80 {
81   Value~'~#1'~is~unknown~or~deprecated~in\\
82   pdf~version~#2.~Ignored.
83 }
84 \msg_new:nnn
85 { hyp }
86 { replace-deprecated-or-unknown-value-in-pdf-version }
87 {
88   Value~'~#1'~is~unknown~or~deprecated~in\\
89   pdf~version~#2. Value~'~#3'~is used instead.
90 }
```

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:

```
91 \msg_new:nnn
92 { hyp }
93 { unknown-key }
94 {
95   unknown~key~#2~of~module~'~#1'~set~to~'~#3'.
96 }
```

```

97 \msg_new:nnn
98 { hyp }
99 { unknown-key-to-Hyp }
100 {
101   ignored-in-family-Hyp-unknown-key-#1.
102 }

```

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

```

103 \cs_new:Npn \__hyp_clist_display:n #1 {*-#1\\}
104 \msg_new:nnn
105 { hyp }
106 { unknown-choice }
107 {
108   Value-#3'-is-invalid-for-key-#1'.\\
109   The-key-accepts-only-the-choices\\
110   \clist_map_function:nN { #2 }\__hyp_clist_display:n
111 }
112
113 \msg_new:nnn
114 { hyp }
115 { unknown-choice+empty }
116 {
117   Value-#3'-is-invalid-for-key-#1'.\\
118   The-key-accepts-only-the-choices\\
119   \clist_map_function:nN { #2 }\__hyp_clist_display:n
120   An-empty-value-removes-the-setting.
121 }
122
123 \msg_new:nnn
124 { hyp }
125 { no-bool }
126 {
127   Value-#2'-is-invalid-for-key-#1'.\\
128   The-key-accepts-only-the-choices\\
129   *-true\\
130   *-false \\
131   *-and-an-empty-value-which-removes-the-setting.\\
132   No-value-is-equivalent-to-using-true'.
133 }

```

2 Variants

```

134 \cs_generate_variant:Nn\pdf_destination:nn {nf}
135 \cs_generate_variant:Nn\pdf_object_ref:n {e}

```

3 Overwriting/providing commands from hyperref

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

```

136 \chardef\Hy@VersionChecked=1 %don't check the version!
137 %\cs_set_protected:Npn \PDF@SetupDoc{}
138 %\PDF@FinishDoc{}% dummy needed for hyperref ...

```


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

```
139 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n
```

(End definition for `\hypercalcbp`. This function is documented on page ??.)

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

```
140 \providecommand\@pdfborder{}
141 \providecommand\@pdfborderstyle{}
142 \newcommand\OBJ@OCG@view {} % needed in hyperref
143 \def\Hy@numberline#1{#1\c_space_tl} %needed by bookmark
```

We force unicode as option. As bookmarks are still using `\pdfstringdef` we also need to load `puenc.def`.

```
144 \HyPsd@LoadUnicode
145 \Hy@unicodetrue
146 \let\HyPsd@pdfencoding\HyPsd@pdfencoding@unicode
147 \Hy@DisableOption{unicode}
```

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

```
148 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
149 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:
150 \legacy_if:nT { Hy@setpdfversion }
151 {
152   \msg_warning:nn { hyp }{ pdfversion-disabled }
153 }
154 \Hy@DisableOption{pdfversion}
```

`\Acrobatmenu` should use the new internal link command

```
155 \RenewDocumentCommand \Acrobatmenu { m m }
156 {
157   \hyper@linknamed {#1} {#2}
158 }
```

`\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!

```
159 % TODO should go at some time ...
160 % \kv@set@family@handler{Hyp}
161 % { \msg_warning:nnx {hyp}{unknown-key-to-Hyp}{#1} }
162 \cs_set_protected:Npn \hypersetup #1
163 {
164   %\kvsetkeys{Hyp} {#1}
165   \keys_set:nn { hyp / setup }{ #1 }
166 }
167 % TODO for now unknown keys should only give warnings.
168 \keys_define:nn { hyp / setup }
169 {
170   unknown .code:n =
171   {
172     \msg_warning:nnxxx { hyp } { unknown-key }
173     { hyp~/~setup }{ \l_keys_key_str } { #1 }
174   }
175 }
```

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`)

```

176
177 \bool_new:N \l__hyp_href_url_encode_bool
178 \bool_new:N \l__hyp_href_url_ismap_bool
179 \tl_new:N \l__hyp_href_url_protocol_tl
180 \tl_new:N \l__hyp_href_pdf_destination_tl
181 \tl_new:N \l__hyp_href_pdf_page_tl
182 \tl_new:N \l__hyp_href_run_parameter_tl
183 \cs_new_protected:Npn \__hyp_href_url_format {\begingroup\Url}
184
185
186 \keys_define:nn { hyp / href }
187 {
188   ,urlencode .bool_set:N = \l__hyp_href_url_encode_bool
189   ,format .code:n = { \cs_set_eq:NN \__hyp_href_url_format #1 },
190   ,protocol .tl_set:N = \l__hyp_href_url_protocol_tl
191   ,destination .tl_set:N = \l__hyp_href_pdf_destination_tl
192   ,pdfremotestartview .code:n =
193     {
194       \keys_set:nn { hyp / setup }
195       { pdfremotestartview = #1 }
196     }
197   ,page .code:n =
198     {
199       \tl_set:Nn \l__hyp_href_pdf_page_tl {#1}
200       \tl_set:Nn \Hy@href@page {#1}
201     }
202   ,ismap .bool_set:N = \l__hyp_href_url_ismap_bool
203   ,run-parameter .tl_set:N = \l__hyp_href_run_parameter_tl
204   ,nextactionraw .code:n =
205     { %perhaps some safety match later, see hyperref code
206       \tl_if_empty:nTF
207       {
208         \pdfdict_remove:nn{l_hyp/annot/A}{Next}
209       }
210       {
211         \pdfdict_put:nnn{l_hyp/annot/A}{Next}{#1}
212         \tl_set:Nn \Hy@href@nextactionraw {/Next~#1}
213         \keys_set:nn {hyp / setup }{ pdfnewwindow = true}
214       }
215     }
216   ,afrelationship .code:n =
217     {
218       \pdfdict_put:nnx
219       { l_pdffile/Filespec}{AFRelationship}{ \pdf_name_from_unicode_e:n {#1}}
220     }
221 }
222
223
224 \keys_define:nn { hyp / setup }

```

```

225 {
226   ,href / urlencode .bool_set:N = \l__hyp_href_url_encode_bool
227   ,href / urlencode .default:n = {true}
228   ,href / urlencode .initial:n = {false}
229   ,href / protocol .tl_set:N = \l__hyp_href_url_protocol_tl
230   ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
231   ,href / format .code:n = { \cs_set_eq:NN \__hyp_href_url_format #1 }
232 }
233
234 \hook_new_pair:nn{cmd/href/before}{cmd/href/after}
235
236 \DeclareRobustCommand*{\href}[1][{}]{%
237   \mode_leave_vertical:
238   \hook_use:n{cmd/href/before}
239   \group_begin:
240   \keys_set:nn { hyp / href } {#1}
241   \let\protect\unexpandable@protect
242   \bool_if:NTF \l__hyp_href_url_encode_bool
243     {
244       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
245     }
246     {
247       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
248     }
249   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
250 }
251
252 \begingroup
253   \catcode'\$=6 %
254   \catcode'\#=12 %
255   \gdef\href@#1{\expandafter\href@split$1##\}%
256   \gdef\href@split$1#$2#$3\\$4{%
257     \hyper@@link{$1}{$2}{$4}%<---_hyp-docstrip doubling!
258   }
259   \hook_use:n{cmd/href/after}
260 }%
261 \endgroup
262
263 \hook_new_pair:nn{cmd/hrefurl/before}{cmd/hrefurl/after}
264
265 \DeclareRobustCommand*{\hrefurl}[1][{}]{%
266 {
267   \mode_leave_vertical:
268   \hook_use:n{cmd/href/before}
269   \group_begin:
270   \keys_set:nn { hyp / href } {#1}
271   \let\protect\unexpandable@protect
272   \bool_if:NTF \l__hyp_href_url_encode_bool
273     {
274       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
275     }
276     {
277       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
278     }

```

```

279 \hyper@normalise\__hyp_href_url_aux:nn}
280
281 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
282 {
283   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
284   \group_end:
285   \hook_use:n{cmd/href/after}
286 }
287
288 \hook_new_pair:nn{cmd/hrefpdf/before}{cmd/hrefpdf/after}
289 \DeclareRobustCommand*{\hrefpdf}[1] []
290 {
291   \mode_leave_vertical:
292   \hook_use:n{cmd/hrefpdf/before}
293   \group_begin:
294   \keys_set:nn { hyp / href } {#1}
295   \let\protect\@unexpandable@protect
296   \hyper@normalise\__hyp_href_pdf_aux:nn
297 }
298
299 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
300 {
301   \exp_args:Nnno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
302   \group_end:
303   \hook_use:n{cmd/hrefpdf/after}
304 }
305
306 \hook_new_pair:nn{cmd/hrefrun/before}{cmd/hrefrun/after}
307 \DeclareRobustCommand*{\hrefrun}[1] []
308 {
309   \mode_leave_vertical:
310   \hook_use:n{cmd/hrefrun/before}
311   \group_begin:
312   \keys_set:nn { hyp / href } {#1}
313   \let\protect\@unexpandable@protect
314   \hyper@normalise\__hyp_href_run_aux:nn
315 }
316
317 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
318 {
319   \exp_args:Nnno\hyper@linklaunch{#2}{#1}{\l__hyp_href_run_parameter_tl}
320   \group_end:
321   \hook_use:n{cmd/hrefrun/after}
322 }
323
324
325 \hook_new_pair:nn{cmd/url/before}{cmd/url/after}
326
327 \DeclareRobustCommand*{\url}[1] []
328 {
329   \mode_leave_vertical:
330   \hook_use:n{cmd/url/before}
331   \group_begin:
332   \keys_set:nn { hyp / href } {#1}

```

```

333 \let\protect \@unexpandable@protect
334 \bool_if:NTF \l__hyp_href_url_encode_bool
335 {
336   \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
337 }
338 {
339   \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
340 }
341 \hyper@normalise\__hyp_href_url_aux:n
342 }
343
344 \cs_new_protected:Npn \__hyp_href_url_aux:n #1
345 {
346   \exp_args:Nno
347   \hyper@linkurl{\__hyp_href_url_format {#1}}
348   {\l__hyp_href_url_protocol_tl#1}
349   \group_end:
350   \hook_use:n{cmd/url/after}
351 }
352

```

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

```

353
354 \def\urldef#1#2{\begingroup\def\url{\hyper@normalise\url@}\setbox\z@\hbox\bgroup
355   \def\url@HyperHook##1\endgroup{\url@def{#1}{#2}}%
356   % Because hyperref breaks \urldef and does not define its own (Grrrr!)...
357   \def\url@##1{\egroup\endgroup\DeclareRobustCommand#1{#2{##1}}}%
358   #2}
359

```

make the new commands compatible with `\pdfstringdef`:

```

360 \NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
361 \pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
362 \pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}
363 \pdfstringdefDisableCommands{\let\hrefrun\__hyp_secondoftwowithopt:wnn}

```

4 Compability commands

4.1 Metadata

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

```

364 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
365 {
366   %\tl_set:cn {@#1}{#2}
367   \AddToDocumentProperties[hyperref]{#1}{#2}
368 }
369 \cs_generate_variant:Nn \__hyp_store_metadata:nn {xn,nx,xx}

```

4.2 citecolor

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

```

370 \hook_new:n{hyp/link/cite}
371 \color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
372 \color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
373 \keys_define:nn { hyp / setup }
374 {
375   ,citecolor .code:n = {\__hyp_color_set:ne {hyp/color/cite}{#1}}
376   ,citebordercolor
377   .code:n = {\__hyp_color_set:ne {hyp/color/citeborder}{#1}}
378 }
379 \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
380 {
381   \keys_set:nn { hyp / setup }
382   {
383     ,linkbordercolor= hyp/color/citeborder
384     ,linkcolor       = hyp/color/cite
385   }
386 }
```

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

```

387 \bool_lazy_and:nnF
388 { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
389 { \msg_error:nn { hyp}{ missing-resource-management } }
```

Outlines/bookmarks require the bookmark package. TODO check pdfpagemode if bookmarks are suppressed.

```

390 \legacy_if:nT { Hy@bookmarks }
391 {
392   \AddToHook{begindocument/before}[hyperref/bookmark]{\RequirePackage{bookmark}}
393 }
394 \legacy_if:nT { Hy@draft }
395 {
396   \PassOptionsToPackage{draft}{bookmark}
397 }
```

6 Reference and label commands

The code uses the l3ref-tmp package which must have been loaded as pdfmanagement is a requirement. The commands use after the module prefix always `_ref`.

At first a label command which add the space commands from LaTeX:

```

\__hyp_ref_label:nn
\__hyp_ref_if_exist:nn 398 %
\__hyp_ref_check:nn 399 \cs_new_protected:Npn \__hyp_ref_label:nn #1 #2 %label/attributes
\__hyp_ref_value:nn 400 {
```

```

401 \bsphack
402 \ref_label:nn{#1}{#2}
403 \esphack
404 }
405 \cs_generate_variant:Nn \__hyp_ref_label:nn {en}

```

This provides a condition which tests if a label/attribute combination is known

```

406 \prg_new_eq_conditional:Nn \__hyp_ref_if_exist:nn \ref_if_exist:nn { p , T , F, TF }
407 \prg_generate_conditional_variant:Nnn \__hyp_ref_if_exist:nn {en} { p , T , F, TF }

```

This checks if the label/attribute is known and issues a warning if not. It then also triggers the standard rerun message.

```

408 \cs_new_protected:Npn \__hyp_ref_check:nn #1 #2 %label/attribute
409 {
410   \__hyp_ref_if_exist:nnF {#1}{#2}
411   {
412     \protect\G@refundefinedtrue
413     \@latex@warning
414     {
415       Reference~‘\tl_to_str:n {#1}’~with~attribute~‘\tl_to_str:n {#2}’~
416       on~page~\thepage~\space undefined
417     }
418   }
419 }
420 \cs_generate_variant:Nn \__hyp_ref_check:nn {en}

```

This retrieves a value, it is a simple wrapper around the `\ref_value:nn`

```

421 \cs_new:Npn \__hyp_ref_value:nn #1 #2
422 {
423   \ref_value:nn{#1}{#2}
424 }
425 \cs_generate_variant:Nn \__hyp_ref_value:nn {en}

```

(End definition for `__hyp_ref_label:nn` and others.)

7 Variables

7.1 Private temporary variables

At first a few generic tmp variables

```

\l__hyp_tmpa_tl
\l__hyp_tmpa_seq
\l__hyp_tmpa_int
\l__hyp_tmpa_box
\l__hyp_tmpa_str
426 \box_new:N \l__hyp_tmpa_box
427 \tl_new:N \l__hyp_tmpa_tl
428 \seq_new:N \l__hyp_tmpa_seq
429 \int_new:N \l__hyp_tmpa_int
430 \str_new:N \l__hyp_tmpa_str

```

(End definition for `\l__hyp_tmpa_tl` and others.)

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

TODO: document and check use!

```

\l__hyp_dest_name_tmpa_tl
\l__hyp_uri_tmpa_tl
\l__hyp_filename_tmpa_tl
\l__hyp_text_tmpa_str\l__hyp_text_tmpa_str
431 \tl_new:N \l__hyp_dest_name_tmpa_tl
432 \tl_new:N \l__hyp_uri_tmpa_tl
433 \tl_new:N \l__hyp_filename_tmpa_tl

```

```

434 \tl_new:N \l__hyp_para_tmpa_tl
435 \str_new:N \l__hyp_text_tmpa_str
436 \str_new:N \g__hyp_text_tmpa_str

```

(End 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.

```

437 \tl_const:Nn \c__hyp_dest_undefined_tl {UNDEFINED}

```

(End 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
`\c__hyp_map_annot_hyp_prop` annot names to hyperref names and back.
`\c__hyp_map_hyp_annot_prop`

```

438 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
439 {url,link,file,menu,run}
440 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
441 {
442   URI    = url,
443   GoTo   = link,
444   GoToR  = file,
445   Named  = menu,
446   Launch = run
447 }
448 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
449 {
450   url    = URI,
451   link   = GoTo,
452   file   = GoToR,
453   menu   = Named,
454   run    = Launch
455 }
456

```

(End 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
`\g__hyp_dest_pdfstartview_tl` the current and remote files. The instruction is in “PDF format” but without the leading
`\l__hyp_dest_pdfremotestartview_tl` slash!

```

457 \tl_new:N \g__hyp_dest_pdfstartpage_tl
458 \tl_new:N \g__hyp_dest_pdfstartview_tl
459 \tl_new:N \l__hyp_dest_pdfremotestartview_tl

```

(End 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

460 \tl_new:N \l__hyp_text_enc_uri_print_tl
461 \tl_new:N \l__hyp_text_enc_info_print_tl
462 \tl_new:N \l__hyp_text_enc_dest_tl
463 \tl_new:N \l__hyp_text_enc_dest_print_tl
464 \tl_new:N \l__hyp_text_enc_file_print_tl
465 \tl_new:N \l__hyp_text_enc_para_print_tl
466
467 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
468 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
469 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
470 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
471 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
472 \tl_set:Nn \l__hyp_text_enc_para_print_tl {utf8/string}

```

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

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

```
473 \tl_new:N \l__hyp_dest_pdfview_tl
```

(End 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`

```
474 \str_new:N \g__hyp_bordercolormodel_str
```

(End 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_colorrurl_bool`

`\l_hyp_annot_colormenu_bool`

```

475 \seq_map_inline:Nn \c__hyp_annot_types_seq
476 {
477   \bool_new:c {l_hyp_annot_color#1_bool}
478 }

```

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

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

`\l_hyp_annot_ocgcolorurl_bool`

`\l_hyp_annot_ocgcolorfile_bool`

`\l_hyp_annot_ocgcolorrurl_bool`

`\l_hyp_annot_ocgcolormenu_bool`

```

479 \seq_map_inline:Nn \c__hyp_annot_types_seq
480 {
481   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
482 }

```

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

not_Named_bool\l__hyp_annot_Launch_bool

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

```
483 \seq_map_inline:Nn \c_pdfannot_link_types_seq
484 {
485   \bool_new:c {l__hyp_annot_#1_bool}
486   \bool_set_true:c {l__hyp_annot_#1_bool}
487 }
```

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

7.5 Boxes

\l__hyp_dest_box

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

```
488 \box_new:N \l__hyp_dest_box
```

(End 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.

```
489 \regex_const:Nn \c__hyp_dest_startview_regex
490 {
491   \A\ *
492   (?:
493     (?:XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
494     |
495     (?:Fit\b|FitB\b)
496     |
497     (?:\b(?:FitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1})
498     |
499     (?:FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
500   )
501 }
```

(End definition for \c__hyp_dest_startview_regex.)

7.7 PDF dictionaries

l__hyp_page/Trans

This dictionary is used for page transitions.

```
502 \pdfdict_new:n {l__hyp_page/Trans}
503 \pdfdict_put:nnn {l__hyp_page/Trans}{Type}{/Trans}
```

(End 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

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

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

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

__hyp_text_purify:nN

```
505 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
506 {
507     \str_set:Nx #2 {\text_purify:n { #1 } }
508 }
```

(End 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

```
509 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
510 {
511
512 }
```

(End 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

```
513 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
514 {
515     \pdf_string_from_unicode:nVN { #1 } #2 #2
516 }
```

(End 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

```
517 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
518 {
519     \group_begin:
520     \Hy@pdfstringtrue
```

```

521 \hook_use:n {hyp/text/pdfstring}
522 \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
523 \__hyp_text_cleanup:N \l__hyp_text_tmpa_str
524 \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
525 \str_gset_eq:NN \g__hyp_text_tmpa_str\l__hyp_text_tmpa_str
526 \group_end:
527 \str_set_eq:NN #3 \g__hyp_text_tmpa_str
528 }
529 \cs_generate_variant:Nn \__hyp_text_pdfstring:nnN {xnN,onN,ooN,noN}

```

(End definition for `__hyp_text_pdfstring:nnN`.)

!!! temporary until all instances are gone

```

530 \cs_new_protected:Npn \Hy@pstringdef #1 #2
531 { \__hyp_text_pdfstring:xnN {#2} {utf8/string-raw}#1 }

```

This is a special version for info keys:

`__hyp_text_pdfstring_info:nN`

```

532 \cs_new_protected:Npn \__hyp_text_pdfstring_info:nN #1 #2
533 {
534   \__hyp_text_pdfstring:noN { #1 } { \l__hyp_text_enc_info_print_tl } #2
535 }

```

(End 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-28 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
536 \cs_new_protected:Npn\__hyp_PageLabels_gpush:
537   {
538     \pdfmanagement_add:nnx {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
539   }
540
541 \def\Hy@PutCatalog #1 {}
542
543
544 \legacy_if:nT { Hy@pdfpagelabels }
545   {
546     \cs_set_protected:Npn \HyPL@StorePageLabel #1
547       {
548         \tl_gput_right:Nx \HyPL@Labels { \the\Hy@abspage<<#1>> }
549         \__hyp_PageLabels_gpush:
550       }
551   }

```

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

10 Core Hyperref Commands

Every hyperref has to define eight core command:

```

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

```

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

10.1 Anchors / destinations

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

```

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

```

The <destination name> is encoded with the method stored in 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`

```

552 \Hy@WrapperDef \__hyp_destination:nn #1 #2
553 {
554   \mode_if_horizontal:T { \@savs\spacefactor }
555   \Hy@SaveLastskip      %defined in hyperref
556   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
557   \__hyp_text_pdfstring:xoN
558   { \HyperDestNameFilter{#1} }
559   { \l__hyp_text_enc_dest_tl }
560   \l__hyp_tmpa_tl
561   \str_if_eq:nnTF {#2} {fitrbox}
562   {
563     \exp_args:NV
564     \pdf_destination:nnnn \l__hyp_tmpa_tl
565     { \box_wd:N \l__hyp_dest_box }
566     { \box_ht:N \l__hyp_dest_box }
567     { \box_dp:N \l__hyp_dest_box }
568   }
569   {
570     \exp_args:NV
571     \pdf_destination:nf
572     { \l__hyp_tmpa_tl }
573     { #2 }
574   }
575   \Hy@RestoreLastskip %defined in hyperref
576   \mode_if_horizontal:T { \spacefactor\@savs }
577 }

```

(End 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.

```

\hyper@anchor
\hyper@anchorstart
\hyper@anchorend
578 \cs_new_protected:Npn \hyper@anchor #1
579 {
580   \exp_args:NnV
581   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
582 }
583
584 \cs_new_protected:Npn \hyper@anchorstart #1
585 {
586   \Hy@activeanchortrue
587   \exp_args:NnV
588   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
589 }
590
591 \cs_new_protected:Npn \hyper@anchorend
592 {
593   \Hy@activeanchorfalse
594 }

```

(End definition for \hyper@anchor, \hyper@anchorstart, and \hyper@anchorend. These functions are documented on page ??.)

10.2 GoTo Links

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

At first the internal link commands:

```

595 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
596 {
597   \mode_leave_vertical:
598   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
599   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
600   {
601     \msg_warning:nnx
602     { hyp }
603     { empty-destination-name }
604     { \c__hyp_dest_undefined_tl }
605     \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
606   }
607   {
608     \__hyp_text_pdfstring:xoN
609     { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
610     { \l__hyp_text_enc_dest_tl }
611     \l__hyp_dest_name_tmpa_tl
612   }
613   \exp_args:No
614   \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
615 }
616
617 \cs_new_protected:Npn \__hyp_link_goto_end:
618 {
619   \pdfannot_link_goto_end:
620 }
```

Now the three `hyperref` commands. The splitted commands `\hyper@linkstart` and `\hyper@linkend` are used for footnotemarks, toc and natbib-cites.

`\hyper@link` `\hyper@link{<context>}{<destination name>}{<link text>}`

This creates a complete GoTo link around the `<link text>` pointing to `<destination name>`. The hook `hyp/link/<context>` is executed at the begin if it exists.

The only `<context>` for which a hook is predefined is `cite`. Packages which want to use another `<context>` should initialize the hook like this:

```

\IfHookExistsTF{hyp/link/context}{ }
{ \NewHook{hyp/link/context} }
```

The hook code is executed in a group but before all the pdfannot hooks.

| | |
|--|---|
| <code>\hyper@linkstart</code> <code>\hyper@linkend</code> | <code>\hyper@linkstart{<context>}{<destination name>}</code> <code>\hyper@linkend</code> |
|--|---|

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.

```

621
622 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
623 {
624   \bool_if:NTF \l__hyp_annot_GoTo_bool
625   {
626     \Hy@VerboseLinkStart{#1}{#2}
627     \group_begin:
628     \hook_use:n {hyp/link/#1}
629     \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
630     \__hyp_link_goto_end:
631     \group_end:
632     \Hy@VerboseLinkStop
633   }
634   {{\let\protect\relax#3}}
635 }
636
637 \cs_new_protected:Npn \hyper@linkstart #1 #2 % #1 context, #2=destination name
638 {
639   \bool_if:NT \l__hyp_annot_GoTo_bool
640   {
641     \Hy@VerboseLinkStart{#1}{#2}% only for debug
642     \group_begin:
643     \hook_use:n {hyp/link/#1}
644     \__hyp_link_goto_begin:nw {#2}
645   }
646 }
647
648 \cs_new_protected:Npn \hyper@linkend
649 {
650   \bool_if:NT \l__hyp_annot_GoTo_bool
651   {
652     \__hyp_link_goto_end:
653     \group_end:
654     \Hy@VerboseLinkStop
655   }
656 }

```

10.3 URI links

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

```

657 \pdfdict_new:n {l_hyp/annot/A/URI}
658 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
659 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
660

```



```

661 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
662 {
663   \bool_if:NTF \l__hyp_annot_URI_bool
664   {
665     \group_begin:
666     \__hyp_text_pdfstring:xoN
667       { #2}
668       { \l__hyp_text_enc_uri_print_tl }
669     \l__hyp_uri_tmpa_tl
670     \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
671     \bool_if:NT \l__hyp_href_url_ismap_bool
672     {
673       \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
674     }
675     \cs_set_eq:NN \# \c_hash_str
676     \cs_set_eq:NN \% \c_percent_str
677     \Hy@safe@activestrue
678     \mode_leave_vertical:
679     \pdfannot_dict_put:nnx {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
680     \pdfannot_link:nnx { URI }
681     {
682     }
683     {
684       \let\protect\relax
685       #1
686       \Hy@xspace@end
687       \Hy@VerboseLinkStop %where is the start??
688     }
689     \group_end:
690   }
691   {{\let\protect\relax#1}}
692 }
693

```

10.4 GoToR Links files

```

694 \pdfdict_new:n {l_hyp/annot/A/GoToR}
695 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
696 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
697
698
699 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
700 {
701   \bool_if:NTF \l__hyp_annot_GoToR_bool
702   {
703     \group_begin:
704     \tl_set:Nx \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
705     \exp_args:Nx
706     \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
707     {
708       \pdfdict_put:nnx { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n {ap
709       \exp_args:Nnox
710       \pdffile_embed_file:nnn
711       {}

```

```

712         {\l__hyp_filename_tmpa_tl }
713         {\__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
714     }
715     \pdfdict_put:nnx
716     {l_hyp/annot/A/GoToR}
717     {F}
718     {\pdf_object_ref:e {\__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
719     \__hyp_text_pdfstring:nnN
720     { #3 }
721     { \l__hyp_text_enc_dest_print_tl }
722     \l__hyp_dest_name_tmpa_tl
723     \tl_if_blank:eTF {#3}
724     {
725         \pdfdict_put:nnx {l_hyp/annot/A/GoToR}{D}
726         {
727             [
728                 \int_eval:n
729                 { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
730                 /\l__hyp_dest_pdfremotestartview_tl
731             ]
732         }
733     }
734     {
735         \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
736     }
737     \mode_leave_vertical:
738     \pdf_object_unnamed_write:nx{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
739     \pdfannot_dict_put:nnx {link/GoToR}{A}{\pdf_object_ref_last:}
740     \pdfannot_link:nxn %expansion??
741     { GoToR }
742     {
743     }
744     {
745         \let\protect\relax
746         #1\Hy@xspace@end
747         \Hy@VerboseLinkStop %where is the start??
748     }
749     \group_end:
750 }
751 {{\let\protect\relax#1}}
752 }

```

10.5 Launch links

We define `\hyper@linklaunch` for naming consistency

```

753 \pdfdict_new:n {l_hyp/annot/A/Launch}
754 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
755 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
756
757 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
758 {
759     \bool_if:NTF \l__hyp_annot_Launch_bool
760     {
761         \group_begin:

```

```

762 \__hyp_text_pdfstring:nnN
763 { #1 }
764 { \l__hyp_text_enc_file_print_tl }
765 \l__hyp_filename_tmpa_tl
766 \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
767 \__hyp_text_pdfstring:noN
768 { #3 }
769 { \l__hyp_text_enc_para_print_tl }
770 \l__hyp_para_tmpa_tl
771 \bool_if:nTF
772 {
773   \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
774   ||
775   \pdf_version_compare_p:Nn > {1.9}
776 }
777 {
778   \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
779 }
780 {
781   \pdfdict_put:nnx {l_hyp/annot/A/Launch}{Win}
782   {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
783 }
784 \mode_leave_vertical:
785 \pdfannot_dict_put:nxx {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}>>}
786 \pdfannot_link:nxn
787 { Launch }
788 {
789   % /A
790   % <<
791   % \pdfdict_use:n {l_hyp/annot/A/Launch}
792   % >>
793 }
794 {
795   \let\protect\relax
796   #2\Hy@xspace@end
797   \Hy@VerboseLinkStop %where is the start??
798 }
799 \group_end:
800 }
801 {{\let\protect\relax#2}}
802 }

```

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

```

803 \use:x
804 { % filename, anchor text, linkname
805   \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str ##1 \exp_not:N \ \ ##2 #
806 }
807 {
808   \hyper@linklaunch {#1}{#2}{#3}
809 }

```

10.6 Named links (menu)

We also define \hyper@linknamed for consistency.

```

810 \pdfdict_new:n {l_hyp/annot/A/Named}
811 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
812 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
813
814 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
815 {
816   \bool_if:NTF \l__hyp_annot_Named_bool
817   {
818     \group_begin:
819     \pdfmeta_standard_verify:nnTF {named_actions}{#1}
820     {
821       \mode_leave_vertical:
822       \pdfdict_put:nnx {l_hyp/annot/A/Named}{N}
823       {\pdf_name_from_unicode_e:n{#1}}
824       \pdfannot_dict_put:nnx {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}>>}
825       \pdfannot_link:nxn { Named }
826       {
827         /A
828         <<
829         \pdfdict_use:n { l_hyp/annot/A/Named }
830         >>
831       }
832       {
833         #2
834         \Hy@xspace@end
835         \Hy@VerboseLinkStop
836       }
837     }
838     {
839       \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
840       #2
841     }
842     \group_end:
843   }
844   {\let\protect\relax#2}}
845 }
846

```

11 Link decorations

11.1 Functions to export and select colors

We support two input syntax: color expressions and model with values.

```

\__hyp_color_export:nnN \__hyp_color_export:nnN {<color>} {<export format>} {tlvar}

```

This exports a color as space separated numbers as needed in in /C. {<color>} should have either the format [model]{value} or be a color expression. For examples: [rgb]{1,0,.5} or red!50!blue. The export format is either space-sep-cmyk or space-sep-rgb.

`_hyp_color_export:nnN` Color keys need to parse color expressions. Two input types are supported: `color=[rgb]{1,0,.5}`
`_hyp_color_export_aux:wnnN` and `color=red!50!blue`. The colors are exported to PDF syntax (space separated numbers).

```

847 \cs_new_protected:Npn \_hyp_color_export:nnN #1 #2 #3
848 {
849   \tl_if_head_eq_charcode:nNTF {#1}[ %]
850   {
851     \_hyp_color_export_aux:wnnN #1 {#2} #3
852   }
853   {
854     \color_export:nnN {#1} {#2} #3
855   }
856 }
857
858 \cs_new_protected:Npn \_hyp_color_export_aux:wnnN [#1] #2 #3 #4
859 {
860   \color_export:nnnn {#1}{#2}{#3}#4
861 }
862
863 \cs_generate_variant:Nn \_hyp_color_export:nnN {xVN}

```

(End definition for _hyp_color_export:nnN and _hyp_color_export_aux:wnnN.)

`_hyp_color_select:n` `_hyp_color_select:nN {<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`.

```

864 \cs_new_protected:Npn \_hyp_color_select:n #1
865 {
866   \tl_if_head_eq_charcode:nNTF {#1}[ %]
867   {
868     \_hyp_color_select_aux:wn #1
869   }
870   {
871     \color_select:n {#1}
872   }
873 }
874
875 \cs_new_protected:Npn \_hyp_color_select_aux:wn [#1] #2
876 {
877   \color_select:nn {#1}{#2}
878 }
879
880 \cs_generate_variant:Nn \_hyp_color_select:n {e}

```

(End definition for _hyp_color_select:n and _hyp_color_select_aux:wn.)

`_hyp_color_set:nn` `_hyp_color_set:nN {< name >} {<color>}`

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

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

```

881 \cs_new_protected:Npn \__hyp_color_set:nn #1 #2
882 {
883   \tl_if_head_eq_charcode:nNTF {#2}[ %]
884   {
885     \__hyp_color_set_aux:nwn { #1 } #2
886   }
887   {
888     \color_set:nn {#1} {#2}
889   }
890 }
891
892 \cs_new_protected:Npn \__hyp_color_set_aux:nwn #1 [#2] #3
893 {
894   \color_set:nnn {#1}{#2}{#3}
895 }
896
897 \cs_generate_variant:Nn \__hyp_color_set:nn {ne}

```

(End 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.

```

898 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
899 {
900   \hook_gput_code:nnn
901     {pdfannot/link/#2/begin}
902     {hyp/color}
903     {
904       \bool_if:cT { l_hyp_annot_color#1_bool }
905       {
906         \group_begin:
907         \color_select:n { hyp/color/#1}
908       }
909     }
910   \hook_gput_code:nnn
911     {pdfannot/link/#2/end}
912     {hyp/color}
913     {
914       \bool_if:cT { l_hyp_annot_color#1_bool }
915       {
916         \group_end:
917       }
918     }
919 }

```

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

```

920 \keys_define:nn { hyp / setup }
921 {

```

```

922     ,colorlinks .meta:n =
923     {
924         ,pdfborder={0~0~0}
925         ,pdfborderstyle=
926         ,colorurl  =#1
927         ,colorlink =#1
928         ,colorrun  =#1
929         ,colormenu =#1
930         ,colorfile =#1
931     }
932     ,colorlinks .default:n = {true}
933 }

colorurl (setup key)
colorlink (setup key)
colorrun (setup key)
colormenu (setup key)
colorfile (setup key)
urlcolor (setup key)
linkcolor (setup key)
runcolor (setup key)
menucolor (setup key)
filecolor (setup key)
allcolor (setup key)
934 \seq_map_inline:Nn \c__hyp_annot_types_seq
935 {
936     \keys_define:nn { hyp / setup }
937     {
938         ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
939         ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
940     }
941 }
942
943 \keys_define:nn { hyp / setup }
944 {
945     ,allcolors .meta:n =
946     {
947         ,urlcolor=#1
948         ,linkcolor=#1
949         ,runcolor=#1
950         ,filecolor=#1
951         ,menucolor=#1
952     }
953     ,allcolors .value_required:n = true
954 }

```

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)
955 \keys_define:nn { hyp / setup }
956 {
957     ,bordercolormodel .choices:nn =
958     {rgb,cmyk}
959     { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
960     ,bordercolormodel .initial:n ={rgb}
961 }
962 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
963 {

```

```

964 \keys_define:nn { hyp / setup }
965 {
966   #1bordercolor .code:n =
967   {
968     \tl_if_empty:nTF { ##1 }
969     {
970       \pdfannot_dict_remove:nn
971       {link/#2}
972       { C }
973     }
974     {
975       \__hyp_color_export:xVN {##1}\g__hyp_bordercolormodel_str \l__hyp_tmpa_tl
976       \pdfannot_dict_put:nnx
977       {link/#2}
978       { C }
979       { [\l__hyp_tmpa_tl] }
980     }
981   }
982 }
983 }
984
985 \keys_define:nn { hyp / setup }
986 {
987   ,allbordercolors .meta:n =
988   {
989     ,linkbordercolor=#1
990     ,urlbordercolor =#1
991     ,filebordercolor=#1
992     ,menubordercolor=#1
993     ,runbordercolor =#1
994   }
995   ,allbordercolors .value_required:n = true
996 }
997

```

11.3.2 Borderwidth and -arc

```

998 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
999 {
1000   \keys_define:nn { hyp / setup }
1001   {
1002     #1border .code:n =
1003     {
1004       \tl_if_empty:nTF { ##1 }
1005       {
1006         \pdfannot_dict_remove:nn
1007         {link/#2}
1008         { Border }
1009       }
1010       {
1011         \pdfannot_dict_put:nnn
1012         {link/#2}
1013         { Border }
1014         { [##1] }

```



```

1015         }
1016     }
1017 }
1018 }
1019 \keys_define:nn { hyp / setup }
1020 {
1021     ,pdfborder .code:n =
1022     {
1023         \tl_if_empty:nTF { #1 }
1024         {
1025             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1026             {
1027                 \pdfannot_dict_remove:nn
1028                 {link/##2}
1029                 { Border }
1030             }
1031         }
1032         {
1033             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1034             {
1035                 \pdfannot_dict_put:nnn
1036                 {link/##2}
1037                 { Border }
1038                 { [#1] }
1039             }
1040         }
1041     }
1042     ,pdfborder .initial:n = {0~0~1},
1043 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1044 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1045 {
runborderstyle (setup key) 1046     \keys_define:nn { hyp / setup }
fileborderstyle (setup key) 1047     {
menuborderstyle (setup key) 1048         #1borderstyle .code:n =
1049         {
1050             \tl_if_empty:nTF { ##1 }
1051             {
1052                 \pdfannot_dict_remove:nn
1053                 {link/#2}
1054                 { BS }
1055             }
1056             {
1057                 \pdfannot_dict_put:nnn
1058                 {link/#2}
1059                 { BS }
1060                 { <<##1>> }
1061             }
1062         }
1063     }

```

```

1064 }
1065 \keys_define:nn { hyp / setup }
1066 {
1067   ,pdfborderstyle .code:n =
1068   {
1069     \tl_if_empty:nTF { #1 }
1070     {
1071       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1072       {
1073         \pdfannot_dict_remove:nn
1074         {link/##2}
1075         { BS }
1076       }
1077     }
1078     {
1079       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1080       {
1081         \pdfannot_dict_put:nnn
1082         {link/##2}
1083         { BS }
1084         { <<#1>> }
1085       }
1086     }
1087   }
1088   ,pdfborderstyle .initial:n = {},
1089 }

```

11.4 ocgcolorlinks

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

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

```

1090 \cs_new_protected:Npn \__hyp_ocg_init:
1091 {
1092   \pdf_object_new:nn { __hyp/OCG/View } { dict }
1093   \pdf_object_new:nn { __hyp/OCG/Print } { dict }
1094   \pdf_object_new:nn { __hyp/OCG/config } { dict }
1095   \pdf_object_new:nn { __hyp/OCG/refarray } { array }
1096   \pdf_object_write:nx { __hyp/OCG/refarray }
1097   {
1098     \pdf_object_ref:n { __hyp/OCG/View }
1099     \c_space_tl
1100     \pdf_object_ref:n { __hyp/OCG/Print }
1101   }
1102   \pdf_object_write:nn { __hyp/OCG/View }
1103   {
1104     /Type/OCG
1105     /Name(View)
1106     /Usage
1107     <<
1108     /Print <</PrintState/OFF>>~
1109     /View <</ViewState/ON >>~

```

```

1110         >>
1111     }
1112     \pdf_object_write:nn { __hyp/OCG/Print }
1113     {
1114         /Type/OCG
1115         /Name(Print)
1116         /Usage
1117         <<
1118             /Print <</PrintState/ON>>~
1119             /View <</ViewState/OFF>>~
1120         >>
1121     }
1122     \pdfmanagement_add:nnx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/Print} }
1123     \pdfmanagement_add:nnx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/Print} }
1124     \pdf_object_write:nx { __hyp/OCG/config }
1125     {
1126         /OFF[\pdf_object_ref:n { __hyp/OCG/Print } ]
1127         /AS[
1128             <<
1129                 /Event/View
1130                 /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1131                 /Category[/View]
1132             >>
1133             <<
1134                 /Event/Print
1135                 /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1136                 /Category[/Print]
1137             >>
1138             <<
1139                 /Event/Export
1140                 /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1141                 /Category[/Print]
1142             >>
1143         ]
1144     }
1145     \pdfmanagement_add:nnx { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG/Print} }
1146     \cs_gset:Npn \__hyp_ocg_init: {}
1147 }

```

(End 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.

```

1148 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1149 {
1150     \hook_gput_code:nnn
1151     {pdfannot/link/#2/begin}
1152     {hyp/ocg}
1153     {
1154         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1155         {
1156             \__hyp_ocg_init:
1157             \group_begin:
1158             \hbox_set:Nw \l__hyp_tmpa_box
1159         }
1160     }

```

```

1160     }
1161     \hook_gput_code:nnn
1162     {pdfannot/link/#2/end}
1163     {hyp/ocg}
1164     {
1165         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1166         {
1167             \hbox_set_end:
1168             \mbox
1169             {
1170                 \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1171                 \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1172                 \pdf_emc:
1173                 \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1174                 \group_begin:
1175                 \color_select:n { hyp/color/#1 }
1176                 \box_use_drop:N \l__hyp_tmpa_box
1177                 \group_end:
1178                 \pdf_emc:
1179             }
1180             \group_end:
1181         }
1182     }
1183 }

```

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

```

ocgcolorlink (setup key) 1184 \bool_lazy_or:nnTF
ocgcolorurl (setup key) 1185 { \pdf_version_compare_p:Nn > {1.4} }
ocgcolorfile (setup key) 1186 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
ocgcolormenu (setup key) 1187 {
ocgcolorrun (setup key) 1188     \keys_define:nn { hyp / setup }
1189     {
1190         ,_ocgcolorlinks .meta:n =
1191         {
1192             ocgcolorlink=#1,
1193             ocgcolorurl=#1,
1194             ocgcolorfile=#1,
1195             ocgcolorrun=#1,
1196             ocgcolormenu=#1
1197         }
1198         ,_ocgcolorlinks .default:n = true
1199     }
1200 }
1201 {
1202     \keys_define:nn { hyp / setup }
1203     {
1204         ,_ocgcolorlinks .code:n =
1205         {
1206             \msg_warning:nnxx
1207             { hyp }
1208             { ignore-deprecated-or-unknown-option-in-pdf-version }
1209             { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1210         }
1211     }

```

```

1212 }
1213
1214 \keys_define:nn { hyp / setup }
1215 {
1216   ,ocgcolorlinks .choice:
1217   ,ocgcolorlinks / true .meta:n =
1218   {
1219     pdfborder      = {0~0~0},
1220     pdfborderstyle = {},
1221     colorlinks     = false,
1222     _ocgcolorlinks = true
1223   }
1224   ,ocgcolorlinks / false .meta:n =
1225   {
1226     _ocgcolorlinks = false
1227   }
1228   ,ocgcolorlinks .default:n = {true}
1229 }
1230
1231 \seq_map_inline:Nn \c__hyp_annot_types_seq
1232 {
1233   \bool_lazy_or:nnTF
1234   { \pdf_version_compare_p:Nn > {1.4} }
1235   { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1236   {
1237     \keys_define:nn { hyp / setup }
1238     {
1239       ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1240     }
1241   }
1242   {
1243     \keys_define:nn { hyp / setup }
1244     {
1245       ,ocgcolor#1 .code:n=
1246       {
1247         \msg_warning:nnxx
1248         { hyp }
1249         { ignore-deprecated-or-unknown-option-in-pdf-version }
1250         { ocgcolor#1 }
1251         { \pdf_version_major:.\pdf_version_minor: }
1252       }
1253     }
1254   }
1255 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1256 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1257 {
1258   \keys_define:nn { hyp / setup }
1259   {
1260     ,#1highlight .choices:nn =
1261     { /I, /N, /O, /P}

```

```

1262     {
1263         \pdfannot_dict_put:nnn
1264         {link/#2}
1265         { H }
1266         { ##1 }
1267     }
1268 }
1269 ,#1highlight / .code:n =
1270 {
1271     \pdfannot_dict_remove:nn
1272     {link/#2}
1273     { H }
1274 }
1275 }
1276 ,#1highlight / unknown .code:n =
1277 {
1278     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1279     { pdfhighlight }
1280     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1281     { \exp_not:n {##1} }
1282 }
1283 }
1284 }
1285
1286
1287 \keys_define:nn { hyp / setup }
1288 {
1289     ,pdfhighlight .choices:nn =
1290     { /I, /N, /O, /P}
1291     {
1292         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1293         {
1294             \pdfannot_dict_put:nnn
1295             {link/####2}
1296             { H }
1297             { #1 }
1298         }
1299     }
1300     ,pdfhighlight / .code:n =
1301     {
1302         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1303         {
1304             \pdfannot_dict_remove:nn
1305             {link/##2}
1306             { H }
1307         }
1308     }
1309     ,pdfhighlight .initial:n = {/I},
1310     ,pdfhighlight / unknown .code:n =
1311     {
1312         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1313         { pdfhighlight }
1314         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1315         { \exp_not:n {##1} }

```

```

1316     }
1317 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1318 \keys_define:nn { hyp / setup }
hideurl (setup key) 1319 {
hidefile (setup key) 1320   hidelinks .meta:n =
hiderun (setup key) 1321   {
hidemenu (setup key) 1322     ,colorlinks      = false
1323     ,ocgcolorlinks = false
1324     ,pdfborder      = { 0~0~0 }
1325     ,pdfborderstyle=
1326   }
1327 }
1328
1329 \seq_map_inline:Nn \c__hyp_annot_types_seq
1330 {
1331   \keys_define:nn { hyp / setup }
1332   {
1333     hide#1 .meta:n =
1334     {
1335       ,color#1      = false
1336       ,ocgcolor#1   = false
1337       ,#1border      = { 0~0~0 }
1338       ,#1borderstyle =
1339     }
1340   }
1341 }

```

11.7 color schemes and settings

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

```

colorscheme (setup key)
1342 \keys_define:nn { hyp / setup }
1343 {
1344   colorscheme .code:n =
1345   {
1346     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1347     {
1348       \keys_set:nn { hyp /setup }
1349       {
1350         ##1 = ##2
1351       }
1352     }
1353   }
1354 }
1355 \keys_set:nn { hyp / setup } {colorscheme=phetype}

```

12 Keys

12.1 Ignored keys

The following are ignored (with or without warnings)

```
unicode (setup key)
pdfencoding (setup key) 1356 \keys_define:nn { hyp / setup }
pdfversion (setup key) 1357 {
1358     ,unicode .code:n = {}
1359     ,pdfencoding .code:n = {}
1360     ,pdfversion .code:n =
1361     {
1362         \msg_warning:nn { hyp }{ pdfversion-disabled }
1363     }
1364 }
1365 %
```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```
verbose (setup key)
debug (setup key) 1366 \keys_define:nn { hyp / setup }
draft (setup key) 1367 {
1368     ,verbose .choice:
1369     ,verbose / true .code:n = { \Hy@verbosetrue}
1370     ,verbose / false .code:n = { \Hy@verbosefalse}
1371     ,verbose .default:n = {true}
1372     ,debug .meta:n = {verbose=#1}
1373     ,debug .default:n = {true}
1374 }
1375 \keys_define:nn { hyp / setup }
1376 {
1377     ,draft .code:n =
1378     {
1379         \Hy@drafttrue
1380         \PassOptionsToPackage{draft}{bookmark}
1381     }
1382     ,final .code:n =
1383     {
1384         \Hy@finaltrue
1385         \PassOptionsToPackage{final}{bookmark}
1386     }
1387 }

extension (setup key)
hypertextnames (setup key) 1388 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1389 {
1390     ,extension .tl_set:N = \XR@ext
1391     ,extension .initial:n= pdf
1392     ,hypertextnames .choice:
1393     ,hypertextnames / true .code:n = { \Hy@hypertextnamestrue}
1394     ,hypertextnames / false .code:n = { \Hy@hypertextnamesfalse}

extension (setup key)
linktoc (setup key)
linktocpage (setup key)
plainpages (setup key)
localanchorname (setup key)
linkfileprefix (setup key)
```



```

1395 ,hypertextnames .default:n = {true}
1396 ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1397 ,localanchorname .choice:
1398 ,localanchorname / true .code:n = { \Hy@localanchorname true }
1399 ,localanchorname / false .code:n = { \Hy@localanchorname false }
1400 ,localanchorname .default:n = {true}
1401 ,naturalnames .choice:
1402 ,naturalnames / true .code:n = { \Hy@naturalname true }
1403 ,naturalnames / false .code:n = { \Hy@naturalname false }
1404 ,naturalnames .default:n = {true}
1405 ,pageanchor .choice:
1406 ,pageanchor / true .code:n = { \Hy@pageanchor true }
1407 ,pageanchor / false .code:n = { \Hy@pageanchor false }
1408 ,pageanchor .default:n = {true}
1409 ,plainpages .choice:
1410 ,plainpages / true .code:n = { \Hy@plainpage true }
1411 ,plainpages / false .code:n = { \Hy@plainpage false }
1412 ,plainpages .default:n = {true}
1413 }
1414
1415 \keys_define:nn { hyp / setup }
1416 {
1417 ,linktoc .choices:nn = { none, section, all, page }
1418 {
1419 \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1420 }
1421 ,linktoc / unknown .code:n =
1422 {
1423 \msg_warning:nxxx { hyp } { unknown-choice }
1424 { linktoc }
1425 { none, section, all, page }
1426 { \exp_not:n {#1} }
1427 }
1428 ,linktocpage .choice:
1429 ,linktocpage / true .meta:n = {linktoc=page}
1430 ,linktocpage / false .meta:n = {linktoc=section}
1431 ,linktocpage .default:n = true
1432 }
1433

```

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

```

url (setup key) 1434 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1435 {
menu (setup key) 1436 \keys_define:nn { hyp / setup }
run (setup key) 1437 {
1438 ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1439 }
1440 }

```

```

1441 \keys_define:nn { hyp / setup }
1442 {
1443 ,baseurl .code:n =
1444 {
1445 \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl

```

```

1446     \tl_if_empty:NTF \l__hyp_tmpa_tl
1447     {
1448         \pdfmanagement_remove:nn {Catalog} { URI }
1449     }
1450     {
1451         \pdfmanagement_add:nnx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1452     }
1453 }
1454 %only false does something ...
1455 ,bookmarks .choice:
1456 ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1457 ,bookmarks / true .code:n = {}
1458 ,bookmarks .default:n = {true}
1459 ,bookmarksnumbered .choice:
1460 ,bookmarksnumbered / false .code:n = { \Hy@bookmarksnumberedfalse }
1461 ,bookmarksnumbered / true .code:n = { \Hy@bookmarksnumberedtrue }
1462 ,bookmarksnumbered .default:n = {true}
1463 ,bookmarksopen .choice:
1464 ,bookmarksopen / false .code:n = { \Hy@bookmarksopenfalse }
1465 ,bookmarksopen / true .code:n = { \Hy@bookmarksopentruetrue }
1466 ,bookmarksopen .default:n = {true}
1467 ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1468 ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1469 ,pdfcenterwindow .choice:
1470 ,pdfcenterwindow / false .code:n =
1471 {
1472     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1473 }
1474 ,pdfcenterwindow / true .code:n =
1475 {
1476     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1477 }
1478 ,pdfcenterwindow / .code:n =
1479 {
1480     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1481 }
1482 ,pdfcenterwindow / unknown .code:n =
1483 {
1484     \msg_warning:nnxx { hyp } { no-bool }
1485     { pdfcenterwindow }
1486     { \exp_not:n {#1} }
1487 }
1488 ,pdfcenterwindow .default:n = true
1489 ,pdffdirection .choice:
1490 ,pdffdirection / L2R .code:n =
1491 {
1492     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1493 }
1494 ,pdffdirection / R2L .code:n =
1495 {
1496     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1497 }
1498 ,pdffdirection / .code:n =
1499 {

```

```

1500     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1501   }
1502 ,pdfdirection / unknown .code:n =
1503   {
1504     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1505     { pdfdirection }
1506     { L2R , R2L }
1507     { \exp_not:n {#1} }
1508   }
1509 ,pdffitdisplaydoctitle .choice:
1510 ,pdffitdisplaydoctitle / false .code:n =
1511   {
1512     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1513   }
1514 ,pdffitdisplaydoctitle / true .code:n =
1515   {
1516     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1517   }
1518 ,pdffitdisplaydoctitle .default:n = true
1519 ,pdfduplex .choices:nn =
1520 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1521 {
1522   \pdf_version_compare:NnTF > {1.6}
1523   {
1524     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1525       { PrintDuplex } { /#1 }
1526   }
1527   {
1528     \msg_warning:nnxx
1529       {hyp}
1530       {ignore-deprecated-or-unknown-option-in-pdf-version}
1531       {pdfduplex}
1532       {\pdf_version:}
1533   }
1534 }%
1535 ,pdfduplex / .code:n =
1536   {
1537     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1538   }
1539 ,pdfduplex / unknown .code:n =
1540   {
1541     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1542     { pdfduplex }
1543     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1544     { \exp_not:n {#1} }
1545   }
1546 ,pdffitwindow .choice:
1547 ,pdffitwindow / false .code:n =
1548   {
1549     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1550   }
1551 ,pdffitwindow / true .code:n =
1552   {
1553     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }

```

```

1554     }
1555 ,pdffitwindow / .code:n =
1556     {
1557         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1558     }
1559 ,pdffitwindow .default:n = true
1560 ,pdffitwindow / unknown .code:n =
1561     {
1562         \msg_warning:nnxx { hyp } { no-bool }
1563         { pdffitwindow }
1564         { \exp_not:n {#1} }
1565     }
1566 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1567 ,pdflinkmargin .initial:n = {1pt}
1568 ,pdfmenubar .choice:
1569 ,pdfmenubar / true .code:n =
1570     {
1571         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1572     }
1573 ,pdfmenubar / false .code:n =
1574     {
1575         \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1576         { HideMenubar } { true }
1577     }
1578 ,pdfmenubar / .code:n =
1579     {
1580         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1581     }
1582 ,pdfmenubar .default:n = true
1583 ,pdfmenubar / unknown .code:n =
1584     {
1585         \msg_warning:nnxx { hyp } { no-bool }
1586         { pdfmenubar }
1587         { \exp_not:n {#1} }
1588     }
1589 ,pdfnewwindow .choice:
1590 ,pdfnewwindow / true .code:n =
1591     {
1592         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1593         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1594     }
1595 ,pdfnewwindow / false .code:n =
1596     {
1597         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1598         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1599     }
1600 ,pdfnewwindow / .code:n =
1601     {
1602         \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1603         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1604     }
1605 ,pdfnonfullscreenpagemode .choices:nn =
1606 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1607 {

```

```

1608     \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1609     { NonFullScreenPageMode} {/#1}
1610 }
1611 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1612 {
1613     \pdf_version_compare:NnTF < {1.6}
1614     {
1615         %message
1616     }
1617     {
1618         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1619         {NonFullScreenPageMode}{/UseAttachments}
1620     }
1621 }
1622 ,pdfnonfullscreenpagemode / .code:n =
1623 {
1624     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1625 }
1626 ,pdfnonfullscreenpagemode / unknown .code:n =
1627 {
1628     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1629     { pdfnonfullscreenpagemode }
1630     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1631     { \exp_not:n {#1} }
1632 }
1633 ,pdfnumcopies .code:n =
1634 {
1635     \pdf_version_compare:NnTF > {1.6}
1636     {
1637         \tl_if_empty:nTF {#1}
1638         {
1639             \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1640         }
1641         {
1642             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1643             {NumCopies}{#1}
1644         }
1645     }
1646     {
1647         \msg_warning:nnxx
1648         {hyp}
1649         {ignore-deprecated-or-unknown-option-in-pdf-version}
1650         {pdfnumcopies}
1651         {\pdf_version:}
1652     }
1653 }
1654 ,pdfpagelayout .choices:nn =
1655 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1656 { \pdfmanagement_add:nnx {Catalog} { PageLayout }{ /#1 } }
1657 ,pdfpagelayout / .code:n =
1658 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1659 ,pdfpagelayout / unknown .code:n =
1660 {
1661     \msg_warning:nnxxx { hyp } { unknown-choice+empty }

```

```

1662     { pdfpagelayout }
1663     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1664     { \exp_not:n {#1} }
1665 }
1666 ,pdfpagemode .choices:nn =
1667 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1668 { \pdfmanagement_add:nnx {Catalog} { PageMode }{ /#1 } }
1669 ,pdfpagemode / UseAttachments .code:n =
1670 {
1671     \pdf_version_compare:NnTF > {1.5}
1672     {
1673         \pdfmanagement_add:nnx {Catalog} { PageMode }{ /UseAttachments }
1674     }
1675     {
1676         \msg_warning:nnxx
1677         {hyp}
1678         {ignore-deprecated-or-unknown-value-in-pdf-version}
1679         {UseAttachments}
1680         {\pdf_version:}
1681     }
1682 }
1683 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1684 ,pdfpagemode / unknown .code:n =
1685 {
1686     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1687     { pdfpagemode }
1688     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1689     { \exp_not:n {#1} }
1690 }
1691 ,pdfpagescrop .code:n =
1692 {
1693     \tl_if_empty:nTF %or blank?
1694     {
1695         \pdfmanagement_remove:nn {Pages} { CropBox }
1696     }
1697     {
1698         \pdfmanagement_add:nnx {Pages} { CropBox } { [#1] }
1699     }
1700 }
1701 ,pdfpicktraybypdfsize .choice:
1702 ,pdfpicktraybypdfsize / true .code:n =
1703 {
1704     \pdf_version_compare:NnTF > {1.6}
1705     {
1706         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1707         { PickTrayByPDFSize } { true }
1708     }
1709     {
1710         \msg_warning:nnxx
1711         {hyp}
1712         {ignore-deprecated-or-unknown-option-in-pdf-version}
1713         {pdfpicktraybypdfsize}
1714         {\pdf_version:}
1715     }

```

```

1716     }
1717 ,pdfpicktraybypdfsize / false .code:n =
1718 {
1719     \pdf_version_compare:NnTF > {1.6}
1720     {
1721         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1722         { PickTrayByPDFSize } { false }
1723     }
1724     {
1725         \msg_warning:nnxx
1726         {hyp}
1727         {ignore-deprecated-or-unknown-option-in-pdf-version}
1728         {pdfpicktraybypdfsize}
1729         {\pdf_version:}
1730     }
1731 }
1732 ,pdfpicktraybypdfsize / .code:n =
1733 {
1734     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1735 }
1736 ,pdfpicktraybypdfsize / unknown .code:n =
1737 {
1738     \msg_warning:nnxx { hyp } { no-bool }
1739     { picktraybypdfsize }
1740     { \exp_not:n {#1} }
1741 }
1742 ,pdfprintarea .choices:nn =
1743 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1744 {
1745     \pdf_version_compare:NnTF < {2.0}
1746     {
1747         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1748         { PrintArea } { /#1 }
1749     }
1750     {
1751         \msg_warning:nnxx
1752         {hyp}
1753         {ignore-deprecated-or-unknown-option-in-pdf-version}
1754         {pdfprintarea}
1755         {\pdf_version:}
1756     }
1757 }%
1758 ,pdfprintarea / .code:n =
1759 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1760 ,pdfprintarea / unknown .code:n =
1761 {
1762     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1763     { pdfprintarea }
1764     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1765     { \exp_not:n {#1} }
1766 }
1767 ,pdfprintclip .choices:nn =
1768 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1769 {

```

```

1770     \pdf_version_compare:NnTF < {2.0}
1771     {
1772         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1773         { PrintClip } { /#1 }
1774     }
1775     {
1776         \msg_warning:nnxx
1777         {hyp}
1778         {ignore-deprecated-or-unknown-option-in-pdf-version}
1779         {pdfprintclip}
1780         {\pdf_version:}
1781     }
1782 }%
1783 ,pdfprintclip / .code:n =
1784 {
1785     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1786 }
1787 ,pdfprintclip / unknown .code:n =
1788 {
1789     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1790     { pdfprintclip }
1791     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1792     { \exp_not:n {#1} }
1793 }
1794 ,pdfprintpagerange .code:n =
1795 {
1796     \pdf_version_compare:NnTF > {1.6}
1797     {
1798         \tl_if_empty:nTF { #1}
1799         {
1800             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1801             { PrintPageRange }
1802         }
1803         {
1804             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1805             {PrintPageRange}{[#1]}
1806         }
1807     }
1808     {
1809         \msg_warning:nnxx
1810         {hyp}
1811         {ignore-deprecated-or-unknown-option-in-pdf-version}
1812         {pdfprintpagerange}
1813         {\pdf_version:}
1814     }
1815 }
1816 ,pdfprintscaling .choices:nn =
1817 { None, AppDefault }
1818 {
1819     \pdf_version_compare:NnTF > {1.5}
1820     {
1821         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1822         { PrintScaling } { /#1 }
1823     }

```



```

1824         {
1825             \msg_warning:nxxx
1826             {hyp}
1827             {ignore-deprecated-or-unknown-option-in-pdf-version}
1828             {pdfprintscaling}
1829             {\pdf_version:}
1830         }
1831     }%
1832 ,pdfprintscaling / .code:n =
1833     {
1834         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1835     }
1836 ,pdfprintscaling / unknown .code:n =
1837     {
1838         \msg_warning:nxxx { hyp } { unknown-choice+empty }
1839         { pdfprintarea }
1840         { None, AppDefault }
1841         { \exp_not:n {#1} }
1842     }
1843 ,pdfremotestartview .code:n =
1844     {
1845         \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1846         \exp_args:NNV
1847         \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1848         {
1849             \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1850         }
1851         {
1852             \msg_warning:nxxx {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1853             \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1854         }
1855     }
1856 ,pdfremotestartview .initial:n = {Fit}
1857 % pdfstartpage is special as it shares code with pdfstartview
1858 ,pdfstartpage .code:n =
1859     {
1860         \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1861         \bool_if:nTF
1862         { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1863         {
1864             \pdfmanagement_remove:nn {Catalog} { OpenAction }
1865         }
1866         {
1867             \pdfmanagement_add:nnx {Catalog} { OpenAction }
1868             {
1869                 [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1870             }
1871         }
1872     }
1873 ,pdfstartpage .initial:n =1
1874 ,pdfstartview .code:n =
1875     {
1876         \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1877         \exp_args:NNV

```

```

1878 \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1879 {
1880   \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1881 }
1882 {
1883   \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1884   \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1885 }
1886 \bool_if:nTF
1887 { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1888 {
1889   \pdfmanagement_remove:nn {Catalog} { OpenAction }
1890 }
1891 {
1892   \pdfmanagement_add:nnx {Catalog} { OpenAction }
1893   {
1894     [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1895   }
1896 }
1897 }
1898 ,pdfstartview .initial:n = Fit
1899 ,pdftoolbar .choice:
1900 ,pdftoolbar / true .code:n =
1901 {
1902   \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1903 }
1904 ,pdftoolbar / false .code:n =
1905 {
1906   \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
1907   { HideToolbar } { true }
1908 }
1909 ,pdftoolbar / true .code:n =
1910 {
1911   \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1912 }
1913 ,pdftoolbar .default:n = true
1914 ,pdftoolbar / unknown .code:n =
1915 {
1916   \msg_warning:nxxx { hyp } { no-bool }
1917   { pdftoolbar }
1918   { \exp_not:n {#1} }
1919 }
1920 % pdfview see below.
1921 ,pdfviewarea .choices:nn =
1922 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1923 {
1924   \pdf_version_compare:NnTF < {2.0}
1925   {
1926     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
1927     { ViewArea } { /#1 }
1928   }
1929   {
1930     \msg_warning:nxxx
1931     {hyp}

```

```

1932         {ignore-deprecated-or-unknown-option-in-pdf-version}
1933         {pdfviewarea}
1934         {\pdf_version:}
1935     }
1936 }%
1937 ,pdfviewarea / .code:n =
1938 {
1939     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1940 }
1941 ,pdfviewarea / unknown .code:n =
1942 {
1943     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1944     { pdfviewarea }
1945     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1946     { \exp_not:n {#1} }
1947 }
1948 ,pdfviewclip .choices:nn =
1949 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1950 {
1951     \pdf_version_compare:NnTF < {2.0}
1952     {
1953         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1954         { ViewClip } { /#1 }
1955     }
1956     {
1957         \msg_warning:nnxx
1958         {hyp}
1959         {ignore-deprecated-or-unknown-option-in-pdf-version}
1960         {pdfviewclip}
1961         {\pdf_version:}
1962     }
1963 }%
1964 ,pdfviewclip / .code:n =
1965 {
1966     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
1967 }
1968 ,pdfviewclip / unknown .code:n =
1969 {
1970     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1971     { pdfviewclip }
1972     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1973     { \exp_not:n {#1} }
1974 }
1975 ,pdfwindowui .choice:
1976 ,pdfwindowui / true .code:n =
1977 {
1978     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
1979 }
1980 ,pdfwindowui / false .code:n =
1981 {
1982     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1983     { HideWindowUI } { true }
1984 }
1985 ,pdfwindowui / .code:n =

```

```

1986     {
1987       \pdfmanagement_remove:nn {Catalog / ViewerPreferences} {HideWindowUI}
1988     }
1989     ,pdfwindowui / unknown .code:n =
1990     {
1991       \msg_warning:nnxx {hyp} {no-bool}
1992       { pdfwindowui }
1993       { \exp_not:n {#1} }
1994     }
1995     ,pdfwindowui .default:n = true
1996   }

```

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

```

1997   \keys_define:nn {hyp / setup}
1998   {
1999     ,pdfview .code:n =
2000     {
2001       \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
2002       \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
2003       {
2004         { xyz }
2005         {
2006           \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq} > { 1 }
2007           {
2008             \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2009             \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2010             {
2011               \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2012             }
2013             {
2014               \tl_set:Nx \l__hyp_dest_pdfview_tl
2015               {
2016                 \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2017               }
2018             }
2019           }
2020           {
2021             \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2022           }
2023         }
2024         { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2025         { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2026         { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2027         { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2028         { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2029         { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2030         { fitr }
2031         {
2032           \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq} = {1}
2033           {
2034             \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2035           }
2036           {
2037             %ensure 4 values ...

```

```

2038 \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2039 \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2040 \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2041 \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2042 \hbox_set_to_wd:Nnn \l__hyp_dest_box
2043 {
2044   \fp_eval:n
2045   {
2046     round
2047     (
2048       abs
2049       (
2050         \seq_item:Nn\l__hyp_tmpa_seq{4}
2051         -
2052         (\seq_item:Nn\l__hyp_tmpa_seq{2})
2053       ),
2054       3
2055     )
2056   }bp
2057 }{}
2058 \box_set_dp:Nn \l__hyp_dest_box
2059 {
2060   \fp_eval:n
2061   {
2062     round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2063   }bp
2064 }
2065 \box_set_ht:Nn \l__hyp_dest_box
2066 {
2067   \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2068 }
2069 }
2070 }
2071 }
2072 {
2073   \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2074   \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2075 }
2076 }
2077 ,pdfview .initial:n = {xyz}
2078 }

```

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.

```

2079 \keys_define:nn { hyp / setup }
2080 {
2081   ,pdflang .code:n =
2082   {
2083     \tl_if_empty:nTF { #1 }
2084     {

```

```

2085         \pdfmanagement_remove:nn {Catalog} { Lang }
2086     }
2087     {
2088         \pdfmanagement_add:nnx {Catalog} { Lang } { (#1) }
2089     }
2090     \__hyp_store_metadata:nn {pdflang}{#1}
2091 }
2092 }

```

12.3.1 “info keys”

pdfauthor (*setup key*) The keys store their value also in the metadata container, so that hyperxmp can use
pdftitle (*setup key*) them.

```

pdfcreator (setup key) 2093 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
pdfsubject (setup key) 2094 {
pdfproducer (setup key) 2095     \keys_define:nn { hyp / setup }
pdfkeywords (setup key) 2096     {
2097         pdf#1 .code:n =
2098         {
2099             \tl_if_blank:nTF {##1}
2100             {
2101                 \pdfmanagement_remove:nn {Info}{#2}
2102             }
2103             {
2104                 \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2105                 \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2106                 {
2107                     \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2108                 }
2109             }
2110             \__hyp_store_metadata:nn {pdf#1}{##1}
2111         }
2112     }
2113     \keys_define:nn { hyp / info }
2114     {
2115         #2 .code:n =
2116         {
2117             \tl_if_blank:nTF {##1}
2118             {
2119                 \pdfmanagement_remove:nn {Info}{#2}
2120             }
2121             {
2122                 \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2123                 \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2124                 {
2125                     \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2126                 }
2127             }
2128             \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2129         }
2130     },unknown .code:n =
2131     {
2132         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2133         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}

```

```

2134         {
2135             \exp_args:Nno
2136             \pdfmanagement_add:nnx {Info}
2137             { \l_keys_key_str } {\l__hyp_tmpa_str}
2138         }
2139     }
2140 }
2141 }
2142 \__hyp_setup_info_key:nn {author} {Author}
2143 \__hyp_setup_info_key:nn {title} {Title}
2144 \__hyp_setup_info_key:nn {producer} {Producer}
2145 \__hyp_setup_info_key:nn {creator} {Creator}
2146 % ignored key: addtopdfcreator
2147 \__hyp_setup_info_key:nn {subject} {Subject}
2148 \__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.

```

2149 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2150 {
2151     \keys_define:nn { hyp / setup }
2152     {
2153         pdf#1 .code:n =
2154         {
2155             \tl_if_blank:nTF {##1}
2156             {
2157                 \pdfmanagement_remove:nn {Info}{#2}
2158             }
2159             {
2160                 \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
2161             }
2162             \__hyp_store_metadata:nn {pdf#1}{##1}
2163         }
2164     }
2165     \keys_define:nn { hyp / info }
2166     {
2167         #2 .code:n =
2168         {
2169             \tl_if_blank:nTF {##1}
2170             {
2171                 \pdfmanagement_remove:nn {Info}{#2}
2172             }
2173             {
2174                 \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
2175             }
2176             \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2177         }
2178     }
2179 }
2180
2181 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2182 \__hyp_setup_info_date_key:nn {moddate} {ModDate}

```

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

```

2183 \keys_define:nn { hyp / setup }
2184 {
2185   ,pdftrapped .code:n =
2186   {
2187     \exp_args:Nne
2188     \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2189   }
2190   ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2191   {
2192     \pdfmanagement_add:nxx {Info}{Trapped}
2193     {/
2194       \str_uppercase:f { \str_head:n { #1 } }
2195       \str_lowercase:f { \str_tail:n { #1 } }
2196     }
2197     \__hyp_store_metadata:nx {pdftrapped}
2198     {
2199       \str_uppercase:f { \str_head:n { #1 } }
2200       \str_lowercase:f { \str_tail:n { #1 } }
2201     }
2202   }
2203   ,_pdftrapped / unknown .code:n =
2204   {
2205     \msg_warning:nnxxx { hyp } { unknown-choice }
2206     { pdftrapped }
2207     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2208     { \exp_not:n {#1} }
2209   }
2210 }

```

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

```

2211 \keys_define:nn { hyp / setup }
2212 {
2213   pdfinfo .code:n =
2214   {
2215     \keys_set:nn { hyp / info } { #1 }
2216   }
2217 }

```

Now we set some default values

```

2218 \keys_set:nn { hyp / setup } {pdfcreator = LaTeX-with-hyperref}
2219 \keys_set:nn { hyp / setup } {pdfauthor = }
2220 \keys_set:nn { hyp / setup } {pdftitle = }
2221 \keys_set:nn { hyp / setup } {pdfsubject = }

```

12.4 hyperxmp keys

hyperxmp defines lots of keys for `\hypersetup`. They now longer work with this driver. So we provide them, but they are only stored as metadata:

```

2222 \clist_map_inline:nn
2223 {
2224   ,pdfcopyright
2225   ,pdftype
2226   ,pdflicenseurl

```



```

2227 ,pdfauthortitle
2228 ,pdfcaptionwriter
2229 ,pdfmetalang
2230 ,pdfapart
2231 ,pdfaconformance
2232 ,pdfuapart
2233 ,pdfxstandard
2234 ,pdfsource
2235 ,pdfdocumentid
2236 ,pdfinstanceid
2237 ,pdfversionid
2238 ,pdfrendition
2239 ,pdfpublication
2240 ,pdfpubtype
2241 ,pdfbytes
2242 ,pdfnumpages
2243 ,pdfissn
2244 ,pdfeissn
2245 ,pdfisbn
2246 ,pdfbookedition
2247 ,pdfpublisher
2248 ,pdfvolumenum
2249 ,pdfissuenum
2250 ,pdfpagerange
2251 ,pdfdoi
2252 ,pdfurl
2253 ,pdfidentifier
2254 ,pdfsubtitle
2255 ,pdfpubstatus
2256 ,pdfcontactaddress
2257 ,pdfcontactcity
2258 ,pdfcontactregion
2259 ,pdfcontactpostcode
2260 ,pdfcontactcountry
2261 ,pdfcontactphone
2262 ,pdfcontactemail
2263 ,pdfcontacturl
2264 ,pdfdate
2265 }
2266 {
2267   \keys_define:nn { hyp / setup }
2268   {
2269     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2270   }
2271 }

```

12.5 Transitions

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

```

2272 \keys_define:nn { hyp / setup }
2273 {
2274   pdfpageduration .code:n =
2275   {
2276     \tl_if_blank:nTF { #1 }

```

```

2277     {
2278     \pdfmanagement_remove:nn {Page}{Dur}
2279     }
2280     {
2281     \pdfmanagement_add:nnn {Page}{Dur}{#1}
2282     }
2283   }
2284 }

```

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}}

```

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

```

2285 \keys_define:nn { hyp / setup }
2286 {
2287   pdfpagetransition .code:n =
2288   {
2289     \tl_if_blank:nTF {#1}
2290     {
2291       \pdfmanagement_remove:nn {Page}{Trans}
2292     }
2293     {
2294       \group_begin:
2295       \keys_set:nn { hyp / trans }{style=R,#1}
2296       \pdf_object_unnamed_write:nx { dict }
2297       {
2298         \pdfdict_use:n {l__hyp_page/Trans}
2299       }
2300       \pdfmanagement_add:nnx {Page}{Trans}{\pdf_object_ref_last:}
2301       \group_end:

```

```

2302     }
2303 }
2304 }
2305 \keys_define:nn { hyp / trans }
2306 {
2307   ,style .choices:nn =
2308     {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2309     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2310   ,style / unknown .code:n =
2311     {
2312       \msg_warning:nnxxx { hyp } { unknown-choice }
2313       { trans / style }
2314       { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2315       { \exp_not:n {#1} }
2316     }
2317   ,duration .code:n =
2318     {
2319       \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}
2320     }
2321   ,direction .choices:nn =
2322     {H,V}
2323     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2324   ,direction .choices:nn =
2325     {0,90,180,270,315}
2326     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ #1 } }
2327   ,direction / None .code:n =
2328     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ /None } }
2329   ,direction / unknown .code:n =
2330     {
2331       \msg_warning:nnxxx { hyp } { unknown-choice }
2332       { trans / direction }
2333       {
2334         H~(horizontal,~only~Split,~Blinds),
2335         V~(vertical,~only~Split,~Blinds),
2336         0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2337         90~(bottom~to~top,~only~Wipe),
2338         180~(right~to~left,~only~Wipe),
2339         270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2340         315~(top~left~to~bottom,~only~Glitter),
2341         None~(only~Fly)
2342       }
2343       { \exp_not:n {#1} }
2344     }
2345   ,motion .choices:nn =
2346     {I,O}
2347     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2348   ,motion / unknown .code:n =
2349     {
2350       \msg_warning:nnxxx { hyp } { unknown-choice }
2351       { trans / motion }
2352       { I~(inwards) , O~(outwards) }
2353       { \exp_not:n {#1} }
2354     }
2355   ,scale .code:n =

```

```

2356     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2357     ,opaque .choices:nn = {true,false}
2358     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B } { #1} }
2359     ,opaque / unknown .code:n =
2360     {
2361         \msg_warning:nnxxx { hyp } { unknown-choice }
2362         { trans / B }
2363         { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2364         { \exp_not:n {#1} }
2365     }
2366     % try to set unknown keys as style
2367     ,unknown .code:n =
2368     {
2369         % warning ...
2370         \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2371     }
2372 }

```

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

```

2373 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}

```

Unfinished Form field code

```

2374 \NewDocumentCommand \MakeFieldObject { m m }
2375 {
2376     \pdfxform_new:nnn { #2 }{ } { #1 }
2377 }
2378
2379
2380 \prop_new:N    \g__hyp_AcroForm_CoFields_prop
2381 \prop_new:N    \g__hyp_AcroForm_Fields_prop
2382
2383 \let\HyField@afields\ltx@empty
2384 \let\HyField@cofields\ltx@empty
2385 %% UF test for old pdftex removed
2386 %\let\HyField@AuxAddToFields\ltx@gobble
2387 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2388 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2389
2390 % the value doesn't matter, but with a prop we avoid duplicates and it is
2391 % clearly faster than removing them from a sequence
2392 \def\HyField@AuxAddToFields#1
2393 {
2394     \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2395 }%
2396
2397 %fields with empty key get a value too -- lets hope that
2398 %this give the expected behaviour
2399 \def\HyField@AuxAddToCoFields #1 #2
2400 {
2401     \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2402 }
2403
2404 \Hy@AtBeginDocument
2405 {
2406     \if@filesw

```

```

2407 \immediate\write\@mainaux{%
2408 \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2409 }%
2410 \immediate\write\@mainaux{%
2411 \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2412 }%
2413 \fi
2414 \let\HyField@AfterAuxOpen\@firstofone
2415 }%
2416
2417 \def\HyField@AddToFields
2418 {
2419 \exp_args:Nx\HyField__hypAddToFields
2420 {
2421 \pdfannot_box_ref_last:
2422 }
2423 \ifx\Fld@calculate@code\ltx@empty
2424 \else
2425 \begingroup
2426 \Hy@safe@activetrue
2427 \edef\Hy@temp{%
2428 \endgroup
2429 \if@files
2430 \write\@mainaux
2431 {
2432 \string\HyField@AuxAddToCoFields
2433 {
2434 \Fld@calculate@sortkey
2435 }
2436 {
2437 \pdfannot_box_ref_last:
2438 }
2439 }
2440 \fi
2441 }%
2442 \Hy@temp
2443 \fi
2444 }%
2445
2446 \def\HyField__hypAddToFields#1{
2447 \HyField@AfterAuxOpen{%
2448 \if@files
2449 \write\@mainaux{%
2450 \string\HyField@AuxAddToFields{#1}%
2451 }%
2452 \fi
2453 }%
2454 }%
2455
2456 \ExplSyntaxOff
2457 \ExplSyntaxOn
2458 \tl_new:N \l__hyp_CheckmarkYes_tl
2459 \tl_set:Nn \l__hyp_CheckmarkYes_tl { __hyp_xform_CheckMarkYes }
2460 \tl_new:N \l__hyp_CheckmarkOff_tl

```

```

2461 \tl_set:Nn \l__hyp_CheckmarkOff_tl { __hyp_xform_CheckMarkOff }
2462
2463 \def\@Form[#1]
2464 {
2465   \@ifundefined{textcolor}{\let\textcolor\@gobble}{}
2466   \kvsetkeys{Form}{#1}
2467   \pdf@ifdraftmode{}
2468   {
2469     \Hy@FormObjects
2470     \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2471     {
2472       \pdfmanagement_add:nnx { Catalog / AcroForm } { Fields }{##1}
2473       %\pdfmanagement_show:n { Catalog / AcroForm }
2474     }
2475     \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2476     {
2477       \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2478       {
2479         \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2480       }
2481       \seq_sort:Nn \l__hyp_tmpa_seq
2482       {
2483         \int_compare:nNTF { \pdf@strcmp{##1}{##2} } > { 0 }
2484         { \sort_return_swapped: }
2485         { \sort_return_same: }
2486       }
2487       \seq_map_inline:Nn \l__hyp_tmpa_seq
2488       {
2489         \pdfmanagement_add:nnx { Catalog / AcroForm }
2490         { CO }
2491         {
2492           \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2493         }
2494       }
2495     }
2496     \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2497     {ZaDb} {\pdf_object_ref:n {\__hyp/Font/ZaDb} }
2498     \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2499     {Helv} {\pdf_object_ref:n {\__hyp/Font/Helv} }
2500     \pdfmanagement_add:nnx {Catalog /AcroForm}
2501     {DA}{(/Helv~10~Tf~0~g)}
2502     \pdfmeta_standard_verify:nTF {form_no_NeedAppearances}
2503     {
2504       \legacy_if:nT { HyField@NeedAppearances }
2505       {
2506         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2507       }
2508     }
2509     {
2510       \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2511     }
2512   }
2513   \MakeFieldObject
2514   {

```

```

2515         \group_begin:
2516         \fontfamily{pzd}
2517         \fontencoding{U}
2518         \fontseries{m}
2519         \fontshape{n}
2520         \selectfont
2521         \char123
2522     \group_end:
2523 }
2524 {__hyp_xform_Ding}
2525 \MakeFieldObject
2526 {
2527     \group_begin:
2528     \fontfamily{pzd}
2529     \fontencoding{U}
2530     \fontseries{m}
2531     \fontshape{n}
2532     \selectfont
2533     \phantom{\char123}
2534     \group_end:
2535 }
2536 {__hyp_xform_DingOff}
2537 \MakeFieldObject
2538 {
2539     \group_begin:
2540     \fontfamily{pzd}
2541     \fontencoding{U}
2542     \fontseries{m}
2543     \fontshape{n}
2544     \selectfont
2545     \char51
2546     \group_end:
2547 }
2548 {__hyp_xform_CheckMarkYes}
2549 \MakeFieldObject
2550 {
2551     \group_begin:
2552     \fontfamily{pzd}
2553     \fontencoding{U}
2554     \fontseries{m}
2555     \fontshape{n}
2556     \selectfont
2557     \phantom{\char51} %perhaps xetex needs some small glyph ..
2558     \group_end:
2559 }
2560 {__hyp_xform_CheckMarkOff}
2561 \MakeFieldObject
2562 {
2563     \fbox{\textcolor{yellow}{\textsf{Submit}}}} %color?
2564 }
2565 {__hyp_xform_Submit}
2566 \MakeFieldObject
2567 {
2568     \fbox{\textcolor{yellow}{\textsf{SubmitP}}}} %color?

```

```

2569     }
2570     {__hyp_xform_SubmitP}
2571 }
2572 \ExplSyntaxOff
2573 \let\@endForm\ltx@empty
2574 \let\HyAnn@AbsPageLabel\ltx@empty
2575 \let\Fld@pageobjref\ltx@empty
2576
2577 \ExplSyntaxOn
2578 \newcount\HyAnn@Count
2579 \HyAnn@Count=\ltx@zero
2580 \def\HyAnn@AbsPageLabel
2581 {
2582   \global\advance\HyAnn@Count by\ltx@one
2583   %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2584   %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}%
2585   %\zref@refused{HyAnn@\the\HyAnn@Count}%
2586   \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
2587   \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
2588 }%
2589 \def\Fld@pageobjref
2590 {
2591   \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
2592   {
2593     /P~\pdf_pageobject_ref:n
2594     {
2595       \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
2596     }
2597   }
2598 }
2599 \ExplSyntaxOff
2600 \ExplSyntaxOn
2601 %% check if the attr should be set through
2602 %% hooks.
2603 %% check if options are missing.
2604 \def\@TextField[#1]#2{% parameters, label
2605   \def\Fld@name{#2}%
2606   \let\Fld@default\ltx@empty
2607   \let\Fld@value\@empty
2608   \def\Fld@width{\DefaultWidthofText}%
2609   \def\Fld@height{%
2610     \ifFld@multiline
2611       \DefaultHeightofTextMultiline
2612     \else
2613       \DefaultHeightofText
2614     \fi
2615   }%
2616   \begingroup
2617     \expandafter\HyField@SetKeys\expandafter{%
2618       \DefaultOptionsofText,#1%
2619     }%
2620     \PDFForm@Name
2621     \HyField@FlagsText
2622     \ifFld@hidden\def\Fld@width{1sp}\fi

```



```

2623 \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2624 \LayoutTextField{#2}{%
2625 \leavevmode
2626 \HyAnn@AbsPageLabel
2627 \Hy@escapeform\PDFForm@Text
2628 \pdfannot_box:nnnn
2629 {\Fld@width}
2630 {\Fld@height}
2631 {Opt} %is this correct?
2632 {\PDFForm@Text}
2633 \MakeTextField{\Fld@width}{\Fld@height}
2634 \HyField@AddToFields
2635 }%
2636 \endgroup
2637 }
2638 \providecommand\@curropt{}
2639 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2640 \def\Fld@name{#2}
2641 \let\Fld@default\relax
2642 \let\Fld@value\relax
2643 \def\Fld@width{\DefaultWidthofChoiceMenu}
2644 \def\Fld@height{\DefaultHeightofChoiceMenu}
2645 \begingroup
2646 \Fld@menulength=0 %
2647 \@tempdima\z@
2648 \clist_map_variable:nNn { #3 } \@curropt
2649 %\@for\@curropt:=#3\do
2650 {%
2651 \expandafter\Fld@checkequals\@curropt==\%
2652 \Hy@StepCount\Fld@menulength
2653 \settowidth{\@tempdimb}{\@currDisplay}%
2654 \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2655 }%
2656 \advance\@tempdima by~15\p@
2657 \begingroup
2658 \HyField@SetKeys{#1}
2659 \edef\x{\endgroup
2660 \noexpand\expandafter
2661 \noexpand\HyField@SetKeys
2662 \noexpand\expandafter{%
2663 \expandafter\noexpand\curname DefaultOptionsof%
2664 \ifFld@radio
2665 Radio%
2666 \else
2667 \ifFld@combo
2668 \ifFld@popdown
2669 PopdownBox%
2670 \else
2671 ComboBox%
2672 \fi
2673 \else
2674 ListBox%
2675 \fi
2676 \fi

```

```

2677         \endcsname
2678     }%
2679 } \x
2680 \HyField@SetKeys{#1}%
2681 \PDFForm@Name
2682 \ifFld@hidden\def\Fld@width{1sp}\fi
2683 \ifx\Fld@value\relax
2684     \let\Fld@value\Fld@default
2685 \fi
2686 \LayoutChoiceField{#2}{%
2687     \ifFld@radio
2688         \HyField@FlagsRadioButton
2689         \__hypRadio{#3}%
2690     \else
2691         \beginngroup
2692             \HyField@FlagsChoice
2693             \ifdim\Fld@width<\@tempdima
2694                 \ifdim\@tempdima<1cm\@tempdima1cm\fi
2695                 \edef\Fld@width{\the\@tempdima}%
2696             \fi
2697             \ifFld@combo
2698             \else
2699                 \@tempdima=\the\Fld@menulength\Fld@charsize
2700                 \advance\@tempdima by~\Fld@borderwidth bp %
2701                 \advance\@tempdima by~\Fld@borderwidth bp %
2702                 \edef\Fld@height{\the\@tempdima}%
2703             \fi
2704             \__hypListbox{#3}%
2705         \endgroup
2706     \fi
2707 }%
2708 \endgroup
2709 }
2710 \tl_new:N \l__hyp_RadioYes_tl
2711 \tl_set:Nn \l__hyp_RadioYes_tl { __hyp_xform_Ding }
2712 \def\__hypRadio#1{%
2713     \Fld@listcount=0-%
2714     %\show\Fld@default
2715     \EdefEscapeName\Fld@default{\Fld@default}%
2716     \clist_map_variable:nNn { #1 } \@curropt
2717     %\@for\@curropt:=#1\do
2718     {%
2719         \expandafter\Fld@checkequals\@curropt==\%
2720         \EdefEscapeName\@currValue{\@currValue}%
2721         \Hy@StepCount\Fld@listcount
2722         \@currDisplay\space
2723         \leavevmode
2724         \HyAnn@AbsPageLabel
2725         \Hy@escapeform\PDFForm@Radio
2726         \pdfannot_box:nnnn
2727             {\Fld@width}
2728             {\Fld@height}
2729             {0pt} %is this correct?
2730         {

```

```

2731         \PDFForm@Radio
2732     /AP
2733     <<
2734     /N
2735     <<
2736     /\currValue\c_space_tl \pdfxform_ref:o {__hyp_xform_Ding}
2737     %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2738     >>
2739     >>
2740     }
2741     {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2742     \int_compare:nNt { \Fld@listcount } = { 1 }
2743     { \HyField@AddToFields }
2744     \c_space_tl % deliberate space between radio buttons
2745     % to do: --> should be configurable
2746 }%
2747 }
2748
2749 \newcount\Fld@listcount
2750 \def\__hypListbox#1
2751 {
2752     \HyField@PDFChoices{#1}
2753     \mode_leave_vertical:
2754     \HyAnn@AbsPageLabel
2755     \Hy@escapeform\PDFForm@List
2756     \pdf_link_user:nnn
2757     {widget} %perhaps we need more types??
2758     {\PDFForm@List}
2759     {\MakeChoiceField{\Fld@width}{\Fld@height}}
2760     \HyField@AddToFields
2761 }
2762
2763
2764 \def\@PushButton[#1]#2{% parameters, label
2765     \def\Fld@name{#2}%
2766     \group_begin:
2767     \exp_args:No\HyField@SetKeys
2768     {
2769         \DefaultOptionsofPushButton,#1
2770     }
2771     \PDFForm@Name
2772     \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2773     {
2774         \HyField@FlagsPushButton
2775         \legacy_if:nT {Fld@hidden}
2776         {
2777             \def\Fld@width{1sp}
2778         }
2779         \LayoutPushButtonField
2780         {
2781             \mode_leave_vertical:
2782             \HyAnn@AbsPageLabel
2783             \Hy@escapeform\PDFForm@Push
2784             \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}

```

```

2785         \pdfannot_box:nnnn
2786         {\box_wd:N\l_tmpa_box}
2787         {\box_ht:N\l_tmpa_box}
2788         {\box_dp:N\l_tmpa_box} %is this correct?
2789         {\PDFForm@Push}
2790         {\box_use:N\l_tmpa_box}
2791         \HyField@AddToFields
2792     }
2793 }
2794 {
2795     \msg_error:nn { hyp }{ pdfa-no-push-button }
2796     \LayoutPushButtonField
2797     {
2798         \mode_leave_vertical:
2799         \MakeButtonField{#2}
2800     }
2801 }
2802 \group_end:
2803 }
2804
2805 \def\@Submit[#1]#2
2806 {
2807     \def\Fld@width {\DefaultWidthofSubmit}
2808     \def\Fld@height{\DefaultHeightofSubmit}
2809     \group_begin:
2810         \exp_args:No\HyField@SetKeys
2811         {
2812             \DefaultOptionsofSubmit,#1
2813         }
2814         \HyField@FlagsPushButton
2815         \HyField@FlagsSubmit
2816         \legacy_if:nT { Fld@hidden }
2817         {
2818             \def\Fld@width{1sp}
2819         }
2820         \mode_leave_vertical:
2821         \HyAnn@AbsPageLabel
2822         \Hy@escapeform\PDFForm@Submit
2823         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2824         \pdfannot_box:nnnn
2825         {\box_wd:N\l_tmpa_box}
2826         {\box_ht:N\l_tmpa_box}
2827         {\box_dp:N\l_tmpa_box} %is this correct?
2828         {
2829             \PDFForm@Submit
2830             /AP<<
2831             /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2832             /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2833             >>
2834         }
2835         \HyField@AddToFields
2836         \box_use:N\l_tmpa_box
2837
2838     \group_end:

```

```

2839 }
2840
2841 \def\@Reset[#1]#2
2842 {
2843   \def\Fld@width {\DefaultWidthofReset}
2844   \def\Fld@height{\DefaultHeightofReset}
2845   \group_begin:
2846     \exp_args:No\HyField@SetKeys
2847     {
2848       \DefaultOptionsofReset,#1
2849     }
2850     \mode_leave_vertical:
2851     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2852     {
2853       \HyField@FlagsPushButton
2854       \legacy_if:nT { Fld@hidden }
2855       { \def\Fld@width{1sp} }
2856       \HyAnn@AbsPageLabel
2857       \Hy@escapeform\PDFForm@Reset
2858       \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2859       \pdfannot_box:nnnn
2860       {\box_wd:N\l_tmpa_box}
2861       {\box_ht:N\l_tmpa_box}
2862       {\box_dp:N\l_tmpa_box} %is this correct?
2863       { \PDFForm@Reset }
2864       \HyField@AddToFields
2865       \box_use:N \l_tmpa_box
2866     }
2867     {
2868       \msg_error:nn { hyp }{ pdfa-no-reset-button }
2869       \MakeButtonField{#2}
2870     }
2871   \group_end:
2872 }
2873
2874 \def\@CheckBox[#1]#2
2875 {% parameters, label
2876   \def\Fld@name{#2}
2877   \def\Fld@default{0}
2878   \group_begin:
2879     \def\Fld@width {\DefaultWidthofCheckBox}
2880     \def\Fld@height{\DefaultHeightofCheckBox}
2881     \exp_args:No\HyField@SetKeys
2882     {
2883       \DefaultOptionsofCheckBox,#1
2884     }
2885     \PDFForm@Name
2886     \HyField@FlagsCheckBox
2887     \legacy_if:nT { Fld@hidden }
2888     {
2889       \def\Fld@width{1sp}
2890     }
2891     \LayoutCheckField{#2}
2892     {

```

```

2893         \mode_leave_vertical:
2894         \HyAnn@AbsPageLabel
2895         \Hy@escapeform\PDFForm@Check
2896         \pdfannot_box:nnnn
2897         {\Fld@width}
2898         {\Fld@height}
2899         {Opt} %is this correct?
2900         {\PDFForm@Check}
2901         \HyField@AddToFields %check if this works with xelatex ...
2902     }
2903     \group_end:
2904 }
2905 \ExplSyntaxOff
2906
2907 %hm. Should a luatex driver use type1 fonts in fields????
2908 \ExplSyntaxOn
2909 \def\Hy@FormObjects
2910 {
2911     \pdf_object_new:nn {__hyp/Encoding/pdfdoc } { dict }
2912     \pdf_object_new:nn {__hyp/Font/ZaDb } { dict }
2913     \pdf_object_new:nn {__hyp/Font/Helv } { dict }
2914     \pdf_object_write:nx {__hyp/Encoding/pdfdoc }
2915     {
2916         /Type/Encoding
2917         /Differences[
2918             24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2919             /ring/tilde
2920             \c_space_tl
2921             39/quotesingle
2922             \c_space_tl
2923             96/grave %
2924             \iow_newline:
2925             128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2926             /fraction/guilsinglleft/guilsinglright/minus/perthousand
2927             /quotedblbase/quotedblleft/quotedblright/quoteleft
2928             /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2929             /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2930             \iow_newline:
2931             164/currency
2932             \c_space_tl
2933             166/brokenbar
2934             \c_space_tl
2935             168/dieresis/copyright/ordfeminine
2936             \c_space_tl
2937             172/logicalnot/.notdef/registered/macron/degree/plusminus
2938             /twosuperior/threesuperior/acute/mu
2939             \c_space_tl
2940             183/periodcentered/cedilla/onesuperior/ordmasculine
2941             \c_space_tl
2942             188/onequarter/onehalf/threequarters
2943             \iow_newline:
2944             192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
2945             /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
2946             /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute

```

```

2947         /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
2948         /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
2949         /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
2950         /cedilla/egrave/eacute/ecircumflex/edieresis/igrave
2951         /iacute/icircumflex/idieresis/eth/ntilde/ograve/oacute
2952         /ocircumflex/otilde/odieresis/divide/oslash/ugrave
2953         /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
2954     ]
2955 }
2956 \pdf_object_write:nn {__hyp/Font/ZaDb }
2957 {
2958     /Type/Font
2959     /Subtype/Type1
2960     /Name/ZaDb
2961     /BaseFont/ZapfDingbats
2962 }
2963 \pdf_object_write:nx {__hyp/Font/Helv }
2964 {
2965     /Type/Font
2966     /Subtype/Type1
2967     /Name/Helv
2968     /BaseFont/Helvetica
2969     /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
2970 }
2971 \global\let\Hy@FormObjects\relax
2972 }
2973 \ExplSyntaxOff
2974 \providecommand*{\Fld@pageobjref}{\}
2975 \ifcsname pdf@escapestring\endcsname
2976   \def\Hy@escapeform#1{%
2977     \ifHy@pdfescapeform
2978       \let\Hy@escapestring\pdfescapestring
2979     \else
2980       \let\Hy@escapestring\@firstofone
2981     \fi
2982   }%
2983   \Hy@escapeform{}\}
2984 \else
2985   \let\Hy@escapestring\@firstofone
2986   \def\Hy@escapeform#1{%
2987     \ifHy@pdfescapeform
2988       \def\Hy@escapestring##1{%
2989         \noexpand\Hy@escapestring{\noexpand##1}%
2990       }%
2991       \edef\Hy@temp{#1}%
2992       \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{\}\@nil
2993     \def\Hy@escapestring##1{%
2994       \@ifundefined{Hy@esc@\string##1}{%
2995         ##1%
2996         \ThisShouldNotHappen
2997       }{%
2998         \csname Hy@esc@\string##1\endcsname
2999       }%
3000     }%

```

```

3001     \else
3002         \let\Hy@escapestring\@firstofone
3003     \fi
3004 }%
3005 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3006     \ifx\#3\%
3007     \else
3008         \expandafter
3009         \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3010         \ltx@ReturnAfterFi{%
3011             \Hy__hypescapeform#3\@nil
3012         }%
3013     \fi
3014 }%
3015 \fi
3016 \def\PDFForm@Name{%
3017     \PDFForm__hypName\Fld@name
3018     \ifx\Fld@altname\relax
3019     \else
3020         \PDFForm__hypName\Fld@altname
3021     \fi
3022     \ifx\Fld@mappingname\relax
3023     \else
3024         \PDFForm__hypName\Fld@mappingname
3025     \fi
3026 }
3027 \def\PDFForm__hypName#1{%
3028     \begingroup
3029     \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3030         \ifHy@unicode
3031             \Hy@unicodetrue
3032         \fi
3033     \fi
3034     \pdfstringdef\Hy@gtemp#1%
3035     \endgroup
3036     \let#1\Hy@gtemp
3037 }
3038 \def\Fld@X@additionalactions{%
3039     \ifx\Fld@keystroke@code\@empty
3040     \else
3041         /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3042     \fi
3043     \ifx\Fld@format@code\@empty
3044     \else
3045         /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3046     \fi
3047     \ifx\Fld@validate@code\@empty
3048     \else
3049         /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3050     \fi
3051     \ifx\Fld@calculate@code\@empty
3052     \else
3053         /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3054     \fi

```



```

3055 \ifx\Fld@onfocus@code\@empty
3056 \else
3057 /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3058 \fi
3059 \ifx\Fld@onblur@code\@empty
3060 \else
3061 /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3062 \fi
3063 \ifx\Fld@onmousedown@code\@empty
3064 \else
3065 /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3066 \fi
3067 \ifx\Fld@onmouseup@code\@empty
3068 \else
3069 /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3070 \fi
3071 \ifx\Fld@onenter@code\@empty
3072 \else
3073 /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3074 \fi
3075 \ifx\Fld@onexit@code\@empty
3076 \else
3077 /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3078 \fi
3079 }
3080 \ExplSyntaxOn
3081 \def\Fld@additionalactions
3082 {%
3083 \exp_args:N\str_if_eq:nnF {\Fld@X@additionalactions}{\}
3084 {
3085 \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3086 {/AA<<\Fld@X@additionalactions>>}
3087 }
3088 }
3089 \ExplSyntaxOff
3090 \def\Fld@annotnames{%
3091 /T(\Fld@name)%
3092 \ifx\Fld@altname\relax
3093 \else
3094 /TU(\Fld@altname)%
3095 \fi
3096 \ifx\Fld@mappingname\relax
3097 \else
3098 /TM(\Fld@mappingname)%
3099 \fi
3100 }
3101 \ExplSyntaxOn
3102 \def\PDFForm@Check
3103 {
3104 /Subtype/Widget
3105 ~\Fld@annotflags
3106 ~\Fld@pageobjref
3107 ~\Fld@annotnames
3108 /FT/Btn

```

```

3109 \Fld@flags
3110 /Q~\Fld@align
3111 /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3112 /AP
3113 <<
3114 /N
3115 <<
3116 /Yes~\pdfxform_ref:o{\l__hyp_CheckmarkYes_tl}
3117 /Off~\pdfxform_ref:o{\l__hyp_CheckmarkOff_tl}
3118 >>
3119 >>
3120 /MK<<
3121 \int_compare:nNf {\Fld@rotation}={0}
3122 {
3123 /R~\Fld@rotation
3124 }
3125 \tl_if_empty:NF\Fld@bordercolor
3126 {
3127 /BC[\Fld@bordercolor]
3128 }
3129 \tl_if_empty:NF\Fld@bcolor
3130 {
3131 /BG[\Fld@bcolor]
3132 }
3133 /CA(\Hy@escapestring{\Fld@cbsymbol})%
3134 >>
3135 /DA
3136 (
3137 /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3138 \tl_if_empty:NF \Fld@color
3139 {
3140 \c_space_tl \Fld@color
3141 }
3142 )
3143 /H/P
3144 \legacy_if:nTF {\Fld@checked}
3145 {
3146 /V/Yes /AS/Yes
3147 }
3148 {
3149 /V/Off /AS/Off
3150 }
3151 \Fld@additionalactions
3152 }
3153 \ExplSyntaxOff
3154 \ExplSyntaxOn
3155 \def\PDFForm@Push
3156 {
3157 /Subtype/Widget
3158 ~\Fld@annotflags
3159 ~\Fld@pageobjref
3160 ~\Fld@annotnames
3161 /FT/Btn
3162 ~\Fld@flags

```

```

3163 /H/P
3164 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3165 \bool_if:nT
3166 {
3167     !\int_compare_p:nNn {\Fld@rotation} = {0}
3168     ||
3169     \tl_if_exist_p:N \Fld@bordercolor
3170 }
3171 {
3172     /MK
3173     <<
3174     \int_compare:nNf {\Fld@rotation} = {0}
3175     {
3176         /R~\Fld@rotation
3177     }
3178     \tl_if_exist:NT \Fld@bordercolor
3179     {
3180         /BC[\Fld@bordercolor]
3181     }
3182     >>
3183 }
3184 /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3185 \Fld@additionalactions
3186 }
3187
3188 \ExplSyntaxOff
3189 \def\PDFForm@List{%
3190 /Subtype/Widget%
3191 \Fld@annotflags
3192 \Fld@pageobjref
3193 \Fld@annotnames
3194 /FT/Ch%
3195 \Fld@flags
3196 /Q \Fld@align
3197 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3198 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3199     \ifx\Fld@bordercolor\relax\else 1\fi
3200     \ifx\Fld@bcolor\relax \else 1\fi
3201     \space
3202 \else
3203 /MK<<%
3204     \ifnum\Fld@rotation=\z@
3205     \else
3206         /R \Fld@rotation
3207     \fi
3208     \ifx\Fld@bordercolor\relax
3209     \else
3210         /BC[\Fld@bordercolor]%
3211     \fi
3212     \ifx\Fld@bcolor\relax
3213     \else
3214         /BG[\Fld@bcolor]%
3215     \fi
3216 >>%

```

```

3217 \fi
3218 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3219 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3220 \Fld@choices
3221 \Fld@additionalactions
3222 }
3223 \ExplSyntaxOn
3224 \def\PDFForm@Radio
3225 {
3226 /Subtype/Widget
3227 ~\Fld@annotflags
3228 ~\Fld@pageobjref
3229 ~\Fld@annotnames
3230 /FT/Btn
3231 \Fld@flags
3232 /H/P
3233 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3234 /MK<<
3235 \ifnum\Fld@rotation=\z@
3236 \else
3237 /R~\Fld@rotation
3238 \fi
3239 \ifx\Fld@bordercolor\relax
3240 \else
3241 /BC[\Fld@bordercolor]%
3242 \fi
3243 \ifx\Fld@bcolor\relax
3244 \else
3245 /BG[\Fld@bcolor]%
3246 \fi
3247 /CA(\Hy@escapestring{\Fld@radiosymbol})%
3248 >>
3249 /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3250 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3251 \ifx\Fld@default\@empty
3252 /V/Off%
3253 /DV/Off%
3254 \else
3255 /V/\Fld@default
3256 /DV/\Fld@default
3257 \fi
3258 \Fld@additionalactions
3259 }
3260 \ExplSyntaxOff
3261 \ExplSyntaxOn
3262 % Does an appeareance dict make sense here?
3263 \def\PDFForm@Text
3264 {
3265 /Subtype/Widget
3266 ~\Fld@annotflags
3267 ~\Fld@pageobjref
3268 ~\Fld@annotnames
3269 /FT/Tx
3270 ~\Fld@flags

```

```

3271 /Q~\Fld@align
3272 /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3273 \bool_if:nT
3274 {
3275     !\int_compare_p:nNn {\Fld@rotation} = {0}
3276     ||
3277     \tl_if_exist_p:N \Fld@bordercolor
3278     ||
3279     \tl_if_exist_p:N \Fld@bcolor
3280 }
3281 {
3282     /MK
3283     <<
3284     \int_compare:nNnF {\Fld@rotation} = {0}
3285     {
3286         /R~\Fld@rotation
3287     }
3288     \tl_if_exist:NT \Fld@bordercolor
3289     {
3290         /BC[\Fld@bordercolor]
3291     }
3292     \tl_if_exist:NT \Fld@bcolor
3293     {
3294         /BG[\Fld@bcolor]
3295     }
3296     >>
3297 }
3298 /DA
3299 (
3300     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3301     \tl_if_empty:NF {\c_space_tl\Fld@color}
3302 )
3303 /DV(\Hy@escapestring{\Fld@default})
3304 /V(\Hy@escapestring{\Fld@value})
3305 ~\Fld@additionalactions
3306 \int_compare:nNnT { \Fld@maxlen}>{0}
3307 {
3308     /MaxLen~\Fld@maxlen
3309 }
3310 }
3311 \ExplSyntaxOff
3312
3313 \def\PDFForm@Submit{%
3314     /Subtype/Widget%
3315     \Fld@annotflags
3316     \Fld@pageobjref
3317     \Fld@annotnames
3318     /FT/Btn%
3319     \Fld@flags
3320     /H/P%
3321     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3322     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3323     \ifx\Fld@bordercolor\relax\else 1\fi
3324     \space

```

```

3325 \else
3326 /MK<<%
3327 \ifnum\Fld@rotation=\z@
3328 \else
3329 /R \Fld@rotation
3330 \fi
3331 \ifx\Fld@bordercolor\relax
3332 \else
3333 /BC[\Fld@bordercolor]%
3334 \fi
3335 >>%
3336 \fi
3337 /A<<%
3338 /S/SubmitForm%
3339 /F<<%
3340 /FS/URL%
3341 /F(\Hy@escapestring{\Form@action})%
3342 >>%
3343 \Fld@submitflags
3344 >>%
3345 \Fld@additionalactions
3346 }
3347 \ExplSyntaxOn
3348 \def\PDFForm@Reset{%
3349 /Subtype/Widget%
3350 \Fld@annotflags
3351 \Fld@pageobjref
3352 \Fld@annotnames
3353 /FT/Btn%
3354 \Fld@flags
3355 /H/P%
3356 /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3357 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3358 \ifx\Fld@bordercolor\relax\else 1\fi
3359 \space
3360 \else
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 >>%
3371 \fi
3372 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3373 /A<</S/ResetForm>>%
3374 \Fld@additionalactions
3375 }%
3376
3377
3378 %these patterns are used in hyperref checks.

```

```

3379 %it is unclear if they are really useful and if a backend support is
3380 %needed.
3381 \str_case:VnF \c_sys_backend_str
3382 {
3383   { pdfmode }
3384   {
3385     \def\HyPat@ObjRef
3386     {
3387       [0-9]*[1-9][0-9]*~0~R
3388     }
3389   }
3390   { dvipdfmx }
3391   {
3392     \def\HyPat@ObjRef
3393     {
3394       @[\~]+
3395     }
3396   }
3397   { xdvipdfmx }
3398   {
3399     \def\HyPat@ObjRef
3400     {
3401       @[\~]+
3402     }
3403   }
3404 }
3405 { %also set in hyperref sty, so probably not needed.
3406   \def\HyPat@ObjRef/{.+}
3407 }
3408
3409
3410 \ExplSyntaxOff
3411 % UF: removed Hy@writebookmark
3412 %   \Hy@currentbookmarklevel{0}
3413 %   \Hy@numberline
3414 %   \__hypwritetorep
3415 %   counter{bookmark@seq@number}
3416 % removed \HyPsd@SanitizeForOutFile, not needed
3417 % removed \currentpdfbookmark, defined by bookmark,
3418 % should use \newcommand there
3419 % removed \subpdfbookmark, defined by bookmark,
3420 % should use \newcommand there
3421 % removed \belowpdfbookmark, defined by bookmark,
3422 % should use \newcommand there
3423 % removed \pdfbookmark, defined by bookmark,
3424 % \BOOKMARK
3425 % \@BOOKMARK
3426 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3427 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3428 %% removed \ReadBookmarks / unneeded with bookmark.
3429 %% removed \Hy@OutlineName
3430 %% removed \check@bm@number
3431 %% removed \calc@bm@number
3432

```

```

3433 \ifHy@implicit
3434 \else
3435   \expandafter\endinput
3436 \fi
3437 \newlength\Hy@SectionHShift
3438 \def\Hy@SectionAnchorHref#1{%
3439   \ifx\protect\@typeset@protect
3440     \Hy__hypSectionAnchor{#1}%
3441   \fi
3442 }
3443 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%
3444   \leavevmode
3445   \hbox to Opt{%
3446     \kern-\Hy@SectionHShift
3447     \Hy@raisedlink{%
3448       \hyper@anchorstart{#1}\hyper@anchorend
3449     }%
3450     \hss
3451   }%
3452 }
3453 \let\H@old@ssect\@ssect
3454 \def\@ssect#1#2#3#4#5{%
3455   \Hy@MakeCurrentHrefAuto{section*}%
3456   \setlength{\Hy@SectionHShift}{#1}%
3457   \begingroup
3458     \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3459     \toks\tw@{\expandafter{%
3460       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3461       #5%
3462     }}%
3463   \edef\x{\endgroup
3464     \the\toks@{\the\toks\tw@}%
3465   }\x
3466 }
3467 \let\H@old@schapter\@schapter
3468 \def\@schapter#1{%
3469   \begingroup
3470     \let\@mkboth\@gobbletwo
3471     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3472     \Hy@raisedlink{%
3473       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3474     }%
3475   \endgroup
3476   \H@old@schapter{#1}%
3477 }
3478 \ltx@ifundefined{chapter}{-}{%
3479   \let\Hy@org@chapter\@chapter
3480   \def\@chapter{%
3481     \def\Hy@next{%
3482       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3483       \Hy@raisedlink{%
3484         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3485       }%
3486     }%

```



```

3487 \ifnum\c@secnumdepth>\m@ne
3488 \ltx@ifundefined{if@mainmatter}%
3489 \iftrue{\csname if@mainmatter\endcsname}%
3490 \let\Hy@next\relax
3491 \fi
3492 \fi
3493 \Hy@next
3494 \Hy@org@chapter
3495 }%
3496 }
3497 \let\H@old@part\@part
3498 \begingroup\expandafter\expandafter\expandafter\endgroup
3499 \expandafter\ifx\csname chapter\endcsname\relax
3500 \let\Hy@secnum@part\z@
3501 \else
3502 \let\Hy@secnum@part\m@ne
3503 \fi
3504 \def\@part{%
3505 \ifnum\Hy@secnum@part>\c@secnumdepth
3506 \phantomsection
3507 \fi
3508 \H@old@part
3509 }
3510 \let\H@old@spart\@spart
3511 \def\@spart#1{%
3512 \Hy@MakeCurrentHrefAuto{part*}%
3513 \Hy@raisedlink{%
3514 \hyper@anchorstart{\@currentHref}\hyper@anchorend
3515 }%
3516 \H@old@spart{#1}%
3517 }
3518 \let\H@old@sect\@sect
3519 \def\@sect#1#2#3#4#5#6[#7]#8{%
3520 \ifnum #2>\c@secnumdepth
3521 \expandafter\@firstoftwo
3522 \else
3523 \expandafter\@secondoftwo
3524 \fi
3525 {%
3526 \Hy@MakeCurrentHrefAuto{section*}%
3527 \setlength{\Hy@SectionHShift}{#3}%
3528 \begingroup
3529 \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[\@currentHref]}%
3530 \toks\tw@\expandafter{%
3531 \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3532 #8%
3533 }%
3534 \edef\x{\endgroup
3535 \the\toks@\the\toks\tw@}%
3536 }\x
3537 }{%
3538 \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[\@currentHref]{#8}%
3539 }%
3540 }

```

```

3541 \expandafter\def\csname Parent-4\endcsname{}
3542 \expandafter\def\csname Parent-3\endcsname{}
3543 \expandafter\def\csname Parent-2\endcsname{}
3544 \expandafter\def\csname Parent-1\endcsname{}
3545 \expandafter\def\csname Parent0\endcsname{}
3546 \expandafter\def\csname Parent1\endcsname{}
3547 \expandafter\def\csname Parent2\endcsname{}
3548 \expandafter\def\csname Parent3\endcsname{}
3549 \expandafter\def\csname Parent4\endcsname{}
3550 %%
3551 %% End of file 'hgeneric-testphase.def'.
3552 \</package>

3553 \*colorscheme>
3554 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3555 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3556 % linkcolor=
3557 %,filecolor=
3558 %,urlcolor=
3559 %,menucolor=
3560 %,runcolor=
3561 %,linkbordercolor=
3562 %,filebordercolor=
3563 %,urlbordercolor=
3564 %,menubordercolor=
3565 %,runbordercolor=
3566
3567 \prop_const_from_keyval:Nn \c__hyp_colorscheme_original_prop
3568 {
3569     linkcolor      = [rgb]{1,0,0}, %red
3570     filecolor      = [rgb]{0,1,1}, %cyan
3571     urlcolor       = [rgb]{1,0,1}, %magenta
3572     menucolor      = [rgb]{1, 0, 0}, %red
3573     runcolor       = [rgb]{0,1,1}, %cyan
3574     %-----
3575     linkbordercolor = [rgb]{1, 0 ,0 },
3576     filebordercolor = [rgb]{0, .5, .5},
3577     urlbordercolor  = [rgb]{0, 1, 1},
3578     menubordercolor = [rgb]{1, 0, 0},
3579     runbordercolor  = [rgb]{0, .7, .7}
3580 }
3581
3582 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3583 {
3584     linkcolor      = [rgb]{0,0.2,0.6},
3585     filecolor      = [rgb]{0.8,0,0.8},
3586     urlcolor       = [rgb]{0.8,0,0.8},
3587     menucolor      = [rgb]{0,0.2,0.6},
3588     runcolor       = [rgb]{0.8,0,0.8},
3589     %----- %-----
3590     linkbordercolor = [rgb]{0,0.2,0.6},
3591     filebordercolor = [rgb]{0.8,0,0.8},
3592     urlbordercolor  = [rgb]{0.8,0,0.8},
3593     menubordercolor = [rgb]{0,0.2,0.6},
3594     runbordercolor  = [rgb]{0.8,0,0.8}

```

```

3595 }
3596
3597 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3598 { %two colors: intern/extern
3599   linkcolor      = [rgb]{0.79216, 0, 0.12549},
3600   filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3601   urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3602   menucolor      = [rgb]{0.79216, 0, 0.12549 },
3603   runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3604   %----- %-----
3605   linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3606   filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3607   urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3608   menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3609   runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3610 }
3611
3612 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3613 { %all darkgray
3614   linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3615   filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3616   urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3617   menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3618   runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3619   %----- %-----
3620   linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3621   filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3622   urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3623   menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3624   runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3625 }
3626
3627 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3628 { %dvipsnam.def
3629   linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3630   filecolor      = [rgb]{1, 0, 0}, %Red
3631   urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3632   menucolor      = [rgb]{1, 0, 0}, %Red
3633   runcolor       = [rgb]{1, 0, 0}, %Red
3634   %----- %-----
3635   linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3636   filebordercolor = [rgb]{1, 0, 0}, %Red
3637   urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3638   menubordercolor = [rgb]{1, 0, 0}, %Red
3639   runbordercolor  = [rgb]{1, 0, 0} %Red
3640 }
3641
3642 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3643 { %dvipsnam.def
3644   linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3645   filecolor      = [rgb]{0, 1, 0}, %Green
3646   urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3647   menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3648   runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry

```

```

3649 %----- %-----
3650 linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3651 filebordercolor = [rgb]{0, 1, 0}, %Green
3652 urlbordercolor = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3653 menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3654 runbordercolor = [rgb]{0.64, 0.08, 0.98} %Mulberry
3655 }
3656
3657
3658 \prop_const_from_keyval:Nn \c_hyp_colorscheme_phelype_prop
3659 {
3660 linkcolor = [rgb]{0.50196, 0, 0.02353},
3661 filecolor = [rgb]{0.07451, 0.09412, 0.46667},
3662 urlcolor = [rgb]{0.54118, 0, 0.52941},
3663 menucolor = [rgb]{0.44706, 0.45882, 0},
3664 runcolor = [rgb]{0.07451, 0.46667, 0.46275},
3665 %----- %-----
3666 linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3667 filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3668 urlbordercolor = [rgb]{0.724708, 0.4, 0.717646},
3669 menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3670 runbordercolor = [rgb]{0.444706, 0.680002, 0.67765}
3671 }
3672
3673 \prop_const_from_keyval:Nn \c_hyp_colorscheme_henryford_prop
3674 {
3675 linkcolor = [rgb]{0,0,0},
3676 filecolor = [rgb]{0,0,0},
3677 urlcolor = [rgb]{0,0,0},
3678 menucolor = [rgb]{0,0,0},
3679 runcolor = [rgb]{0,0,0},
3680 %----- %-----
3681 linkbordercolor = [rgb]{0,0,0},
3682 filebordercolor = [rgb]{0,0,0},
3683 urlbordercolor = [rgb]{0,0,0},
3684 menubordercolor = [rgb]{0,0,0},
3685 runbordercolor = [rgb]{0,0,0}
3686 }
3687 </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 | |
|-----------------------------------|--|
| \# | 254, 675 |
| \\$ | 253 |
| \% | 676 |
| \. | 493, 497, 499 |
| @curropt commands: | |
| \@curropt: | 2649, 2717 |
| \\ | 17, 18, 19, 26, 30, 40, 60, 67, 74, 81, 88, 103, 108, 109, 117, 118, 127, 128, 129, 130, 131, 138, 255, 256, 805, 2651, 2719, 3006 |
| _ | 491, 493, 497, 499 |
| A | |
| \A | 491 |
| \Acrobatmenu | 17, 155 |
| \addcontentsline | 12 |
| \AddToDocumentProperties | 367 |
| \AddToHook | 392 |
| \advance | 2582, 2656, 2700, 2701 |
| allcolor (hypersetup key) | 934 |
| \author | 2 |
| B | |
| \b | 495 |
| \begingroup | 183, 252, 354, 2425, 2616, 2645, 2657, 2691, 3028, 3457, 3469, 3498, 3528 |
| \belowpdfbookmark | 3421 |
| \bgroup | 249, 354 |
| \BOOKMARK | 3424 |
| bookmarkstype (hypersetup key) | 12 |
| bool commands: | |
| \bool_if:N | 242, 272, 334, 624, 639, 650, 663, 671, 701, 759, 816, 904, 914, 1154, 1165 |
| \bool_if:nTF | 771, 1861, 1886, 3165, 3273 |
| \bool_lazy_and:nnTF | 387 |
| \bool_lazy_or:nnTF | 1184, 1233 |
| \bool_new:N | 177, 178, 477, 481, 485 |
| \bool_set_true:N | 486 |
| bordercolormodel (hypersetup key) | 12, 955 |
| box commands: | |
| \box_dp:N | 567, 2788, 2827, 2862 |
| \box_ht:N | 566, 2787, 2826, 2861 |
| \box_new:N | 426, 488 |
| \box_set_dp:Nn | 2058 |
| \box_set_ht:Nn | 2065 |
| \box_use:N | 1171, 2790, 2836, 2865 |
| \box_use_drop:N | 1176 |
| \box_wd:N | 565, 2786, 2825, 2860 |
| \l_tmpa_box | 2784, 2786, 2787, 2788, 2790, 2823, 2825, 2826, 2827, 2836, 2858, 2860, 2861, 2862, 2865 |
| C | |
| \catcode | 253, 254 |
| \char | 2521, 2533, 2545, 2557 |
| \chardef | 136 |
| \cite | 31 |
| clist commands: | |
| \clist_map_function:nN | 110, 119 |
| \clist_map_inline:nn | 2222 |
| \clist_map_variable:nNn | 2648, 2716 |
| color commands: | |
| \color_export:nnN | 854 |
| \color_export:nnnN | 860 |
| \color_select:n | 871, 907, 1175 |
| \color_select:nn | 877 |
| \color_set:nn | 2, 372, 888 |
| \color_set:nnn | 2, 371, 894 |
| color names: | |
| hyp/annot/file | 474 |
| hyp/annot/link | 474 |
| hyp/annot/menu | 474 |
| hyp/annot/run | 474 |
| hyp/annot/url | 474 |
| colorfile (hypersetup key) | 934 |
| colorlink (hypersetup key) | 934 |
| colorlinks (hypersetup key) | 920 |
| colormenu (hypersetup key) | 934 |
| colorrn (hypersetup key) | 934 |
| colorscheme (hypersetup key) | 1, 1342 |
| colorurl (hypersetup key) | 934 |
| cs commands: | |
| \cs_generate_variant:Nn | 134, 135, 369, 405, 420, 425, 529, 863, 880, 897 |
| \cs_gset:Npn | 1146 |
| \cs_if_exist_p:N | 388 |
| \cs_new:Npn | 103, 421 |
| \cs_new_protected:Npn | 183, 281, 299, 317, 344, 364, 399, 408, 505, 509, 513, 517, 530, 532, 536, 578, 584, 591, 595, 617, 622, 637, 648, 661, 699, 757, 814, 847, 858, 864, 875, 881, 892, 1090, 2093, 2149 |

| | | | |
|--|---|------------------------------------|--|
| <code>\cs_set_eq:NN</code> | 139, 148, 149, 189, 231, 675, 676, 1419 | <code>\EdefEscapeName</code> | 2715, 2720 |
| <code>\cs_set_protected:Npn</code> | 137, 162, 546, 805 | <code>\egroup</code> | 357 |
| <code>\csname</code> | 2663, 2998, 3009, 3489, 3499, 3541, 3542, 3543, 3544, 3545, 3546, 3547, 3548, 3549 | <code>\else</code> | 2424, 2612, 2666, 2670, 2673, 2690, 2698, 2979, 2984, 3001, 3007, 3019, 3023, 3040, 3044, 3048, 3052, 3056, 3060, 3064, 3068, 3072, 3076, 3093, 3097, 3198, 3199, 3200, 3202, 3205, 3209, 3213, 3219, 3236, 3240, 3244, 3250, 3254, 3322, 3323, 3325, 3328, 3332, 3357, 3358, 3360, 3363, 3367, 3434, 3501, 3522 |
| <code>\currentpdfbookmark</code> | 3417 | <code>\endcsname</code> | 2677, 2975, 2998, 3009, 3489, 3499, 3541, 3542, 3543, 3544, 3545, 3546, 3547, 3548, 3549 |
| D | | | |
| <code>\d</code> | 493, 497, 499 | <code>\endgroup</code> | 258, 261, 355, 357, 2428, 2636, 2659, 2705, 2708, 3035, 3463, 3475, 3498, 3534 |
| <code>debug</code> (hypersetup key) | 1366 | <code>\endinput</code> | 3435 |
| <code>\DeclareDocumentMetadata</code> | 3, 10, 13, 14, 17, 19, 27, 33 | exp commands: | |
| <code>\DeclareRobustCommand</code> | 236, 265, 289, 307, 327, 357, 3443 | <code>\exp_args:Ne</code> | 3083 |
| <code>\def</code> | 143, 354, 355, 357, 541, 2388, 2392, 2399, 2417, 2446, 2463, 2580, 2589, 2604, 2605, 2608, 2609, 2622, 2623, 2639, 2640, 2643, 2644, 2682, 2712, 2750, 2764, 2765, 2777, 2805, 2807, 2808, 2818, 2841, 2843, 2844, 2855, 2874, 2876, 2877, 2879, 2880, 2889, 2909, 2976, 2986, 2988, 2993, 3005, 3016, 3027, 3038, 3081, 3090, 3102, 3155, 3189, 3224, 3263, 3313, 3348, 3385, 3392, 3399, 3406, 3438, 3454, 3468, 3480, 3481, 3504, 3511, 3519, 3541, 3542, 3543, 3544, 3545, 3546, 3547, 3548, 3549 | <code>\exp_args:Nne</code> | 2187 |
| <code>\DefaultHeightofCheckBox</code> | 2880 | <code>\exp_args:Nnno</code> | 301, 319 |
| <code>\DefaultHeightofChoiceMenu</code> | 2644 | <code>\exp_args:Nno</code> | 283, 346, 2135 |
| <code>\DefaultHeightofReset</code> | 2844 | <code>\exp_args:Nnox</code> | 709 |
| <code>\DefaultHeightofSubmit</code> | 2808 | <code>\exp_args:NNV</code> | 1846, 1877 |
| <code>\DefaultHeightofText</code> | 2613 | <code>\exp_args:NnV</code> | 580, 587 |
| <code>\DefaultHeightofTextMultiline</code> | 2611 | <code>\exp_args:Nnx</code> | 2370 |
| <code>\DefaultOptionsofCheckBox</code> | 2883 | <code>\exp_args:No</code> | 609, 613, 2767, 2810, 2846, 2881 |
| <code>\DefaultOptionsofPushButton</code> | 2769 | <code>\exp_args:NV</code> | 563, 570 |
| <code>\DefaultOptionsofReset</code> | 2848 | <code>\exp_args:Nx</code> | 705, 2128, 2176, 2419 |
| <code>\DefaultOptionsofSubmit</code> | 2812 | <code>\exp_not:N</code> | 805 |
| <code>\DefaultOptionsofText</code> | 2618 | <code>\exp_not:n</code> | 1281, 1315, 1426, 1486, 1507, 1544, 1564, 1587, 1631, 1664, 1689, 1740, 1765, 1792, 1841, 1918, 1946, 1973, 1993, 2208, 2315, 2343, 2353, 2364 |
| <code>\DefaultWidthofCheckBox</code> | 2879 | <code>\expandafter</code> | 255, 2617, 2651, 2660, 2662, 2663, 2719, 2992, 3008, 3435, 3459, 3460, 3498, 3499, 3521, 3523, 3530, 3531, 3541, 3542, 3543, 3544, 3545, 3546, 3547, 3548, 3549 |
| <code>\DefaultWidthofChoiceMenu</code> | 2643 | <code>\ExplSyntaxOff</code> | 9, 2456, 2572, 2599, 2905, 2973, 3089, 3153, 3188, 3260, 3311, 3410 |
| <code>\DefaultWidthofReset</code> | 2843 | <code>\ExplSyntaxOn</code> | 7, 10, 2457, 2577, 2600, 2908, 3080, 3101, 3154, 3223, 3261, 3347 |
| <code>\DefaultWidthofSubmit</code> | 2807 | extension (hypersetup key) | 12, 1388 |
| <code>\DefaultWidthofText</code> | 2608 | F | |
| <code>destlabel</code> (hypersetup key) | 12 | <code>\fbox</code> | 2563, 2568, 2741 |
| dim commands: | | <code>\fi</code> | 2413, 2440, 2443, 2452, 2614, 2622, 2623, |
| <code>\dim_to_decimal_in_bp:n</code> | 11, 139 | | |
| <code>\do</code> | 2649, 2717 | | |
| <code>\documentclass</code> | 20 | | |
| <code>draft</code> (hypersetup key) | 1366 | | |
| E | | | |
| <code>\edef</code> | 2427, 2659, 2695, 2702, 2991, 3463, 3534 | | |

| | |
|---|--|
| __hyp_color_export_aux:wnnN . . . | \l__hyp_href_url_encode_bool . . . |
| 847, 851, 858 | 177, 188, 226, 242, 272, 334 |
| __hyp_color_select:n 37, 864, 864, 880 | __hyp_href_url_format |
| __hyp_color_select:nN 37 | 183, 189, 231, 347 |
| __hyp_color_select_aux:wn | \l__hyp_href_url_ismap_bool |
| 864, 868, 875 | 178, 202, 671 |
| __hyp_color_set:nN 37 | \l__hyp_href_url_protocol_tl |
| __hyp_color_set:nn | 179, 190, 229, 283, 348 |
| 37, 375, 377, 881, 881, 897, 939 | __hyp_link_goto_begin:nw |
| __hyp_color_set_aux:nwn 881, 885, 892 | 595, 629, 644 |
| \c__hyp_colorscheme_daleif_prop 3582 | __hyp_link_goto_end: . 617, 630, 652 |
| \c__hyp_colorscheme_henryford_- | \c__hyp_map_annot_hyp_prop 438 |
| prop 3673 | \c__hyp_map_hyp_annot_prop |
| \c__hyp_colorscheme_julian_prop 3597 | 438, 898, |
| \c__hyp_colorscheme_original_- | 962, 998, 1025, 1033, 1044, 1071, |
| prop 3567 | 1079, 1148, 1256, 1292, 1302, 1434 |
| \c__hyp_colorscheme_phelype_prop | __hyp_ocg_init: 1090, 1090, 1146, 1156 |
| 3658 | l__hyp_page/Trans 502 |
| \c__hyp_colorscheme_szabolcsA_- | __hyp_PageLabels_gpush: 536, 536, 549 |
| prop 3627 | \l__hyp_para_tmpa_tl 434, 770, 773, 782 |
| \c__hyp_colorscheme_szabolcsB_- | \l__hyp_para_tmpa_tl\l__hyp_- |
| prop 3642 | text_tmpa_str\g__hyp_text_- |
| \c__hyp_colorscheme_tivv_prop . 3612 | tmpa_str 431 |
| \l__hyp_dest_box 29, | \l__hyp_RadioYes_tl 2710, 2711 |
| 488, 565, 566, 567, 2042, 2058, 2065 | __hyp_ref_check:nn 398, 408, 420, 2587 |
| \l__hyp_dest_name_tmpa_tl . . 431, | __hyp_ref_if_exist:nn . 398, 406, 407 |
| 598, 599, 605, 609, 611, 614, 722, 735 | __hyp_ref_if_exist:nnTF . . 410, 2591 |
| \l__hyp_dest_pdfremotestartview_- | __hyp_ref_label:nn 398, 399, 405, 2586 |
| tl 457, 730, 1849, 1853 | __hyp_ref_value:nn 398, 421, 425, 2595 |
| \g__hyp_dest_pdfstartpage_tl . . . | __hyp_secondoftwoewithopt:wnn . . . |
| ... 457, 1860, 1862, 1869, 1887, 1894 | 360, 361, 362, 363 |
| \g__hyp_dest_pdfstartview_tl . . . | __hyp_setup_info_date_key:nn . . . |
| 457, 1862, 1869, 1880, 1884, 1887, 1894 | 2149, 2181, 2182 |
| \l__hyp_dest_pdfview_tl | __hyp_setup_info_key:nn . . 2093, |
| 473, 581, 588, | 2142, 2143, 2144, 2145, 2147, 2148 |
| 2011, 2014, 2021, 2024, 2025, 2026, | __hyp_store_metadata:nn |
| 2027, 2028, 2029, 2034, 2038, 2074 | 364, 369, 2090, |
| \c__hyp_dest_startview_regex . . . | 2110, 2128, 2162, 2176, 2197, 2269 |
| 489, 1847, 1878 | __hyp_text_cleanup:N . 509, 509, 523 |
| \c__hyp_dest_undefined_tl | \l__hyp_text_enc_dest_print_tl . . . |
| 437, 604, 605 | 460, 721 |
| __hyp_destination:nn | \l__hyp_text_enc_dest_tl |
| 29, 552, 552, 581, 588 | 29, 460, 559, 610 |
| \l__hyp_filename_tmpa_tl . . . 431, | \l__hyp_text_enc_file_print_tl . . . |
| 704, 706, 712, 713, 718, 765, 766, 782 | 460, 764 |
| __hyp_href_pdf_aux:nn 296, 299 | \l__hyp_text_enc_info_print_tl . . . |
| \l__hyp_href_pdf_destination_tl . . . | 460, 534 |
| 180, 191, 230, 301 | \l__hyp_text_enc_para_print_tl . . . |
| \l__hyp_href_pdf_page_tl 181, 199, 729 | 460, 769 |
| __hyp_href_run_aux:nn 314, 317 | \l__hyp_text_enc_uri_print_tl . . . |
| \l__hyp_href_run_parameter_tl | 244, |
| 182, 203, 319 | 247, 274, 277, 336, 339, 460, 668, 1445 |
| __hyp_href_url_aux:n 341, 344 | |
| __hyp_href_url_aux:nn 279, 281 | |

| | |
|--|----------|
| _hyp_text_pdfstring:nnN | 1318 |
| 517, 517, 529, 531, | 1318 |
| 534, 557, 608, 666, 719, 762, 767, 1445 | 1318 |
| _hyp_text_pdfstring_info:nn | 1318 |
| 532, 532, 2104, 2122, 2132 | 1388 |
| _hyp_text_purify:nn . 505, 505, 522 | 9, 1434 |
| _hyp_text_string_from_unicode:nn | 13 |
| 513, 513, 524 | 13, 1044 |
| \g_hyp_text_tmpa_str . 436, 525, 527 | 934 |
| \l_hyp_text_tmpa_str | 1388 |
| 435, 522, 523, 524, 525 | 1388 |
| \l_hyp_tmpa_box 426, 1158, 1171, 1176 | 1388 |
| \l_hyp_tmpa_int | 1388 |
| 426, 1847, | 1388 |
| 1849, 1878, 1880, 2001, 2002, 2006, | 9, 1434 |
| 2008, 2032, 2039, 2040, 2041, 2050, | 13, 1044 |
| 2052, 2062, 2067, 2479, 2481, 2487 | 934 |
| \l_hyp_tmpa_str | 1388 |
| 426, 2104, 2105, 2107, | 13 |
| 2122, 2123, 2125, 2132, 2133, 2137 | 1184 |
| \l_hyp_tmpa_tl 426, 560, 564, 572, | 1184 |
| 975, 979, 1445, 1446, 1451, 1845, | 1184 |
| 1847, 1876, 1878, 2008, 2009, 2016 | 1184 |
| \l_hyp_uri_tmpa_tl ... 431, 669, 670 | 1184 |
| hyp/annot/file (color name) | 1184 |
| hyp/annot/link (color name) | 1388 |
| hyp/annot/menu (color name) | 2093 |
| hyp/annot/run (color name) | 13 |
| hyp/annot/url (color name) | 13, 1044 |
| hyp/text/pdfstring | 13, 2149 |
| \hypercalcbp | 2093 |
| 11, 17, 139 | 1356 |
| \HyperDestNameFilter | 2211 |
| 558, 609 | 2093 |
| \hypersetup 1, 2, 5, 8, 9, 12, 17, 64, 97, 162 | 13, 2079 |
| \hypersetup keys: | 13 |
| allcolor | 13, 2149 |
| 934 | 2093 |
| bookmarkstype | 11 |
| 12 | 11 |
| bordercolormodel | 2093 |
| 12, 955 | 2093 |
| colorfile | 2093 |
| 934 | 2093 |
| colorlink | 2093 |
| 934 | 2093 |
| colorlinks | 2093 |
| 920 | 2093 |
| colormenu | 2093 |
| 934 | 2093 |
| colorrund | 2093 |
| 934 | 2093 |
| colorscheme | 2093 |
| 1, 1342 | 2093 |
| colorurl | 2093 |
| 934 | 2093 |
| debug | 2093 |
| 1366 | 2093 |
| destlabel | 2093 |
| 12 | 2093 |
| draft | 2093 |
| 1366 | 2093 |
| extension | 2093 |
| 12, 1388 | 2093 |
| file | 2093 |
| 9, 1434 | 2093 |
| fileborderstyle | 2093 |
| 13, 1044 | 2093 |
| filecolor | 2093 |
| 934 | 2093 |
| final | 2093 |
| 1366 | 2093 |
| hidefile | 2093 |
| 1318 | 2093 |
| hidelink | 2093 |
| 1318 | 2093 |
| hidelinks | 2093 |
| 1318 | 2093 |
| hidemenu | 2093 |
| 1318 | 2093 |
| hiderun | 2093 |
| 1318 | 2093 |
| hideurl | 2093 |
| 1318 | 2093 |
| hypertextnames | 2093 |
| 1388 | 2093 |
| link | 2093 |
| 9, 1434 | 2093 |
| linkborder | 2093 |
| 13 | 2093 |
| linkborderstyle | 2093 |
| 13, 1044 | 2093 |
| linkcolor | 2093 |
| 934 | 2093 |
| linkfileprefix | 2093 |
| 1388 | 2093 |
| linktoc | 2093 |
| 1388 | 2093 |
| linktocpage | 2093 |
| 1388 | 2093 |
| localanchorname | 2093 |
| 1388 | 2093 |
| menu | 2093 |
| 9, 1434 | 2093 |
| menuborder | 2093 |
| 13, 1044 | 2093 |
| menuborderstyle | 2093 |
| 13, 1044 | 2093 |
| menucolor | 2093 |
| 934 | 2093 |
| naturalnames | 2093 |
| 1388 | 2093 |
| nesting | 2093 |
| 13 | 2093 |
| ocgcolorfile | 2093 |
| 1184 | 2093 |
| ocgcolorlink | 2093 |
| 1184 | 2093 |
| ocgcolorlinks | 2093 |
| 1184 | 2093 |
| ocgcolormenu | 2093 |
| 1184 | 2093 |
| ocgcolorrun | 2093 |
| 1184 | 2093 |
| ocgcolorurl | 2093 |
| 1184 | 2093 |
| pageanchor | 2093 |
| 1388 | 2093 |
| pdfauthor | 2093 |
| 2093 | 2093 |
| pdfborder | 2093 |
| 13 | 2093 |
| pdfborderstyle | 2093 |
| 13, 1044 | 2093 |
| pdfcreationdate | 2093 |
| 13, 2149 | 2093 |
| pdfcreator | 2093 |
| 2093 | 2093 |
| pdfencoding | 2093 |
| 1356 | 2093 |
| pdfinfo | 2093 |
| 2211 | 2093 |
| pdfkeywords | 2093 |
| 2093 | 2093 |
| pdflang | 2093 |
| 13, 2079 | 2093 |
| pdflinkmargin | 2093 |
| 13 | 2093 |
| pdfmoddate | 2093 |
| 13, 2149 | 2093 |
| pdfproducer | 2093 |
| 2093 | 2093 |
| pdfremotestartview | 2093 |
| 11 | 2093 |
| pdfstartview | 2093 |
| 11 | 2093 |
| pdfsubject | 2093 |
| 2093 | 2093 |
| pdftitle | 2093 |
| 2093 | 2093 |
| pdftrapped | 2093 |
| 2183 | 2093 |
| pdfversion | 2093 |
| 1356 | 2093 |
| pdfview | 2093 |
| 11, 1997 | 2093 |
| plainpages | 2093 |
| 1388 | 2093 |
| run | 2093 |
| 9, 1434 | 2093 |
| runborder | 2093 |
| 13 | 2093 |
| runborderstyle | 2093 |
| 13, 1044 | 2093 |
| runcolor | 2093 |
| 934 | 2093 |
| unicode | 2093 |
| 1356 | 2093 |
| url | 2093 |
| 9, 1434 | 2093 |
| urlborder | 2093 |
| 13 | 2093 |
| urlborderstyle | 2093 |
| 13, 1044 | 2093 |

| | | | |
|----------------------------------|--|----------------------------------|---|
| urlcolor | 934 | \LayoutCheckField | 2891 |
| verbose | 1366 | \LayoutChoiceField | 2686 |
| hypertextnames (hypersetup key) | 1388 | \LayoutPushButtonField | 2779, 2796 |
| hypListbox internal commands: | | \LayoutTextField | 2624 |
| _hypListbox | 2704, 2750 | \leavevmode | 2625, 2723, 3444 |
| hypRadio internal commands: | | legacy commands: | |
| _hypRadio | 2689, 2712 | _legacy_if:nTF | 150, 390, 394, 544, 2504, 2775, 2816, 2854, 2887, 3144 |
| hypwritetorep internal commands: | | \let | 146, 241, 271, 295, 313, 333, 361, 362, 363, 634, 684, 691, 745, 751, 795, 801, 844, 2383, 2384, 2386, 2387, 2414, 2465, 2573, 2574, 2575, 2606, 2607, 2641, 2642, 2684, 2971, 2978, 2980, 2985, 3002, 3036, 3453, 3467, 3470, 3479, 3490, 3497, 3500, 3502, 3510, 3518 |
| _hypwritetorep | 3414 | link (hypersetup key) | 9, 1434 |
| I | | | |
| \ifcase | 3198, 3322, 3357 | linkborder (hypersetup key) | 13 |
| \ifcsname | 2975 | linkborderstyle (hypersetup key) | 13, 1044 |
| \ifdim | 2654, 2693, 2694 | linkcolor (hypersetup key) | 934 |
| \ifnum | 3029, 3198, 3204, 3235, 3322, 3327, 3357, 3362, 3487, 3505, 3520 | linkfileprefix (hypersetup key) | 1388 |
| \iftrue | 3489 | linktoc (hypersetup key) | 1388 |
| \ifx | 2423, 2623, 2683, 3006, 3018, 3022, 3039, 3043, 3047, 3051, 3055, 3059, 3063, 3067, 3071, 3075, 3092, 3096, 3199, 3200, 3208, 3212, 3219, 3239, 3243, 3250, 3251, 3323, 3331, 3358, 3366, 3439, 3499 | linktocpage (hypersetup key) | 1388 |
| \immediate | 2407, 2410 | localanchorname (hypersetup key) | 1388 |
| int commands: | | M | |
| _int_compare:nNnTF | 2006, 2032, 2483, 2742, 3121, 3174, 3284, 3306 | \MakeButtonField | 2784, 2799, 2823, 2858, 2869 |
| _int_compare_p:nNn | 3167, 3275 | \MakeChoiceField | 2759 |
| _int_eval:n | 728 | \MakeFieldObject | 2374, 2513, 2525, 2537, 2549, 2561, 2566 |
| _int_max:nn | 729 | \MakeRadioField | 2741 |
| _int_new:N | 429 | \MakeTextField | 2633 |
| iow commands: | | \mbox | 1168 |
| _iow_newline: | 2924, 2930, 2943 | menu (hypersetup key) | 9, 1434 |
| K | | | |
| \kern | 3446 | menuborder (hypersetup key) | 13 |
| keys commands: | | menuborderstyle (hypersetup key) | 13, 1044 |
| _keys_define:nn | 168, 186, 224, 373, 920, 936, 943, 955, 964, 985, 1000, 1019, 1046, 1065, 1188, 1202, 1214, 1237, 1243, 1258, 1287, 1318, 1331, 1342, 1356, 1366, 1375, 1388, 1415, 1436, 1441, 1997, 2079, 2095, 2113, 2151, 2165, 2183, 2211, 2267, 2272, 2285, 2305 | menucolor (hypersetup key) | 934 |
| _l_keys_key_str | 173, 2137, 2370 | mode commands: | |
| _keys_set:nn | 165, 194, 213, 240, 270, 294, 312, 332, 381, 1348, 1355, 2188, 2215, 2218, 2219, 2220, 2221, 2295, 2370 | _mode_if_horizontal:TF | 554, 576 |
| _keys_set_known:nn | 2373 | _mode_leave_vertical: | 237, 267, 291, 309, 329, 597, 678, 737, 784, 821, 2753, 2781, 2798, 2820, 2850, 2893 |
| \kvsetkeys | 164, 2466 | msg commands: | |
| L | | | |
| \label | 8, 12 | _msg_error:nn | 389, 2795, 2868 |
| | | _msg_line_context: | 68 |
| | | _g_msg_module_name_prop | 11 |
| | | _msg_new:nnn | 36, 43, 48, 52, 56, 63, 70, 77, 84, 91, 97, 104, 113, 123 |
| | | _msg_new:nnnn | 12, 22 |
| | | _msg_warning:nn | 152, 1362 |
| | | _msg_warning:nnn | 161, 601, 839 |
| | | _msg_warning:nnnn | 1206, 1247, 1484, 1528, |

| | |
|--|---|
| 1562, 1585, 1647, 1676, 1710, 1725, 1738, 1751, 1776, 1809, 1825, 1852, 1883, 1916, 1930, 1957, 1991, 2073 | \msg_warning:nnnnn 172, 1278, 1312, 1423, 1504, 1541, 1628, 1661, 1686, 1762, 1789, 1838, 1943, 1970, 2205, 2312, 2331, 2350, 2361 |
| N | |
| naturalnames (hypersetup key) | 1388 |
| nesting (hypersetup key) | 13 |
| \newcommand | 142 , 3418 , 3420 , 3422 |
| \newcount | 2578 , 2749 |
| \NewDocumentCommand | 2374 |
| \NewExpandableDocumentCommand | 360 |
| \newlength | 3437 |
| \noexpand | 2660 , 2661 , 2662 , 2663 , 2989 |
| \nolinkurl | 3 , 4 |
| O | |
| ocgcolorfile (hypersetup key) | 1184 |
| ocgcolorlink (hypersetup key) | 1184 |
| ocgcolorlinks (hypersetup key) | 1184 |
| ocgcolormenu (hypersetup key) | 1184 |
| ocgcolorrun (hypersetup key) | 1184 |
| ocgcolorurl (hypersetup key) | 1184 |
| P | |
| pageanchor (hypersetup key) | 1388 |
| \paperwidth | 3 |
| \PassOptionsToPackage | 396 , 1380 , 1385 |
| pdf commands: | |
| \pdf_bdcobject:nn | 1170 , 1173 |
| \pdf_destination:nn | 25 , 134 , 571 |
| \pdf_destination:nnnn | 564 |
| \pdf_emc: | 1172 , 1178 |
| \pdf_link_user:nnn | 2756 |
| \pdf_name_from_unicode_e:n | 219 , 708 , 823 |
| \pdf_object_if_exist:nTF | 706 |
| \pdf_object_new:nn | 1092 , 1093 , 1094 , 1095 , 2911 , 2912 , 2913 |
| \pdf_object_ref:n | 135 , 718 , 1098 , 1100 , 1122 , 1123 , 1126 , 1130 , 1135 , 1140 , 1145 , 2497 , 2499 , 2969 |
| \pdf_object_ref_last: | 739 , 2300 |
| \pdf_object_unnamed_write:nn | 738 , 2296 |
| \pdf_object_write:nn | 1096 , 1102 , 1112 , 1124 , 2914 , 2956 , 2963 |
| \pdf_pageobject_ref:n | 1869 , 1894 , 2593 |
| \pdf_string_from_unicode:nnN | 515 |
| \pdf_version: | 1532 , 1651 , 1680 , 1714 , 1729 , 1755 , 1780 , 1813 , 1829 , 1934 , 1961 |
| \pdf_version_compare:NnTF | 1522 , 1613 , 1635 , 1671 , 1704 , 1719 , 1745 , 1770 , 1796 , 1819 , 1924 , 1951 |
| \pdf_version_compare_p:Nn | 775 , 1185 , 1234 |
| \pdf_version_major: | 149 , 1186 , 1209 , 1235 , 1251 |
| \pdf_version_minor: | 148 , 1209 , 1251 |
| pdfannot commands: | |
| \pdfannot_box:nnnn | 2628 , 2726 , 2785 , 2824 , 2859 , 2896 |
| \pdfannot_box_ref_last: | 2421 , 2437 |
| \pdfannot_dict_put:nnn | 679 , 739 , 785 , 824 , 976 , 1011 , 1035 , 1057 , 1081 , 1263 , 1294 |
| \pdfannot_dict_remove:nn | 970 , 1006 , 1027 , 1052 , 1073 , 1271 , 1304 |
| \pdfannot_link:nnn | 680 , 740 , 786 , 825 |
| \pdfannot_link_goto_begin:nw | 614 |
| \pdfannot_link_goto_end: | 619 |
| \pdfannot_link_margin:n | 8 , 1566 |
| \c_pdfannot_link_types_seq | 483 |
| pdfauthor (hypersetup key) | 2093 |
| \pdfbookmark | 3423 |
| pdfborder (hypersetup key) | 13 |
| pdfborderstyle (hypersetup key) | 13 , 1044 |
| pdfcreationdate (hypersetup key) | 13 , 2149 |
| pdfcreator (hypersetup key) | 2093 |
| \pdfdest | 11 |
| pdfdict commands: | |
| \pdfdict_new:n | 502 , 657 , 694 , 753 , 810 |
| \pdfdict_put:nnn | 211 , 218 , 503 , 658 , 659 , 670 , 673 , 695 , 696 , 708 , 715 , 725 , 735 , 754 , 755 , 766 , 781 , 811 , 812 , 822 , 1592 , 1593 , 1597 , 1598 , 2309 , 2319 , 2323 , 2326 , 2328 , 2347 , 2356 , 2358 |
| \pdfdict_remove:nn | 208 , 778 , 1602 , 1603 |
| \pdfdict_use:n | 679 , 738 , 785 , 791 , 824 , 829 , 2298 |
| pdfencoding (hypersetup key) | 1356 |
| \pdfescapestring | 2978 |
| pdffile commands: | |
| \pdffile_embed_file:nnn | 710 |
| PDFForm internal commands: | |
| \PDFForm_hypName | 3017 , 3020 , 3024 , 3027 |
| pdfinfo (hypersetup key) | 2211 |
| pdfkeywords (hypersetup key) | 2093 |
| pdflang (hypersetup key) | 13 , 2079 |
| pdflinkmargin (hypersetup key) | 13 |
| pdfmanagement commands: | |
| \pdfmanagement_add:nn | 1575 |

| | |
|--|--|
| \pdfmanagement_add:nnn | 538, 1122, 1123, 1145, 1451, 1476, 1492, 1496, 1516, 1524, 1553, 1608, 1618, 1642, 1656, 1668, 1673, 1698, 1706, 1721, 1747, 1772, 1804, 1821, 1867, 1892, 1906, 1926, 1953, 1982, 2088, 2107, 2125, 2136, 2160, 2174, 2192, 2281, 2300, 2472, 2489, 2496, 2498, 2500, 2506 |
| \pdfmanagement_if_active_p: ... | 388 |
| \pdfmanagement_remove:nn | 1448, 1472, 1480, 1500, 1512, 1537, 1549, 1557, 1571, 1580, 1624, 1639, 1658, 1695, 1734, 1759, 1785, 1800, 1834, 1864, 1889, 1902, 1911, 1939, 1966, 1978, 1987, 2085, 2101, 2119, 2157, 2171, 2278, 2291, 2510 |
| \pdfmanagement_show:n | 2473 |
| pdfmeta commands: | |
| \pdfmeta_standard_verify:nnTF ... | 819, 2772, 2851 |
| \pdfmeta_standard_verify:nTF ... | 2502, 3085 |
| pdfmoddate (hypersetup key) | 13, 2149 |
| pdfproducer (hypersetup key) | 2093 |
| pdfremotestartview (hypersetup key) .. | 11 |
| pdfstartview (hypersetup key) | 11 |
| \pdfstringdef | 8, 9, 17, 21, 26, 3034 |
| \pdfstringdefDisableCommands | 361, 362, 363 |
| pdfsubject (hypersetup key) | 2093 |
| pdftitle (hypersetup key) | 2093 |
| pdftrapped (hypersetup key) | 2183 |
| pdfversion (hypersetup key) | 1356 |
| pdfview (hypersetup key) | 11, 1997 |
| pdfxform commands: | |
| \pdfxform_new:nnn | 2376 |
| \pdfxform_ref:n | 2736, 2737, 2831, 2832, 3116, 3117 |
| \phantom | 2533, 2557 |
| \phantomsection | 3506 |
| plainpages (hypersetup key) | 1388 |
| prg commands: | |
| \prg_generate_conditional_ | |
| variant:Nnn | 407 |
| \prg_new_eq_conditional:NNn ... | 406 |
| prop commands: | |
| \prop_const_from_keyval:Nn | 440, 448, 3567, 3582, 3597, 3612, 3627, 3642, 3658, 3673 |
| \prop_gput:Nnn | 11, 2394, 2401 |
| \prop_if_empty:NnTF | 2475 |
| \prop_item:Nn | 2492 |
| \prop_map_inline:Nn | 898, 962, 998, 1025, 1033, 1044, 1071, 1079, 1148, 1256, 1292, 1302, 1346, 1434, 2470, 2477 |
| \prop_new:N | 2380, 2381 |
| \protect | 241, 271, 295, 313, 333, 412, 634, 684, 691, 745, 751, 795, 801, 844, 3439 |
| \providecommand | 140, 141, 2408, 2411, 2638, 2974 |
| \ProvidesFile | 3 |
| R | |
| \ReadBookmarks | 3428 |
| ref commands: | |
| \ref_if_exist:nn | 406 |
| \ref_label:nn | 402 |
| \ref_value:nn | 23, 423 |
| regex commands: | |
| \regex_const:Nn | 489 |
| \regex_extract_once:NnNTF | 1847, 1878 |
| \relax | 634, 684, 691, 745, 751, 795, 801, 844, 2641, 2642, 2683, 2971, 3018, 3022, 3092, 3096, 3199, 3200, 3208, 3212, 3239, 3243, 3323, 3331, 3358, 3366, 3490, 3499 |
| \RemoveFromHook | 1456 |
| \RenewDocumentCommand | 155 |
| \RequirePackage | 6, 18, 392, 3426 |
| run (hypersetup key) | 9, 1434 |
| runborder (hypersetup key) | 13 |
| runborderstyle (hypersetup key) .. | 13, 1044 |
| runcolor (hypersetup key) | 934 |
| S | |
| \selectfont | 2520, 2532, 2544, 2556 |
| seq commands: | |
| \seq_const_from_clist:Nn | 438 |
| \seq_count:N | 2006, 2032 |
| \seq_get_right:NN | 2008 |
| \seq_item:Nn | 1849, 1880, 2002, 2050, 2052, 2062, 2067 |
| \seq_map_inline:Nn | 475, 479, 483, 934, 1231, 1329, 2487 |
| \seq_new:N | 428 |
| \seq_put_right:Nn | 2039, 2040, 2041, 2479 |
| \seq_set_split:Nnn | 2001 |
| \seq_sort:Nn | 2481 |
| \setbox | 354 |
| \setlength | 3456, 3527 |
| \setpdfinkmargin | 8 |
| \settowidth | 2653 |
| \show | 2714 |

| | | | |
|---|--|---------------------------------|--|
| \Fld@cbsymbol | 3133 | \href@ | 249, 255 |
| \Fld@charsize | .. 2699, 3137, 3218, 3249, 3300, 3356 | \href@split | 255, 256 |
| \Fld@checkequals | 2651, 2719 | \Hy@abspage | 548 |
| \Fld@choices | 3220 | \Hy@activeanchorfalse | 593 |
| \Fld@color 3138, 3140, 3219, 3250, 3301 | | \Hy@activeanchortrue | 586 |
| \Fld@default | ... 2606, 2623, 2641, 2684, 2714, 2715, 2877, 3251, 3255, 3256, 3303 | \Hy@AtBeginDocument | 2388, 2404 |
| \Fld@flags | 3109, 3162, 3195, 3231, 3270, 3319, 3354 | \Hy@bookmarksnumberedfalse ... | 1460 |
| \Fld@format@code | 3043, 3045 | \Hy@bookmarksnumberedtrue ... | 1461 |
| \Fld@height | 2609, 2630, 2633, 2644, 2702, 2728, 2741, 2759, 2808, 2844, 2880, 2898 | \Hy@bookmarksopenfalse | 1464 |
| \Fld@keystroke@code | 3039, 3041 | \Hy@bookmarksopentruetrue | 1465 |
| \Fld@listcount 2713, 2721, 2742, 2749 | | \Hy@bookmarkstype | 1468 |
| \Fld@mappingname 3022, 3024, 3096, 3098 | | \Hy@chapapp | 3471, 3482 |
| \Fld@maxlen | 3306, 3308 | \Hy@colorlink | 32 |
| \Fld@menulength | 2646, 2652, 2699 | \Hy@currentbookmarklevel | 3412 |
| \Fld@name | .. 2605, 2640, 2765, 2876, 3017, 3091 | \Hy@DisableOption | 147, 154 |
| \Fld@onblur@code | 3059, 3061 | \Hy@drafttrue | 1379 |
| \Fld@onclick@code | 3184 | \Hy@escapeform | .. 2627, 2725, 2755, 2783, 2822, 2857, 2895, 2976, 2983, 2986 |
| \Fld@onenter@code | 3071, 3073 | \Hy@escapestring | 2978, 2980, 2985, 2988, 2989, 2992, 2993, 3002, 3005, 3041, 3045, 3049, 3053, 3057, 3061, 3065, 3069, 3073, 3077, 3133, 3184, 3247, 3303, 3304, 3341 |
| \Fld@onexit@code | 3075, 3077 | \Hy@finaltrue | 1384 |
| \Fld@onfocus@code | 3055, 3057 | \Hy@FormObjects | 2469, 2909, 2971 |
| \Fld@onmousedown@code ... | 3063, 3065 | \Hy@gtemp | 3034, 3036 |
| \Fld@onmouseup@code | 3067, 3069 | \Hy@href | 249 |
| \Fld@pageobjref | .. 2575, 2589, 2974, 3106, 3159, 3192, 3228, 3267, 3316, 3351 | \Hy@href@nextactionraw | 212 |
| \Fld@radiosymbol | 3247 | \Hy@href@page | 200 |
| \Fld@rotation | 3121, 3123, 3167, 3174, 3176, 3198, 3204, 3206, 3235, 3237, 3275, 3284, 3286, 3322, 3327, 3329, 3357, 3362, 3364 | \Hy@hypertextnamesfalse | 1394 |
| \Fld@submitflags | 3343 | \Hy@hypertextnametrue | 1393 |
| \Fld@validate@code | 3047, 3049 | \Hy@linkfileprefix | 1396 |
| \Fld@value | .. 2607, 2623, 2642, 2683, 2684, 3304 | \Hy@linktoc | 1419 |
| \Fld@width | 2608, 2622, 2629, 2633, 2643, 2682, 2693, 2695, 2727, 2741, 2759, 2777, 2807, 2818, 2843, 2855, 2879, 2889, 2897 | \Hy@localanchornamefalse | 1399 |
| \Fld@X@additionalactions | .. 3038, 3083, 3086 | \Hy@localanchornamestrue | 1398 |
| \Form@action | 3341 | \Hy@MakeCurrentHrefAuto | .. 3455, 3471, 3482, 3512, 3526 |
| \G@refundefinedtrue | 412 | \Hy@naturalnamesfalse | 1403 |
| \H@old@part | 3497, 3508 | \Hy@naturalnametrue | 1402 |
| \H@old@schapter | 3467, 3476 | \Hy@next | 3481, 3490, 3493 |
| \H@old@ssect | 3518, 3529, 3538 | \Hy@numberline | 143, 3413 |
| \H@old@spart | 3510, 3516 | \Hy@org@chapter | 3479, 3494 |
| \H@old@ssect | 3453, 3458 | \Hy@OutlineName | 3429 |
| | | \Hy@OutlineRerunCheck | 3427 |
| | | \Hy@pageanchorfalse | 1407 |
| | | \Hy@pageanchortrue | 1406 |
| | | \Hy@pdfmajorversion | 149 |
| | | \Hy@pdfminorversion | 148 |
| | | \Hy@pdfstringtrue | 27, 520 |
| | | \Hy@pdfversion | 3029 |
| | | \Hy@plainpagesfalse | 1411 |
| | | \Hy@plainpagetrue | 1410 |
| | | \Hy@pstringdef | 530, 3009 |

| | | | |
|---------------------------|--|----------------------------|--|
| \Hy@PutCatalog | 536 | \hyper@linkend | 31, 32, 648 |
| \Hy@raisedlink | 3447, 3472, 3483, 3513 | \hyper@linkfile | 301, 699 |
| \Hy@RestoreLastskip | 575 | \hyper@linklaunch | 34, 319, 757, 808 |
| \Hy@safe@activestru | 677, 2426 | \hyper@linknamed | 36, 157, 814 |
| \Hy@SaveLastskip | 555 | \hyper@linkstart | 31, 32, 637 |
| \Hy@secnum@part | 3500, 3502, 3505 | \hyper@linkurl | 283, 347, 661 |
| \Hy@SectionAnchorH | 3438, 3460, 3531 | \hyper@normalise | 249, 279, 296, 314, 341, 354 |
| \Hy@SectionHShift | 3437, 3446, 3456, 3527 | \HyPL@Labels | 538, 548 |
| \Hy@StepCount | 2652, 2721 | \HyPL@storePageLabel | 536 |
| \Hy@temp | 2427, 2442, 2991, 2992 | \HyPsd@LoadUnicode | 144 |
| \Hy@unicodetrue | 3031 | \HyPsd@pdfencoding | 146 |
| \Hy@unicodetrue | 145 | \HyPsd@pdfencoding@unicode | 146 |
| \Hy@VerboseAnchor | 556 | \HyPsd@SanitizeForOutFile | 3416 |
| \Hy@verbosefalse | 1370 | \if@files | 2406, 2429, 2448 |
| \Hy@VerboseLinkStart | 626, 641 | \ifFld@combo | 2667, 2697 |
| \Hy@VerboseLinkStop | 632, 654, 687, 747, 797, 835 | \ifFld@hidden | 2622, 2682 |
| \Hy@verbosetrue | 1369 | \ifFld@multiline | 2610 |
| \Hy@VersionChecked | 136 | \ifFld@popdown | 2668 |
| \Hy@WrapperDef | 552 | \ifFld@radio | 2664, 2687 |
| \Hy@xspace@end | 629, 686, 746, 796, 834 | \ifHy@implicit | 3433 |
| \HyAnn@AbsPageLabel | 2574, 2580, 2626, 2724, 2754, 2782, 2821, 2856, 2894 | \ifHy@pdfescapeform | 2977, 2987 |
| \HyAnn@Count | 2578, 2579, 2582, 2583, 2584, 2585, 2586, 2587, 2591, 2595 | \ifHy@unicode | 3030 |
| \HyField@AddToFields | 2417, 2634, 2743, 2760, 2791, 2835, 2864, 2901 | \kv@set@family@handler | 160 |
| \HyField@afields | 2383 | \ltx@empty | 2383, 2384, 2423, 2573, 2574, 2575, 2606 |
| \HyField@AfterAuxOpen | 2388, 2414, 2447 | \ltx@gobble | 2386 |
| \HyField@AuxAddToCoFields | 2387, 2399, 2411, 2432 | \ltx@gobbletwo | 2387 |
| \HyField@AuxAddToFields | 2386, 2392, 2408, 2450 | \ltx@ifUndefined | 3478, 3488 |
| \HyField@cofields | 2384 | \ltx@one | 2582 |
| \HyField@FlagsCheckBox | 2886 | \ltx@ReturnAfterFi | 3010 |
| \HyField@FlagsChoice | 2692 | \ltx@zero | 2579 |
| \HyField@FlagsPushButton | 2774, 2814, 2853 | \m@ne | 3487, 3502 |
| \HyField@FlagsRadioButton | 2688 | \OBJ@OCG@view | 142 |
| \HyField@FlagsSubmit | 2815 | \p@ | 2656 |
| \HyField@FlagsText | 2621 | \pdf@ifdraftmode | 2467 |
| \HyField@PDFChoices | 2752 | \PDF@SetupDoc | 137 |
| \HyField@SetKeys | 2617, 2658, 2661, 2680, 2767, 2810, 2846, 2881 | \pdf@strcmp | 2483 |
| \HyPat@ObjRef | 3385, 3392, 3399, 3406 | \PDFForm@Check | 2895, 2900, 3102 |
| \hyper@@link | 257 | \PDFForm@List | 2755, 2758, 3189 |
| \hyper@anchor | 578 | \PDFForm@Name | 2620, 2681, 2771, 2885, 3016 |
| \hyper@anchorend | 578, 3448, 3473, 3484, 3514 | \PDFForm@Push | 2783, 2789, 3155 |
| \hyper@anchorstart | 578, 3448, 3473, 3484, 3514 | \PDFForm@Radio | 2725, 2731, 3224 |
| \hyper@link | 31, 32, 622 | \PDFForm@Reset | 2857, 2863, 3348 |
| | | \PDFForm@Submit | 2822, 2829, 3313 |
| | | \PDFForm@Text | 2627, 2632, 3263 |
| | | \protected@edef | 598 |
| | | \strip@pt | 3137, 3218, 3249, 3300, 3356 |
| | | \toks@ | 3458, 3464, 3529, 3535 |
| | | \tw@ | 3459, 3464, 3530, 3535 |
| | | \url@ | 354, 357 |
| | | \Url@def | 355 |
| | | \Url@HyperHook | 355 |

| | | | |
|--|---|------------------------------------|---------------------------------------|
| <code>\XR@ext</code> | 1390 | <code>\tl_new:N</code> | 179, |
| <code>\z@</code> | 354, 2647, 3198, 3204, | | 180, 181, 182, 427, 431, 432, 433, |
| | 3235, 3322, 3327, 3357, 3362, 3500 | | 434, 457, 458, 459, 460, 461, 462, |
| <code>\zref@labelbylist</code> | 2584 | | 463, 464, 465, 473, 2458, 2460, 2710 |
| <code>\zref@labelbyprops</code> | 2583 | <code>\tl_set:Nn</code> | 199, 200, 212, 244, |
| <code>\zref@refused</code> | 2585 | | 247, 274, 277, 336, 339, 366, 467, |
| <code>\texorpdfstring</code> | 27 | | 468, 469, 470, 471, 472, 704, 1845, |
| text commands: | | | 1849, 1853, 1876, 2011, 2014, 2021, |
| <code>\text_expand:n</code> | 704 | | 2024, 2025, 2026, 2027, 2028, 2029, |
| <code>\text_purify:n</code> | 8, 507 | | 2034, 2038, 2074, 2459, 2461, 2711 |
| <code>\textcolor</code> | 2465, 2563, 2568 | <code>\tl_set_eq:NN</code> | 605 |
| <code>\textsf</code> | 2563, 2568 | <code>\tl_to_str:N</code> | 706, 713, 718 |
| <code>\the</code> | 548, | <code>\tl_to_str:n</code> | |
| | 2583, 2584, 2585, 2586, 2587, 2591, | | 18, 19, 20, 31, 415, 2160, 2174 |
| | 2595, 2695, 2699, 2702, 3464, 3535 | token commands: | |
| <code>\thepage</code> | 416 | <code>\token_to_str:N</code> | 27 |
| <code>\ThisShouldNotHappen</code> | 2996 | <code>\toks</code> | 3459, 3464, 3530, 3535 |
| <code>\title</code> | 2 | | |
| tl commands: | | | |
| <code>\c_space_tl</code> | 143, 1099, | | |
| | 1130, 1135, 1140, 2736, 2737, 2744, | | |
| | 2920, 2922, 2932, 2934, 2936, 2939, | | |
| | 2941, 3137, 3140, 3272, 3300, 3301 | | |
| <code>\tl_const:Nn</code> | 437 | | |
| <code>\tl_gput_right:Nn</code> | 548 | | |
| <code>\tl_gset:Nn</code> | 1860, 1880, 1884 | | |
| <code>\tl_if_blank:nTF</code> | | | |
| | 723, 2099, 2117, 2155, 2169, 2276, 2289 | | |
| <code>\tl_if_empty:NTF</code> | | | |
| | ... 599, 1446, 3125, 3129, 3138, 3301 | | |
| <code>\tl_if_empty:nTF</code> | | | |
| | 206, 968, 1004, 1023, | | |
| | 1050, 1069, 1637, 1693, 1798, 2083 | | |
| <code>\tl_if_empty_p:N</code> | 1862, 1887 | | |
| <code>\tl_if_eq:NnTF</code> | 2009 | | |
| <code>\tl_if_exist:NTF</code> | 3178, 3288, 3292 | | |
| <code>\tl_if_exist_p:N</code> | 3169, 3277, 3279 | | |
| <code>\tl_if_head_eq_charcode:nNTF</code> | | | |
| | 849, 866, 883 | | |

U

| | |
|-----------------------------------|-------------------|
| unicode (hypersetup key) | 1356 |
| <code>\Url</code> | 3, 5, 183 |
| url (hypersetup key) | 9, 1434 |
| <code>\url</code> | 3–6, 21, 327, 354 |
| urlborder (hypersetup key) | 13 |
| urlborderstyle (hypersetup key) . | 13, 1044 |
| urlcolor (hypersetup key) | 934 |
| <code>\urldef</code> | 5, 21, 354, 356 |
| <code>\Urlfont</code> | 6 |
| <code>\UrlSpecial</code> | 4 |
| use commands: | |
| <code>\use:n</code> | 803 |

V

| | |
|--------------------------------|------|
| verbose (hypersetup key) | 1366 |
|--------------------------------|------|

W

| | |
|---------------------------|------------------------|
| <code>\write</code> | 2407, 2410, 2430, 2449 |
|---------------------------|------------------------|

X

| | |
|-----------------------|------------------------------------|
| <code>\x</code> | 2659, 2679, 3463, 3465, 3534, 3536 |
|-----------------------|------------------------------------|