

The `hyperref-generic` module

A generic driver for `hyperref`

The L^AT_EX Project*

Version 0.95b, released 2021-03-07

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.

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{standard=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])

`fileborderstyle` (*setup key*)

`runborderstyle` (*setup key*)

`menuborderstyle` (*setup key*)

`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}
4 [2021/02/22 v0.95b %
5 generic Hyperref driver for the LaTeX PDF management testphase bundle]
6
7 \RequirePackage{etoolbox} %why?
```

```

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

```

1 messages

Redirect the message name:

```

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

```

At first a message for the testing of the resource management

```

13 \msg_new:nnnn
14 { hyp }
15 { missing-resource-management }
16 { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
17 {
18   Activate-it~with ~\
19   \tl_to_str:n{\RequirePackage{pdfmanagement-testphase}}\
20   \tl_to_str:n{\DeclareDocumentMetadata{<options>}}\
21   before~\tl_to_str:n{\documentclass}
22 }

```

The pdfversion should be set in \DeclareDocumentMetadata

```

23 \msg_new:nnnn
24 { hyp }
25 { pdfversion-disabled }
26 {
27   This~hyperref~driver~ignores~the~pdfversion~key!\
28   Set~the~pdfversion~in~\token_to_str:N \DeclareDocumentMetadata
29 }
30 {
31   For example:\
32   \tl_to_str:n
33   {
34     \DeclareDocumentMetadata { pdfversion=1.7 }
35   }
36 }

```

A generic message for ignored keys.

```

37 \msg_new:nnn
38 { hyp }
39 { key-dropped }
40 {
41   This~hyperref~driver~ignores~the~key~#1!\
42   Please~check~the~documentation.
43 }

```

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

```

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

```

```

51 { pdfa-no-reset-button }
52 { PDF/A:~Reset~action~is~prohibited }

```

pdf/A message for not allowed Named actions

```

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

```

A message if the destination name is empty.

```

57 \msg_new:nnn
58 { hyp }
59 { empty-destination-name }
60 {
61   Empty~destination~name,\\
62   using~'#1'
63 }

```

A message if the destination check fails

```

64 \msg_new:nnn
65 { hyp }
66 { invalid-destination-value }
67 {
68   Invalid~value~'#1'~of~'#2'  \\
69   is~replaced~by~'Fit'~\msg_line_context:.
70 }

```

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

```

71 \msg_new:nnn
72 { hyp }
73 { ignore-deprecated-or-unknown-option-in-pdf-version }
74 {
75   Option~'#1'~is~unknown~or~deprecated~in\\
76   pdf~version~#2.~Ignored.
77 }
78 \msg_new:nnn
79 { hyp }
80 { ignore-deprecated-or-unknown-value-in-pdf-version }
81 {
82   Value~'#1'~is~unknown~or~deprecated~in\\
83   pdf~version~#2.~Ignored.
84 }
85 \msg_new:nnn
86 { hyp }
87 { replace-deprecated-or-unknown-value-in-pdf-version }
88 {
89   Value~'#1'~is~unknown~or~deprecated~in\\
90   pdf~version~#2. Value~'#3'~is used instead.
91 }

```

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:

```

92 \msg_new:nnn
93 { hyp }
94 { unknown-key }
95 {

```

```

96     unknown~key~#2~of~module~'#1'~set~to~'#3'.
97   }
98   \msg_new:nnn
99   { hyp }
100  { unknown-key-to-Hyp }
101  {
102    ignored~in~family~Hyp~unknown~key~#1.
103  }

```

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

```

104  \cs_new:Npn \__hyp_clist_display:n #1 {*~#1\\}
105  \msg_new:nnn
106  { hyp }
107  { unknown-choice }
108  {
109    Value~'#3'~is~invalid~for~key~'#1'.\\
110    The~key~accepts~only~the~choices\\
111    \clist_map_function:nN { #2 } \__hyp_clist_display:n
112  }
113
114  \msg_new:nnn
115  { hyp }
116  { unknown-choice+empty }
117  {
118    Value~'#3'~is~invalid~for~key~'#1'.\\
119    The~key~accepts~only~the~choices\\
120    \clist_map_function:nN { #2 } \__hyp_clist_display:n
121    An~empty~value~removes~the~setting.
122  }
123
124  \msg_new:nnn
125  { hyp }
126  { no-bool }
127  {
128    Value~'#2'~is~invalid~for~key~'#1'.\\
129    The~key~accepts~only~the~choices\\
130    *~true\\
131    *~false \\
132    *~and~an~empty~value~which~removes~the~setting.\\
133    No~value~is~equivalent~to~using~'true'.
134  }

```

2 Variants

```

135  \cs_generate_variant:Nn \pdf_destination:nn {nf}
136  \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

```

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

```



```

138 %\cs_set_protected:Npn \PDF@SetupDoc{}
139 %\PDF@FinishDoc{}% dummy needed for hyperref ...

\hypercalcbp We define a better (expandable) version of \hypercalcbp
140 \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:
141 \providecommand\@pdfborder{}
142 \providecommand\@pdfborderstyle{}
143 \newcommand\OBJ@OCG@view {} % needed in hyperref
144 \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.
145 \HyPsd@LoadUnicode
146 \Hy@unicodetrue
147 \let\HyPsd@pdfencoding\HyPsd@pdfencoding@unicode
148 \Hy@DisableOption{unicode}

The pdfversion should be set in \DeclareDocumentMetadata but we must copy it to the
hyperref command:
149 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
150 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:

151 \legacy_if:nT { Hy@setpdfversion }
152 {
153   \msg_warning:nn { hyp }{ pdfversion-disabled }
154 }
155 \Hy@DisableOption{pdfversion}

\Acrobatmenu should use the new internal link command
156 \RenewDocumentCommand \Acrobatmenu { m m }
157 {
158   \hyper@linknamed {#1} {#2}
159 }

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

```

```

175     }
176 }

```

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`)

```

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

```

```

224
225 \keys_define:nn { hyp / setup }
226 {
227   ,href / urlencode .bool_set:N = \l__hyp_href_url_encode_bool
228   ,href / urlencode .default:n = {true}
229   ,href / urlencode .initial:n = {false}
230   ,href / protocol .tl_set:N = \l__hyp_href_url_protocol_tl
231   ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
232   ,href / format .code:n = { \cs_set_eq:NN \__hyp_href_url_format #1 }
233 }
234
235
236
237 \DeclareRobustCommand*{\href}[1][]{%
238   \begingroup
239   \keys_set:nn { hyp / href } {#1}
240   \let\protect\unexpandable@protect
241   \bool_if:NTF \l__hyp_href_url_encode_bool
242   {
243     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
244   }
245   {
246     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
247   }
248   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
249 }
250
251
252 \DeclareRobustCommand*{\hrefurl}[1][]{
253   \group_begin:
254   \keys_set:nn { hyp / href } {#1}
255   \let\protect\unexpandable@protect
256   \bool_if:NTF \l__hyp_href_url_encode_bool
257   {
258     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
259   }
260   {
261     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
262   }
263   \hyper@normalise\__hyp_href_url_aux:nn}
264
265 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
266 {
267   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
268   \group_end:
269 }
270
271
272 \DeclareRobustCommand*{\hrefpdf}[1]{}
273 {
274   \begingroup
275   \keys_set:nn { hyp / href } {#1}
276   \let\protect\unexpandable@protect
277   \hyper@normalise\__hyp_href_pdf_aux:nn

```

```

278 }
279
280 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
281 {
282   \exp_args:Nnno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}\endgroup
283 }
284
285 \DeclareRobustCommand*\hrefrun}[1] []
286 {
287   \group_begin:
288   \keys_set:nn { hyp / href } {#1}
289   \let\protect\@unexpandable@protect
290   \hyper@normalise\__hyp_href_run_aux:nn
291 }
292
293 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
294 {
295   \exp_args:Nnno\hyper@linklaunch{#2}{#1}{\l__hyp_href_run_parameter_tl}
296   \group_end:
297 }
298
299 \DeclareRobustCommand*\url}[1] []
300 {
301   \group_begin:
302   \keys_set:nn { hyp / href } {#1}
303   \let\protect\@unexpandable@protect
304   \bool_if:NTF \l__hyp_href_url_encode_bool
305   {
306     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
307   }
308   {
309     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
310   }
311   \hyper@normalise\__hyp_href_url_aux:n
312 }
313
314 \cs_new_protected:Npn \__hyp_href_url_aux:n #1
315 {
316   \exp_args:Nno
317   \hyper@linkurl{\__hyp_href_url_format {#1}}
318   {\l__hyp_href_url_protocol_tl#1}
319   \group_end:
320 }
321
322
323 \NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
324 \pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
325 \pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}
326 \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.

```
327 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
328 {
329   %\tl_set:cn {@#1}{#2}
330   \AddToDocumentProperties[hyperref]{#1}{#2}
331 }
332 \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.

```
333 \hook_new:n{hyp/link/cite}
334 \color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
335 \color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
336 \keys_define:nn { hyp / setup }
337 {
338   ,citecolor .code:n = {\__hyp_color_set:nn {hyp/color/cite}{#1}}
339   ,citebordercolor
340     .code:n = {\__hyp_color_set:nn {hyp/color/citeborder}{#1}}
341 }
342 \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
343 {
344   \keys_set:nn { hyp / setup }
345   {
346     ,linkbordercolor= hyp/color/citeborder
347     ,linkcolor      = hyp/color/cite
348   }
349 }
```

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

```
350 \bool_lazy_and:nnF
351 { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
352 { \msg_error:nn { hyp}{ missing-resource-management } }
```

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

```
353 \legacy_if:nT { Hy@bookmarks }
354 {
355   \AddToHook{begindocument/before}[hyperref/bookmark]{\RequirePackage{bookmark}}
356 }
357 \legacy_if:nT { Hy@draft }
```

```

358 {
359   \PassOptionsToPackage{draft}{bookmark}
360 }

```

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
\__hyp_ref_check:nn
\__hyp_ref_value:nn
361 %
362 \cs_new_protected:Npn \__hyp_ref_label:nn #1 #2 %label/attributes
363 {
364   \@bsphack
365   \ref_label:nn{#1}{#2}
366   \@esphack
367 }
368 \cs_generate_variant:Nn \__hyp_ref_label:nn {en}

```

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

```

369 \prg_new_eq_conditional:Nn \__hyp_ref_if_exist:nn \ref_if_exist:nn { p , T , F, TF }
370 \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.

```

371 \cs_new_protected:Npn \__hyp_ref_check:nn #1 #2 %label/attribute
372 {
373   \__hyp_ref_if_exist:nnF {#1}{#2}
374   {
375     \protect\G@refundefinedtrue
376     \@latex@warning
377     {
378       Reference~'\tl_to_str:n {#1}'~with~attribute~'\tl_to_str:n {#2}'~
379       on~page~\thepage~\space undefined
380     }
381   }
382 }
383 \cs_generate_variant:Nn \__hyp_ref_check:nn {en}

```

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

```

384 \cs_new:Npn \__hyp_ref_value:nn #1 #2
385 {
386   \ref_value:nn{#1}{#2}
387 }
388 \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 389 \box_new:N \l__hyp_tmpa_box
\l__hyp_tmpa_int 390 \tl_new:N \l__hyp_tmpa_tl
\l__hyp_tmpa_box 391 \seq_new:N \l__hyp_tmpa_seq
\l__hyp_tmpa_str 392 \int_new:N \l__hyp_tmpa_int
393 \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.

```
\l__hyp_dest_name_tmpa_tl
\l__hyp_uri_tmpa_tl 394 \tl_new:N \l__hyp_dest_name_tmpa_tl
\l__hyp_filename_tmpa_tl 395 \tl_new:N \l__hyp_uri_tmpa_tl
\__hyp_text_tmpa_str\__g__hyp_text_tmpa_str 396 \tl_new:N \l__hyp_filename_tmpa_tl
397 \tl_new:N \l__hyp_para_tmpa_tl
398 \str_new:N \l__hyp_text_tmpa_str
399 \str_new:N \g__hyp_text_tmpa_str
```

TODO: document and check use!

(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.

```
400 \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`

```
401 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
402 {url,link,file,menu,run}
403 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
404 {
405     URI    = url,
406     GoTo   = link,
407     GoToR  = file,
408     Named  = menu,
409     Launch = run
410 }
411 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
412 {
413     url    = URI,
414     link   = GoTo,
415     file   = GoToR,
416     menu   = Named,
417     run    = Launch
418 }
419
```

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

```
420 \tl_new:N \g__hyp_dest_pdfstartpage_tl
421 \tl_new:N \g__hyp_dest_pdfstartview_tl
422 \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
423 \tl_new:N \l__hyp_text_enc_uri_print_tl
424 \tl_new:N \l__hyp_text_enc_info_print_tl
425 \tl_new:N \l__hyp_text_enc_dest_tl
426 \tl_new:N \l__hyp_text_enc_dest_print_tl
427 \tl_new:N \l__hyp_text_enc_file_print_tl
428 \tl_new:N \l__hyp_text_enc_para_print_tl
```

```
429
430 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
431 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
432 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
433 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
434 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
435 \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.

```
436 \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)

`\g__hyp_bordercolormodel_str` This holds the export model for border color etc. It is currently either `space-sep-cmyk`

`hyp/annot/run` (color name) or `space-sep-rgb`. The default is the second. It can be change by the key

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

```
437 \str_new:N \g__hyp_bordercolormodel_str
```

(End definition for `\g__hyp_bordercolormodel_str`.)

7.4 Booleans

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

```
438 \seq_map_inline:Nn \c__hyp_annot_types_seq
439 {
440     \bool_new:c {l_hyp_annot_color#1_bool}
441 }
```

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

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

```
442 \seq_map_inline:Nn \c__hyp_annot_types_seq
443 {
444     \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
445 }
```

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

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

```

446 \seq_map_inline:Nn \c_pdfannot_link_types_seq
447 {
448   \bool_new:c {l_hyp_annot_#1_bool}
449   \bool_set_true:c {l_hyp_annot_#1_bool}
450 }

```

(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

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

```

451 \box_new:N \l__hyp_dest_box

```

(End definition for \l_hyp_dest_box.)

7.6 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.

```

452 \regex_const:Nn \c__hyp_dest_startview_regex
453 {
454     \A\ *
455     (?:
456         (? : XYZ (? : \ +(?: (? : \d+ | \d* \. \d+ ) | null) ) {3} \ )
457         |
458         (? : Fit \b | FitB \b )
459         |
460         (? : (? : FitH | FitV | FitBH | FitBV ) (? : \ +(?: \d+ | \d* \. \d+ ) | \ + null) {1} )
461         |
462         (? : FitR (? : \ + \d+ | \ + \d* \. \d+ ) {4} \ )
463     )
464 }

```

(End definition for `\c__hyp_dest_startview_regex`.)

7.7 PDF dictionaries

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

```
465 \pdfdict_new:n {l__hyp_page/Trans}
466 \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`

```
467 \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
468 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
469 {
470   \str_set:Nx #2 {\text_purify:n { #1 } }
471 }
```

(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
472 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
473 {
474
475 }
```

(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 `l3pdfutils`.

#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`

```

476 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
477 {
478   \pdf_string_from_unicode:nVN { #1 } #2 #2
479 }

```

(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`

```

480 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
481 {
482   \group_begin:
483   \Hy@pdfstringtrue
484   \hook_use:n {hyp/text/pdfstring}
485   \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
486   \__hyp_text_cleanup:N      \l__hyp_text_tmpa_str
487   \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
488   \str_gset_eq:NN \g__hyp_text_tmpa_str\l__hyp_text_tmpa_str
489   \group_end:
490   \str_set_eq:NN #3 \g__hyp_text_tmpa_str
491 }
492 \cs_generate_variant:Nn \__hyp_text_pdfstring:nnN {xnN,onN,xoN,ooN,noN}

```

(End definition for `__hyp_text_pdfstring:nnN`.)

!!! temporary until all instances are gone

```

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

```

This is a special version for info keys:

`__hyp_text_pdfstring_info:nN`

```

495 \cs_new_protected:Npn \__hyp_text_pdfstring_info:nN #1 #2
496 {
497   \__hyp_text_pdfstring:noN { #1 } { \l__hyp_text_enc_info_print_tl } #2
498 }

```

(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: hxtex.def, line 80-110

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

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

dvipdfm: identical to xetex

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

luatex identical to pdftex

The code in `hyperref` inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-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
499 \cs_new_protected:Npn \__hyp_PageLabels_gpush:
500   {
501     \pdfmanagement_add:nnx {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
502   }
503
504 \def\Hy@PutCatalog #1 {}
505
506
507 \legacy_if:nT { Hy@pdfpagelabels }
508   {
509     \cs_set_protected:Npn \HyPL@StorePageLabel #1
510       {
511         \tl_gput_right:Nx \HyPL@Labels { \the\Hy@abspage<<#1>> }
512         \__hyp_PageLabels_gpush:
513       }
514   }
```

(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 `\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

515 \Hy@WrapperDef \__hyp_destination:nn #1 #2
516 {
517   \mode_if_horizontal:T { \@savs\spacefactor }
518   \Hy@SaveLastskip      %defined in hyperref
519   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
520   \__hyp_text_pdfstring:xoN
521   { \HyperDestNameFilter{#1} }
522   { \l__hyp_text_enc_dest_tl }
523   \l__hyp_tmpa_tl
524   \str_if_eq:nnTF {#2} {fitrbx}
525   {
526     \exp_args:NV
527     \pdf_destination:nnnn \l__hyp_tmpa_tl
528     { \box_wd:N \l__hyp_dest_box }
529     { \box_ht:N \l__hyp_dest_box }
530     { \box_dp:N \l__hyp_dest_box }
531   }
532   {
533     \exp_args:NV
534     \pdf_destination:nf
535     { \l__hyp_tmpa_tl }
536     { #2 }
537   }
538   \Hy@RestoreLastskip %defined in hyperref
539   \mode_if_horizontal:T { \spacefactor\@savs }
540 }
```

(End definition for `__hyp_destination:nn`.)

These are the three destinations commands. They are modelled along the xetex version. It is not quite clear if really all three are needed for the backends supported by this driver, but changing the hyperref code would be difficult.

```
\hyper@anchor
\hyper@anchorstart
\hyper@anchorend

541 \cs_new_protected:Npn \hyper@anchor #1
542 {
543   \exp_args:NnV
544   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
545 }
```

```

546
547 \cs_new_protected:Npn \hyper@anchorstart #1
548 {
549   \Hy@activeanchortrue
550   \exp_args:NnV
551     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
552 }
553
554 \cs_new_protected:Npn \hyper@anchorend
555 {
556   \Hy@activeanchorfalse
557 }

```

(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:

```

558 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
559 {
560   \mode_leave_vertical:
561   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
562   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
563     {
564       \msg_warning:nnx
565         { hyp }
566         { empty-destination-name }
567         { \c__hyp_dest_undefined_tl }
568       \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
569     }
570     {
571       \__hyp_text_pdfstring:xoN
572       { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
573       { \l__hyp_text_enc_dest_tl }
574       \l__hyp_dest_name_tmpa_tl
575     }
576   \exp_args:No
577     \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
578 }
579
580 \cs_new_protected:Npn \__hyp_link_goto_end:
581 {
582   \pdfannot_link_goto_end:
583 }

```

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.

`\hyper@linkstart` `\hyper@linkstart{<context>}{<destination name>}`
`\hyper@linkend` `\hyper@linkend`

This creates the start and end commands for a GoTo link around the text between both pointing to `<destination name>`. The hook `hyp/link/<context>` is executed at the begin if it exists as with `\hyper@link`

The commands open and close a group, so should be placed carefully. .

`hyperref` adds a group with `\Hy@colorlink`, we move this outside the link so that it groups the context hook too.

```
584
585 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
586 {
587   \bool_if:NTF \l__hyp_annot_GoTo_bool
588   {
589     \Hy@VerboseLinkStart{#1}{#2}
590     \group_begin:
591     \hook_use:n {hyp/link/#1}
592     \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
593     \__hyp_link_goto_end:
594     \group_end:
595     \Hy@VerboseLinkStop
596   }
597   {\let\protect\relax#3}}
598 }
599
600 \cs_new_protected:Npn \hyper@linkstart #1 #2 % #1 context, #2=destination name
601 {
602   \bool_if:NT \l__hyp_annot_GoTo_bool
603   {
604     \Hy@VerboseLinkStart{#1}{#2}% only for debug
605     \group_begin:
606     \hook_use:n {hyp/link/#1}
607     \__hyp_link_goto_begin:nw {#2}
608   }
609 }
610
611 \cs_new_protected:Npn \hyper@linkend
612 {
613   \bool_if:NT \l__hyp_annot_GoTo_bool
614   {
615     \__hyp_link_goto_end:
616     \group_end:
```

```

617         \Hy@VerboseLinkStop
618     }
619 }

```

10.3 URI links

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

```

620 \pdfdict_new:n {l_hyp/annot/A/URI}
621 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
622 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
623
624 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
625 {
626     \bool_if:NTF \l__hyp_annot_URI_bool
627     {
628         \group_begin:
629         \__hyp_text_pdfstring:xoN
630             { #2}
631             { \l__hyp_text_enc_uri_print_tl }
632             \l__hyp_uri_tmpa_tl
633         \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
634         \bool_if:NT \l__hyp_href_url_ismap_bool
635             {
636                 \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
637             }
638         \cs_set_eq:NN \# \c_hash_str
639         \cs_set_eq:NN \% \c_percent_str
640         \Hy@safe@activetrue
641         \mode_leave_vertical:
642         \pdfannot_dict_put:nnx {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
643         \pdfannot_link:nnx { URI }
644             {
645             }
646             {
647                 \let\protect\relax
648                 #1
649                 \Hy@xspace@end
650                 \Hy@VerboseLinkStop %where is the start??
651             }
652         \group_end:
653     }
654     {{\let\protect\relax#1}}
655 }
656

```

10.4 GoToR Links files

```

657 \pdfdict_new:n {l_hyp/annot/A/GoToR}
658 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
659 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
660
661
662 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname

```



```

663 {
664   \bool_if:NTF \l__hyp_annot_GoToR_bool
665   {
666     \group_begin:
667     \tl_set:Nx \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
668     \exp_args:Nx
669     \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
670     {
671       \pdfdict_put:nnx { l_pdffile/Filespec } { Subtype } { \pdf_name_from_unicode_e:n {ap
672       \exp_args:Nnox
673       \pdffile_embed_file:nnn
674       {}
675       {\l__hyp_filename_tmpa_tl }
676       {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
677     }
678     \pdfdict_put:nnx
679     {l_hyp/annot/A/GoToR}
680     {F}
681     {\pdf_object_ref:e {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
682     \__hyp_text_pdfstring:nnN
683     { #3 }
684     { \l__hyp_text_enc_dest_print_tl }
685     \l__hyp_dest_name_tmpa_tl
686     \tl_if_blank:eTF {#3}
687     {
688       \pdfdict_put:nnx {l_hyp/annot/A/GoToR} {D}
689       {
690         [
691           \int_eval:n
692           { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
693           /\l__hyp_dest_pdfremotestartview_tl
694         ]
695       }
696     }
697     {
698       \pdfdict_put:nno {l_hyp/annot/A/GoToR} {D} { \l__hyp_dest_name_tmpa_tl }
699     }
700     \mode_leave_vertical:
701     \pdf_object_unnamed_write:nx {dict} { \pdfdict_use:n {l_hyp/annot/A/GoToR} }
702     \pdfannot_dict_put:nnx {link/GoToR} {A} { \pdf_object_ref_last: }
703     \pdfannot_link:nxn %expansion??
704     { GoToR }
705     {
706     }
707     {
708       \let\protect\relax
709       #1\Hy@xspace@end
710       \Hy@VerboseLinkStop %where is the start??
711     }
712     \group_end:
713   }
714   {{\let\protect\relax#1}}
715 }

```

10.5 Launch links

We define `\hyper@linklaunch` for naming consistency

```

716 \pdfdict_new:n {l_hyp/annot/A/Launch}
717 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
718 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
719
720 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
721 {
722   \bool_if:NTF \l__hyp_annot_Launch_bool
723   {
724     \group_begin:
725     \__hyp_text_pdfstring:nnN
726       { #1 }
727       { \l__hyp_text_enc_file_print_tl }
728       \l__hyp_filename_tmpa_tl
729     \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
730     \__hyp_text_pdfstring:noN
731       { #3 }
732       { \l__hyp_text_enc_para_print_tl }
733       \l__hyp_para_tmpa_tl
734     \bool_if:NTF
735     {
736       \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
737       ||
738       \pdf_version_compare_p:Nn > {1.9}
739     }
740     {
741       \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
742     }
743     {
744       \pdfdict_put:nnx {l_hyp/annot/A/Launch}{Win}
745       {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
746     }
747     \mode_leave_vertical:
748     \pdfannot_dict_put:nnx {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}>>}
749     \pdfannot_link:nnx
750       { Launch }
751       {
752         % /A
753         % <<
754         % \pdfdict_use:n {l_hyp/annot/A/Launch}
755         % >>
756       }
757       {
758         \let\protect\relax
759         #2\Hy@xspace@end
760         \Hy@VerboseLinkStop %where is the start??
761       }
762     \group_end:
763   }
764   {{\let\protect\relax#2}}
765 }

```

The actually command used by `hyperref` is `\@hyper@launch` which uses a delimited ar-

gument, because of the color the definition is a bit convoluted.

```

766 \use:x
767 { % filename, anchor text, linkname
768 \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str ##1 \exp_not:N \ \ ##2 #
769 }
770 {
771 \hyper@linklaunch {#1}{#2}{#3}
772 }

```

10.6 Named links (menu)

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

```

773 \pdfdict_new:n {l_hyp/annot/A/Named}
774 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
775 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
776
777 \cs_new_protected:Npn \hyper@linknamed #1 #2 % #1 action, #2 link text
778 {
779 \bool_if:NTF \l__hyp_annot_Named_bool
780 {
781 \group_begin:
782 \pdfmeta_standard_verify:nnTF {named_actions}{#1}
783 {
784 \mode_leave_vertical:
785 \pdfdict_put:nnx {l_hyp/annot/A/Named}{N}
786 {\pdf_name_from_unicode_e:n{#1}}
787 \pdfannot_dict_put:nnx {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}>>}
788 \pdfannot_link:nxn { Named }
789 {
790 % /A
791 % <<
792 % \pdfdict_use:n { l_hyp/annot/A/Named }
793 % >>
794 }
795 {
796 #2
797 \Hy@xspace@end
798 \Hy@VerboseLinkStop
799 }
800 }
801 {
802 \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
803 #2
804 }
805 \group_end:
806 }
807 {\let\protect\relax#2}}
808 }
809

```

11 Link decorations

11.1 Functions to export and select colors

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

<code>_hyp_color_export:nnN</code>	<code>_hyp_color_export:nnN {<color>} {<export format>} <tlvar></code>
<code>_hyp_colormodel_export:nnnN</code>	<code>_hyp_colormodel_export:nnnN {<color model>}{<value>} {<export format>}</code>

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} and color=red!50!blue. The colors are exported to PDF syntax (space separated numbers).

```

810 \cs_new_protected:Npn \_hyp_color_export:nnN #1 #2 #3
811 {
812   \tl_if_head_eq_charcode:nNTF {#1}[ %]
813   {
814     \_hyp_colormodel_export:wnnN #1 {#2} #3
815   }
816   {
817     \color_export:nnN {#1} {#2} #3
818   }
819 }
820
821 \cs_new_protected:Npn \_hyp_colormodel_export:wnnN [#1] #2 #3 #4
822 {
823   \color_export:nnnnN {#1}{#2}{#3}#4
824 }
825
826 \cs_generate_variant:Nn \_hyp_color_export:nnN {xVN}

```

(End definition for `_hyp_color_export:nnN` and `_hyp_colormodel_export:nnnN`.)

<code>_hyp_color_select:n</code>	<code>_hyp_color_select:nN {<color>}</code>
-----------------------------------	----------------------------------------------------

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} and color=red!50!blue.

```

827 \cs_new_protected:Npn \_hyp_color_select:n #1
828 {
829   \tl_if_head_eq_charcode:nNTF {#1}[ %]
830   {
831     \_hyp_colormodel_select_aux:wn #1
832   }
833   {
834     \color_select:n {#1}
835   }

```

```

836 }
837
838 \cs_new_protected:Npn \__hyp_colormodel_select_aux:wn [#1] #2
839 {
840   \color_select:nn {#1}{#2}
841 }
842
843 \cs_generate_variant:Nn \__hyp_color_select:n {e}

```

(End definition for `__hyp_color_select:n` and `__hyp_colormodel_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`.

`__hyp_colormodel_set_aux:nwn`

```

844 \cs_new_protected:Npn \__hyp_color_set:nn #1 #2
845 {
846   \tl_if_head_eq_charcode:nNTF {#2}[ %]
847   {
848     \__hyp_colormodel_set_aux:nwn { #1 } #2
849   }
850   {
851     \color_set:nn {#1} {#2}
852   }
853 }
854
855 \cs_new_protected:Npn \__hyp_colormodel_set_aux:nwn #1 [#2] #3
856 {
857   \color_set:nnn {#1}{#2}{#3}
858 }
859
860 \cs_generate_variant:Nn \__hyp_color_set:nn {ne}

```

(End definition for `__hyp_color_set:nn` and `__hyp_colormodel_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.

```

861 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
862 {
863   \hook_gput_code:nnn
864     {pdfannot/link/#2/begin}
865     {hyp/color}
866     {
867       \bool_if:cT { l_hyp_annot_color#1_bool }
868       {
869         \group_begin:
870         \color_select:n { hyp/color/#1}
871       }

```

```

872     }
873     \hook_gput_code:nnn
874     {pdfannot/link/#2/end}
875     {hyp/color}
876     {
877       \bool_if:cT { l_hyp_annot_color#1_bool }
878       {
879         \group_end:
880       }
881     }
882   }

```

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

```

883 \keys_define:nn { hyp / setup }
884 {
885   ,colorlinks .meta:n =
886   {
887     ,pdfborder={0~0~0}
888     ,pdfborderstyle=
889     ,colorurl  =#1
890     ,colorlink =#1
891     ,colorrun  =#1
892     ,colormenu =#1
893     ,colorfile =#1
894   }
895   ,colorlinks .default:n = {true}
896 }

```

```

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)
897 \seq_map_inline:Nn \c__hyp_annot_types_seq
898 {
899   \keys_define:nn { hyp / setup }
900   {
901     ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
902     ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
903   }
904 }
905
906 \keys_define:nn { hyp / setup }
907 {
908   ,allcolors .meta:n =
909   {
910     ,urlcolor=#1
911     ,linkcolor=#1
912     ,runcolor=#1
913     ,filecolor=#1
914     ,menucolor=#1
915   }
916   ,allcolors .value_required:n = true
917 }

```

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*)

```
918 \keys_define:nn { hyp / setup }
919 {
920   ,bordercolormodel .choices:nn =
921     {rgb,cmyk}
922     { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
923   ,bordercolormodel .initial:n = {rgb}
924 }

925 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
926 {
927   \keys_define:nn { hyp / setup }
928   {
929     #1bordercolor .code:n =
930     {
931       \tl_if_empty:nTF { ##1 }
932       {
933         \pdfannot_dict_remove:nn
934           {link/#2}
935           { C }
936       }
937       {
938         \__hyp_color_export:xVN {##1}\g__hyp_bordercolormodel_str \l__hyp_tmpa_tl
939         \pdfannot_dict_put:nnx
940           {link/#2}
941           { C }
942           { [\l__hyp_tmpa_tl] }
943       }
944     }
945   }
946 }

947 \keys_define:nn { hyp / setup }
948 {
949   ,allbordercolors .meta:n =
950   {
951     ,linkbordercolor=#1
952     ,urlbordercolor =#1
953     ,filebordercolor=#1
954     ,menubordercolor=#1
955     ,runbordercolor =#1
956   }
957   ,allbordercolors .value_required:n = true
958 }
959
960
```

11.3.2 Borderwidth and -arc

```

961 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
962 {
963   \keys_define:nn { hyp / setup }
964   {
965     #1border .code:n =
966     {
967       \tl_if_empty:nTF { ##1 }
968       {
969         \pdfannot_dict_remove:nn
970         {link/#2}
971         { Border }
972       }
973       {
974         \pdfannot_dict_put:nnn
975         {link/#2}
976         { Border }
977         { [##1] }
978       }
979     }
980   }
981 }
982 \keys_define:nn { hyp / setup }
983 {
984   ,pdfborder .code:n =
985   {
986     \tl_if_empty:nTF { #1 }
987     {
988       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
989       {
990         \pdfannot_dict_remove:nn
991         {link/##2}
992         { Border }
993       }
994     }
995     {
996       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
997       {
998         \pdfannot_dict_put:nnn
999         {link/##2}
1000         { Border }
1001         { [#1] }
1002       }
1003     }
1004   }
1005   ,pdfborder .initial:n = {0~0~1},
1006 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1007 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1008 {
runborderstyle (setup key) 1009   \keys_define:nn { hyp / setup }
fileborderstyle (setup key)
menuborderstyle (setup key)

```



```

1010     {
1011         #1borderstyle .code:n =
1012         {
1013             \tl_if_empty:nTF { ##1 }
1014             {
1015                 \pdfannot_dict_remove:nn
1016                 {link/#2}
1017                 { BS }
1018             }
1019             {
1020                 \pdfannot_dict_put:nnn
1021                 {link/#2}
1022                 { BS }
1023                 { <<##1>> }
1024             }
1025         }
1026     }
1027 }
1028 \keys_define:nn { hyp / setup }
1029 {
1030     ,pdfborderstyle .code:n =
1031     {
1032         \tl_if_empty:nTF { #1 }
1033         {
1034             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1035             {
1036                 \pdfannot_dict_remove:nn
1037                 {link/##2}
1038                 { BS }
1039             }
1040         }
1041         {
1042             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1043             {
1044                 \pdfannot_dict_put:nnn
1045                 {link/##2}
1046                 { BS }
1047                 { <<##1>> }
1048             }
1049         }
1050     }
1051     ,pdfborderstyle .initial:n = {},
1052 }

```

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.

```

1053 \cs_new_protected:Npn \__hyp_ocg_init:
1054 {
1055     \pdf_object_new:nn { l__hyp_ocg_view_dict_obj } { dict }

```

```

1056 \pdf_object_new:nn { l__hyp_ocg_print_dict_obj } { dict }
1057 \pdf_object_new:nn { l__hyp_ocg_config_dict_obj } { dict }
1058 \pdf_object_new:nn { l__hyp_ocg_ref_array_obj } { array }
1059 \pdf_object_write:nx { l__hyp_ocg_ref_array_obj }
1060 {
1061   \pdf_object_ref:n { l__hyp_ocg_view_dict_obj }
1062   \c_space_tl
1063   \pdf_object_ref:n { l__hyp_ocg_print_dict_obj }
1064 }
1065 \pdf_object_write:nn { l__hyp_ocg_view_dict_obj }
1066 {
1067   /Type/OCG
1068   /Name(View)
1069   /Usage
1070   <<
1071     /Print <</PrintState/OFF>>~
1072     /View <</ViewState/ON >>~
1073   >>
1074 }
1075 \pdf_object_write:nn { l__hyp_ocg_print_dict_obj }
1076 {
1077   /Type/OCG
1078   /Name(Print)
1079   /Usage
1080   <<
1081     /Print <</PrintState/ON>>~
1082     /View <</ViewState/OFF>>~
1083   >>
1084 }
1085 \pdfmanagement_add:nxx { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {l__hyp_ocg_ref_array_obj} }
1086 \pdfmanagement_add:nxx { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {l__hyp_ocg_print_dict_obj} }
1087 \pdf_object_write:nx { l__hyp_ocg_config_dict_obj }
1088 {
1089   /OFF[\pdf_object_ref:n { l__hyp_ocg_print_dict_obj }]
1090   /AS[
1091     <<
1092       /Event/View
1093       /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
1094       /Category[/View]
1095     >>
1096     <<
1097       /Event/Print
1098       /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
1099       /Category[/Print]
1100     >>
1101     <<
1102       /Event/Export
1103       /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
1104       /Category[/Print]
1105     >>
1106   ]
1107 }
1108 \pdfmanagement_add:nxx { Catalog / OCProperties }{ D }{ \pdf_object_ref:n { l__hyp_ocg_ref_array_obj} }
1109 \cs_gset:Npn \__hyp_ocg_init: {}

```

1110 }

(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.

```

1111 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1112 {
1113   \hook_gput_code:nnn
1114     {pdfannot/link/#2/begin}
1115     {hyp/ocg}
1116     {
1117       \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1118       {
1119         \_hyp_ocg_init:
1120         \group_begin:
1121         \hbox_set:Nw \l__hyp_tmpa_box
1122       }
1123     }
1124   \hook_gput_code:nnn
1125     {pdfannot/link/#2/end}
1126     {hyp/ocg}
1127     {
1128       \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1129       {
1130         \hbox_set_end:
1131         \mbox
1132         {
1133           \pdf_bdcobject:nn {OC}{l__hyp_ocg_print_dict_obj}
1134           \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1135           \pdf_emc:
1136           \pdf_bdcobject:nn {OC}{l__hyp_ocg_view_dict_obj}
1137           \group_begin:
1138           \color_select:n { hyp/color/#1 }
1139           \box_use_drop:N \l__hyp_tmpa_box
1140           \group_end:
1141           \pdf_emc:
1142         }
1143       }
1144     }
1145   }
1146 }
```

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

```

ocgcolorlink (setup key) 1147 \pdf_version_compare:NnTF < {1.5}
ocgcolorurl (setup key) 1148 {
ocgcolorfile (setup key) 1149   \keys_define:nn { hyp / setup }
ocgcolormenu (setup key) 1150   {
ocgcolorrun (setup key) 1151     ,_ocgcolorlinks .code:n =
1152     {
1153       \msg_warning:nnxx
1154       { hyp }
1155       { ignore-deprecated-or-unknown-option-in-pdf-version }
1156       { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1157     }

```

```

1158     }
1159   }
1160   {
1161     \keys_define:nn { hyp / setup }
1162     {
1163       ,_ocgcolorlinks .meta:n =
1164       {
1165         ocgcolorlink=#1,
1166         ocgcolorurl=#1,
1167         ocgcolorfile=#1,
1168         ocgcolorrun=#1,
1169         ocgcolormenu=#1
1170       }
1171       ,_ocgcolorlinks .default:n = true
1172     }
1173   }
1174
1175   \keys_define:nn { hyp / setup }
1176   {
1177     ,ocgcolorlinks .choice:
1178     ,ocgcolorlinks / true .meta:n =
1179     {
1180       pdfborder      = {0~0~0},
1181       pdfborderstyle = {},
1182       colorlinks     = false,
1183       _ocgcolorlinks = true
1184     }
1185     ,ocgcolorlinks / false .meta:n =
1186     {
1187       _ocgcolorlinks = false
1188     }
1189     ,ocgcolorlinks .default:n = {true}
1190   }
1191
1192   \seq_map_inline:Nn \c__hyp_annot_types_seq
1193   {
1194     \pdf_version_compare:NnTF < {1.5}
1195     {
1196       \keys_define:nn { hyp / setup }
1197       {
1198         ,ocgcolor#1 .code:n=
1199         {
1200           \msg_warning:nnxx
1201           { hyp }
1202           { ignore-deprecated-or-unknown-option-in-pdf-version }
1203           { ocgcolor#1 }
1204           { \pdf_version_major:.\pdf_version_minor: }
1205         }
1206       }
1207     }
1208     {
1209       \keys_define:nn { hyp / setup }
1210       {
1211         ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }

```

```

1212     }
1213 }
1214 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1215 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1216 {
1217   \keys_define:nn { hyp / setup }
1218   {
1219     ,#1highlight .choices:nn =
1220     { /I, /N, /O, /P}
1221     {
1222       \pdfannot_dict_put:nnn
1223       {link/#2}
1224       { H }
1225       { ##1 }
1226     }
1227     ,#1highlight / .code:n =
1228     {
1229       \pdfannot_dict_remove:nn
1230       {link/#2}
1231       { H }
1232     }
1233     }
1234     ,#1highlight / unknown .code:n =
1235     {
1236       \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1237       { pdfhighlight }
1238       { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1239       { \exp_not:n {##1} }
1240     }
1241   }
1242 }
1243 }
1244
1245
1246 \keys_define:nn { hyp / setup }
1247 {
1248   ,pdfhighlight .choices:nn =
1249   { /I, /N, /O, /P}
1250   {
1251     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1252     {
1253       \pdfannot_dict_put:nnn
1254       {link/####2}
1255       { H }
1256       { #1 }
1257     }
1258   }
1259   ,pdfhighlight / .code:n =
1260   {
1261     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop

```

```

1262         {
1263             \pdfannot_dict_remove:nn
1264             {link/##2}
1265             { H }
1266         }
1267     }
1268     ,pdfhighlight .initial:n = {/I},
1269     ,pdfhighlight / unknown .code:n =
1270     {
1271         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1272         { pdfhighlight }
1273         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1274         { \exp_not:n {#1} }
1275     }
1276 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1277 \keys_define:nn { hyp / setup }
hideurl (setup key) 1278 {
hidefile (setup key) 1279     hidelinks .meta:n =
hiderun (setup key) 1280     {
hidemenu (setup key) 1281         ,colorlinks      = false
1282         ,ocgcolorlinks = false
1283         ,pdfborder      = { 0~0~0 }
1284         ,pdfborderstyle=
1285     }
1286 }
1287
1288 \seq_map_inline:Nn \c__hyp_annot_types_seq
1289 {
1290     \keys_define:nn { hyp / setup }
1291     {
1292         hide#1 .meta:n =
1293         {
1294             ,color#1      = false
1295             ,ocgcolor#1   = false
1296             ,#1border     = { 0~0~0 }
1297             ,#1borderstyle =
1298         }
1299     }
1300 }

```

11.7 color schemes and settings

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

```

colorscheme (setup key)
1301 \keys_define:nn { hyp / setup }
1302 {
1303     colorscheme .code:n =

```

```

1304     {
1305         \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1306         {
1307             \keys_set:nn { hyp /setup }
1308             {
1309                 ##1 = ##2
1310             }
1311         }
1312     }
1313 }
1314 \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) 1315 \keys_define:nn { hyp / setup }
pdfversion (setup key) 1316 {
1317     ,unicode .code:n = {}
1318     ,pdfencoding .code:n = {}
1319     ,pdfversion .code:n =
1320     {
1321         \msg_warning:nn { hyp }{ pdfversion-disabled }
1322     }
1323 }
1324 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
debug (setup key) 1325 \keys_define:nn { hyp / setup }
draft (setup key) 1326 {
final (setup key) 1327     ,verbose .choice:
1328     ,verbose / true .code:n = { \Hy@verbosetrue}
1329     ,verbose / false .code:n = { \Hy@verbosefalse}
1330     ,verbose .default:n = {true}
1331     ,debug .meta:n = {verbose=#1}
1332     ,debug .default:n = {true}
1333 }
1334 \keys_define:nn { hyp / setup }
1335 {
1336     ,draft .code:n =
1337     {
1338         \Hy@drafttrue
1339         \PassOptionsToPackage{draft}{bookmark}
1340     }
1341     ,final .code:n =
1342     {

```

```

1343         \Hy@finaltrue
1344         \PassOptionsToPackage{final}{bookmark}
1345     }
1346 }

extension (setup key)
hypertextnames (setup key) 1347 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1348 {
    pageanchor (setup key) 1349 ,extension .tl_set:N = \XR@ext
    linktoc (setup key) 1350 ,extension .initial:n= pdf
    linktocpage (setup key) 1351 ,hypertextnames .choice:
    plainpages (setup key) 1352 ,hypertextnames / true .code:n = { \Hy@hypertextnametrue}
localanchorname (setup key) 1353 ,hypertextnames / false .code:n = { \Hy@hypertextnamesfalse}
linkfileprefix (setup key) 1354 ,hypertextnames .default:n = {true}
1355 ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1356 ,localanchorname .choice:
1357 ,localanchorname / true .code:n = { \Hy@localanchornametrue }
1358 ,localanchorname / false .code:n = { \Hy@localanchornamefalse }
1359 ,localanchorname .default:n = {true}
1360 ,naturalnames .choice:
1361 ,naturalnames / true .code:n = { \Hy@naturalnametrue}
1362 ,naturalnames / false .code:n = { \Hy@naturalnamesfalse}
1363 ,naturalnames .default:n = {true}
1364 ,pageanchor .choice:
1365 ,pageanchor / true .code:n = { \Hy@pageanchortrue}
1366 ,pageanchor / false .code:n = { \Hy@pageanchorfalse}
1367 ,pageanchor .default:n = {true}
1368 ,plainpages .choice:
1369 ,plainpages / true .code:n = { \Hy@plainpagetrue}
1370 ,plainpages / false .code:n = { \Hy@plainpagesfalse}
1371 ,plainpages .default:n = {true}
1372 }
1373
1374 \keys_define:nn { hyp / setup }
1375 {
1376 ,linktoc .choices:nn = { none, section, all, page }
1377 {
1378     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1379 }
1380 ,linktoc / unknown .code:n =
1381 {
1382     \msg_warning:nnxxx { hyp } { unknown-choice }
1383     { linktoc }
1384     { none, section, all, page }
1385     { \exp_not:n {#1} }
1386 }
1387 ,linktocpage .choice:
1388 ,linktocpage / true .meta:n = {linktoc=page}
1389 ,linktocpage / false .meta:n = {linktoc=section}
1390 ,linktocpage .default:n = true
1391 }
1392

```

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

`url (setup key)`

`file (setup key)`

`menu (setup key)`

`run (setup key)`


```

1393 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1394 {
1395   \keys_define:nn { hyp / setup }
1396   {
1397     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1398   }
1399 }

1400 \keys_define:nn { hyp / setup }
1401 {
1402   ,baseurl .code:n =
1403   {
1404     \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1405     \tl_if_empty:NTF \l__hyp_tmpa_tl
1406     {
1407       \pdfmanagement_remove:nn {Catalog} { URI }
1408     }
1409     {
1410       \pdfmanagement_add:nxx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1411     }
1412   }
1413   %only false does something ...
1414   ,bookmarks .choice:
1415   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1416   ,bookmarks / true .code:n = {}
1417   ,bookmarks .default:n = {true}
1418   ,bookmarksnumbered .choice:
1419   ,bookmarksnumbered / false .code:n = { \Hy@bookmarksnumberedfalse }
1420   ,bookmarksnumbered / true .code:n = { \Hy@bookmarksnumberedtrue }
1421   ,bookmarksnumbered .default:n = {true}
1422   ,bookmarksopen .choice:
1423   ,bookmarksopen / false .code:n = { \Hy@bookmarksopenfalse }
1424   ,bookmarksopen / true .code:n = { \Hy@bookmarksopentru }
1425   ,bookmarksopen .default:n = {true}
1426   ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1427   ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1428   ,pdfcenterwindow .choice:
1429   ,pdfcenterwindow / false .code:n =
1430   {
1431     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1432   }
1433   ,pdfcenterwindow / true .code:n =
1434   {
1435     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }{ CenterWindow }{ true }
1436   }
1437   ,pdfcenterwindow / .code:n =
1438   {
1439     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1440   }
1441   ,pdfcenterwindow / unknown .code:n =
1442   {
1443     \msg_warning:nxxx { hyp } { no-bool }
1444     { pdfcenterwindow }
1445     { \exp_not:n {#1} }

```

```

1446     }
1447 ,pdfcenterwindow      .default:n = true
1448 ,pdfdirection         .choice:
1449 ,pdfdirection / L2R .code:n =
1450 {
1451     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1452 }
1453 ,pdfdirection / R2L .code:n =
1454 {
1455     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1456 }
1457 ,pdfdirection /      .code:n =
1458 {
1459     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1460 }
1461 ,pdfdirection / unknown .code:n =
1462 {
1463     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1464     { pdfdirection }
1465     { L2R , R2L }
1466     { \exp_not:n {#1} }
1467 }
1468 ,pdfdisplaydoctitle   .choice:
1469 ,pdfdisplaydoctitle / false .code:n =
1470 {
1471     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1472 }
1473 ,pdfdisplaydoctitle / true .code:n =
1474 {
1475     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1476 }
1477 ,pdfdisplaydoctitle   .default:n = true
1478 ,pdfduplex            .choices:nn =
1479 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1480 {
1481     \pdf_version_compare:NnTF > {1.6}
1482     {
1483         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1484         { PrintDuplex } { /#1 }
1485     }
1486     {
1487         \msg_warning:nnxx
1488         {hyp}
1489         {ignore-deprecated-or-unknown-option-in-pdf-version}
1490         {pdfduplex}
1491         {\pdf_version:}
1492     }
1493 }%
1494 ,pdfduplex /          .code:n =
1495 {
1496     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1497 }
1498 ,pdfduplex / unknown .code:n =
1499 {

```

```

1500     \msg_warning:nxxx { hyp } { unknown-choice+empty }
1501     { pdfduplex }
1502     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1503     { \exp_not:n {#1} }
1504 }
1505 ,pdffitwindow .choice:
1506 ,pdffitwindow / false .code:n =
1507 {
1508     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1509 }
1510 ,pdffitwindow / true .code:n =
1511 {
1512     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1513 }
1514 ,pdffitwindow / .code:n =
1515 {
1516     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1517 }
1518 ,pdffitwindow .default:n = true
1519 ,pdffitwindow / unknown .code:n =
1520 {
1521     \msg_warning:nxxx { hyp } { no-bool }
1522     { pdffitwindow }
1523     { \exp_not:n {#1} }
1524 }
1525 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1526 ,pdflinkmargin .initial:n = {1pt}
1527 ,pdfmenubar .choice:
1528 ,pdfmenubar / true .code:n =
1529 {
1530     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1531 }
1532 ,pdfmenubar / false .code:n =
1533 {
1534     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1535     { HideMenubar } { true }
1536 }
1537 ,pdfmenubar / .code:n =
1538 {
1539     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1540 }
1541 ,pdfmenubar .default:n = true
1542 ,pdfmenubar / unknown .code:n =
1543 {
1544     \msg_warning:nxxx { hyp } { no-bool }
1545     { pdfmenubar }
1546     { \exp_not:n {#1} }
1547 }
1548 ,pdfnewwindow .choice:
1549 ,pdfnewwindow / true .code:n =
1550 {
1551     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1552     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1553 }

```

```

1554 ,pdfnewwindow / false .code:n =
1555 {
1556   \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1557   \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1558 }
1559 ,pdfnewwindow / .code:n =
1560 {
1561   \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1562   \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1563 }
1564 ,pdfnonfullscreenpagemode .choices:nn =
1565 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1566 {
1567   \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1568   { NonFullScreenPageMode } {/#1}
1569 }
1570 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1571 {
1572   \pdf_version_compare:NnTF < {1.6}
1573   {
1574     %message
1575   }
1576   {
1577     \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1578     {NonFullScreenPageMode}{/UseAttachments}
1579   }
1580 }
1581 ,pdfnonfullscreenpagemode / .code:n =
1582 {
1583   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1584 }
1585 ,pdfnonfullscreenpagemode / unknown .code:n =
1586 {
1587   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1588   { pdfnonfullscreenpagemode }
1589   { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1590   { \exp_not:n {#1} }
1591 }
1592 ,pdfnumcopies .code:n =
1593 {
1594   \pdf_version_compare:NnTF > {1.6}
1595   {
1596     \tl_if_empty:NTF {#1}
1597     {
1598       \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1599     }
1600     {
1601       \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1602       {NumCopies}{#1}
1603     }
1604   }
1605   {
1606     \msg_warning:nnxx
1607     {hyp}

```

```

1608         {ignore-deprecated-or-unknown-option-in-pdf-version}
1609         {pdfnumcopies}
1610         {\pdf_version:}
1611     }
1612 }
1613 ,pdfpagelayout .choices:nn =
1614 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1615 { \pdfmanagement_add:nnx {Catalog} { PageLayout }{ /#1 } }
1616 ,pdfpagelayout / .code:n =
1617 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1618 ,pdfpagelayout / unknown .code:n =
1619 {
1620     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1621     { pdfpagelayout }
1622     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1623     { \exp_not:n {#1} }
1624 }
1625 ,pdfpagemode .choices:nn =
1626 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1627 { \pdfmanagement_add:nnx {Catalog} { PageMode }{ /#1 } }
1628 ,pdfpagemode / UseAttachments .code:n =
1629 {
1630     \pdf_version_compare:NnTF > {1.5}
1631     {
1632         \pdfmanagement_add:nnx {Catalog} { PageMode }{ /UseAttachments }
1633     }
1634     {
1635         \msg_warning:nnxx
1636         {hyp}
1637         {ignore-deprecated-or-unknown-value-in-pdf-version}
1638         {UseAttachments}
1639         {\pdf_version:}
1640     }
1641 }
1642 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1643 ,pdfpagemode / unknown .code:n =
1644 {
1645     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1646     { pdfpagemode }
1647     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1648     { \exp_not:n {#1} }
1649 }
1650 ,pdfpagescrop .code:n =
1651 {
1652     \tl_if_empty:nTF %or blank?
1653     {
1654         \pdfmanagement_remove:nn {Pages} { CropBox }
1655     }
1656     {
1657         \pdfmanagement_add:nnx {Pages} { CropBox } { [#1] }
1658     }
1659 }
1660 ,pdfpicktraybypdfsize .choice:
1661 ,pdfpicktraybypdfsize / true .code:n =

```

```

1662 {
1663   \pdf_version_compare:NnTF > {1.6}
1664   {
1665     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1666     { PickTrayByPDFSize } { true }
1667   }
1668   {
1669     \msg_warning:nnxx
1670     {hyp}
1671     {ignore-deprecated-or-unknown-option-in-pdf-version}
1672     {pdfpicktraybypdfsize}
1673     {\pdf_version:}
1674   }
1675 }
1676 ,pdfpicktraybypdfsize / false .code:n =
1677 {
1678   \pdf_version_compare:NnTF > {1.6}
1679   {
1680     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1681     { PickTrayByPDFSize } { false }
1682   }
1683   {
1684     \msg_warning:nnxx
1685     {hyp}
1686     {ignore-deprecated-or-unknown-option-in-pdf-version}
1687     {pdfpicktraybypdfsize}
1688     {\pdf_version:}
1689   }
1690 }
1691 ,pdfpicktraybypdfsize / .code:n =
1692 {
1693   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1694 }
1695 ,pdfpicktraybypdfsize / unknown .code:n =
1696 {
1697   \msg_warning:nnxx { hyp } { no-bool }
1698   { picktraybypdfsize }
1699   { \exp_not:n {#1} }
1700 }
1701 ,pdfprintarea .choices:nn =
1702 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1703 {
1704   \pdf_version_compare:NnTF < {2.0}
1705   {
1706     \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1707     { PrintArea } { /#1 }
1708   }
1709   {
1710     \msg_warning:nnxx
1711     {hyp}
1712     {ignore-deprecated-or-unknown-option-in-pdf-version}
1713     {pdfprintarea}
1714     {\pdf_version:}
1715   }

```

```

1716     }%
1717 ,pdfprintarea / .code:n =
1718 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1719 ,pdfprintarea / unknown .code:n =
1720 {
1721     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1722     { pdfprintarea }
1723     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1724     { \exp_not:n {#1} }
1725 }
1726 ,pdfprintclip .choices:nn =
1727 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1728 {
1729     \pdf_version_compare:NnTF < {2.0}
1730     {
1731         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1732         { PrintClip } { /#1 }
1733     }
1734     {
1735         \msg_warning:nnxx
1736         {hyp}
1737         {ignore-deprecated-or-unknown-option-in-pdf-version}
1738         {pdfprintclip}
1739         {\pdf_version:}
1740     }
1741 }%
1742 ,pdfprintclip / .code:n =
1743 {
1744     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1745 }
1746 ,pdfprintclip / unknown .code:n =
1747 {
1748     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1749     { pdfprintclip }
1750     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1751     { \exp_not:n {#1} }
1752 }
1753 ,pdfprintpagerange .code:n =
1754 {
1755     \pdf_version_compare:NnTF > {1.6}
1756     {
1757         \tl_if_empty:nTF { #1}
1758         {
1759             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1760             { PrintPageRange }
1761         }
1762         {
1763             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1764             {PrintPageRange}{[#1]}
1765         }
1766     }
1767     {
1768         \msg_warning:nnxx
1769         {hyp}

```

```

1770         {ignore-deprecated-or-unknown-option-in-pdf-version}
1771         {pdfprintpagerange}
1772         {\pdf_version:}
1773     }
1774 }
1775 ,pdfprintscaling .choices:nn =
1776 { None, AppDefault }
1777 {
1778     \pdf_version_compare:NnTF > {1.5}
1779     {
1780         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1781         { PrintScaling } { /#1 }
1782     }
1783     {
1784         \msg_warning:nnxx
1785         {hyp}
1786         {ignore-deprecated-or-unknown-option-in-pdf-version}
1787         {pdfprintscaling}
1788         {\pdf_version:}
1789     }
1790 }%
1791 ,pdfprintscaling / .code:n =
1792 {
1793     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1794 }
1795 ,pdfprintscaling / unknown .code:n =
1796 {
1797     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1798     { pdfprintarea }
1799     { None, AppDefault }
1800     { \exp_not:n {#1} }
1801 }
1802 ,pdfremotestartview .code:n =
1803 {
1804     \tl_set:Nx \l__hyp_tmpa_tl {#1-null-null-null~}
1805     \exp_args:NNV
1806     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1807     {
1808         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1809     }
1810     {
1811         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1812         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1813     }
1814 }
1815 ,pdfremotestartview .initial:n = {Fit}
1816 % pdfstartpage is special as it shares code with pdfstartview
1817 ,pdfstartpage .code:n =
1818 {
1819     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1820     \bool_if:nTF
1821     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1822     {
1823         \pdfmanagement_remove:nn {Catalog} { OpenAction }

```



```

1824     }
1825     {
1826         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1827         {
1828             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1829         }
1830     }
1831 }
1832 ,pdfstartpage .initial:n =1
1833 ,pdfstartview .code:n =
1834 {
1835     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~}
1836     \exp_args:NNV
1837     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1838     {
1839         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1840     }
1841     {
1842         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1843         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1844     }
1845     \bool_if:NTF
1846     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1847     {
1848         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1849     }
1850     {
1851         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1852         {
1853             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1854         }
1855     }
1856 }
1857 ,pdfstartview .initial:n = Fit
1858 ,pdftoolbar .choice:
1859 ,pdftoolbar / true .code:n =
1860 {
1861     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1862 }
1863 ,pdftoolbar / false .code:n =
1864 {
1865     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
1866     { HideToolbar } { true }
1867 }
1868 ,pdftoolbar / true .code:n =
1869 {
1870     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1871 }
1872 ,pdftoolbar .default:n = true
1873 ,pdftoolbar / unknown .code:n =
1874 {
1875     \msg_warning:nnxx { hyp } { no-bool }
1876     { pdftoolbar }
1877     { \exp_not:n {#1} }

```

```

1878     }
1879 % pdfview see below.
1880 ,pdfviewarea .choices:nn =
1881 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1882 {
1883   \pdf_version_compare:NnTF < {2.0}
1884   {
1885     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1886     { ViewArea } { /#1 }
1887   }
1888   {
1889     \msg_warning:nnxx
1890     {hyp}
1891     {ignore-deprecated-or-unknown-option-in-pdf-version}
1892     {pdfviewarea}
1893     {\pdf_version:}
1894   }
1895 }%
1896 ,pdfviewarea / .code:n =
1897 {
1898   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1899 }
1900 ,pdfviewarea / unknown .code:n =
1901 {
1902   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1903   { pdfviewarea }
1904   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1905   { \exp_not:n {#1} }
1906 }
1907 ,pdfviewclip .choices:nn =
1908 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1909 {
1910   \pdf_version_compare:NnTF < {2.0}
1911   {
1912     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1913     { ViewClip } { /#1 }
1914   }
1915   {
1916     \msg_warning:nnxx
1917     {hyp}
1918     {ignore-deprecated-or-unknown-option-in-pdf-version}
1919     {pdfviewclip}
1920     {\pdf_version:}
1921   }
1922 }%
1923 ,pdfviewclip / .code:n =
1924 {
1925   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
1926 }
1927 ,pdfviewclip / unknown .code:n =
1928 {
1929   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1930   { pdfviewclip }
1931   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }

```

```

1932     { \exp_not:n {#1} }
1933   }
1934   ,pdfwindowui .choice:
1935   ,pdfwindowui / true .code:n =
1936   {
1937     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
1938   }
1939   ,pdfwindowui / false .code:n =
1940   {
1941     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1942     { HideWindowUI } { true }
1943   }
1944   ,pdfwindowui / .code:n =
1945   {
1946     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
1947   }
1948   ,pdfwindowui / unknown .code:n =
1949   {
1950     \msg_warning:nnxx { hyp } { no-bool }
1951     { pdfwindowui }
1952     { \exp_not:n {#1} }
1953   }
1954   ,pdfwindowui .default:n = true
1955 }

```

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

```

1956 \keys_define:nn { hyp / setup }
1957 {
1958   ,pdfview .code:n =
1959   {
1960     \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
1961     \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } } {
1962       { xyz }
1963       {
1964         \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
1965         {
1966           \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
1967           \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
1968           {
1969             \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1970           }
1971           {
1972             \tl_set:Nx \l__hyp_dest_pdfview_tl
1973             {
1974               \fp_eval:n { \l__hyp_tmpa_tl * 100 }
1975             }
1976           }
1977         }
1978       }
1979       {
1980         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1981       }
1982     }
1983     { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }

```

```

1984 { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
1985 { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh}}
1986 { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv}}
1987 { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
1988 { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
1989 { fitr }
1990 {
1991   \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
1992   {
1993     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
1994   }
1995   {
1996     %ensure 4 values ...
1997     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
1998     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1999     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2000     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2001     \hbox_set_to_wd:Nnn \l__hyp_dest_box
2002     {
2003       \fp_eval:n
2004       {
2005         round
2006         (
2007           abs
2008           (
2009             \seq_item:Nn\l__hyp_tmpa_seq{4}
2010             -
2011             (\seq_item:Nn\l__hyp_tmpa_seq{2})
2012           ),
2013           3
2014         )
2015       }bp
2016     }{}
2017     \box_set_dp:Nn \l__hyp_dest_box
2018     {
2019       \fp_eval:n
2020       {
2021         round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2022       }bp
2023     }
2024     \box_set_ht:Nn \l__hyp_dest_box
2025     {
2026       \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2027     }
2028   }
2029 }
2030 }
2031 {
2032   \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2033   \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2034 }
2035 }
2036 ,pdfview .initial:n = {xyz}
2037 }

```

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.

```
2038 \keys_define:nn { hyp / setup }
2039 {
2040   ,pdflang .code:n =
2041   {
2042     \tl_if_empty:nTF { #1 }
2043     {
2044       \pdfmanagement_remove:nn {Catalog} { Lang }
2045     }
2046     {
2047       \pdfmanagement_add:nnx {Catalog} { Lang } { (#1) }
2048     }
2049     \__hyp_store_metadata:nn {pdflang}{#1}
2050   }
2051 }
```

12.3.1 “info keys”

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

```
pdfcreator (setup key) 2052 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
pdfsubject (setup key) 2053 {
pdfproducer (setup key) 2054   \keys_define:nn { hyp / setup }
pdfkeywords (setup key) 2055   {
2056     pdf#1 .code:n =
2057     {
2058       \tl_if_blank:nTF {##1}
2059       {
2060         \pdfmanagement_remove:nn {Info}{#2}
2061       }
2062       {
2063         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2064         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2065         {
2066           \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2067         }
2068       }
2069       \__hyp_store_metadata:nn {pdf#1}{##1}
2070     }
2071   }
2072   \keys_define:nn { hyp / info }
2073   {
2074     #2 .code:n =
2075     {
2076       \tl_if_blank:nTF {##1}
2077       {
2078         \pdfmanagement_remove:nn {Info}{#2}
2079       }
2080       {
```

```

2081         \_hyp_text_pdfstring_info:nN {##1}\l_hyp_tmpa_str
2082         \str_if_eq:VnF\l_hyp_tmpa_str{<FEFF>}
2083         {
2084             \pdfmanagement_add:nxx {Info}{#2}{\l_hyp_tmpa_str}
2085         }
2086     }
2087     \exp_args:Nx \_hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2088 }
2089 ,unknown .code:n =
2090 {
2091     \_hyp_text_pdfstring_info:nN {##1}\l_hyp_tmpa_str
2092     \str_if_eq:VnF\l_hyp_tmpa_str{<FEFF>}
2093     {
2094         \exp_args:Nno
2095         \pdfmanagement_add:nxx {Info}
2096         { \l_keys_key_str } {\l_hyp_tmpa_str}
2097     }
2098 }
2099 }
2100 }
2101 \_hyp_setup_info_key:nn {author} {Author}
2102 \_hyp_setup_info_key:nn {title} {Title}
2103 \_hyp_setup_info_key:nn {producer} {Producer}
2104 \_hyp_setup_info_key:nn {creator} {Creator}
2105 % ignored key: addtopdfcreator
2106 \_hyp_setup_info_key:nn {subject} {Subject}
2107 \_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.

```

2108 \cs_new_protected:Npn \_hyp_setup_info_date_key:nn #1 #2
2109 {
2110     \keys_define:nn { hyp / setup }
2111     {
2112         pdf#1 .code:n =
2113         {
2114             \tl_if_blank:nTF {##1}
2115             {
2116                 \pdfmanagement_remove:nn {Info}{#2}
2117             }
2118             {
2119                 \pdfmanagement_add:nxx {Info}{#2}{(\tl_to_str:n {##1})}
2120             }
2121             \_hyp_store_metadata:nn {pdf#1}{##1}
2122         }
2123     }
2124     \keys_define:nn { hyp / info }
2125     {
2126         #2 .code:n =
2127         {
2128             \tl_if_blank:nTF {##1}
2129             {
2130                 \pdfmanagement_remove:nn {Info}{#2}
2131             }

```

```

2132         {
2133             \pdfmanagement_add:nx {Info}{#2}{(\tl_to_str:n {##1})}
2134         }
2135         \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2136     }
2137 }
2138 }
2139
2140 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2141 \__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 ...

```

2142 \keys_define:nn { hyp / setup }
2143 {
2144     ,pdftrapped .code:n =
2145     {
2146         \exp_args:Nne
2147         \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2148     }
2149     ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2150     {
2151         \pdfmanagement_add:nx {Info}{Trapped}
2152         {/
2153             \str_uppercase:f { \str_head:n { #1 } }
2154             \str_lowercase:f { \str_tail:n { #1 } }
2155         }
2156         \__hyp_store_metadata:nx {pdftrapped}
2157         {
2158             \str_uppercase:f { \str_head:n { #1 } }
2159             \str_lowercase:f { \str_tail:n { #1 } }
2160         }
2161     }
2162     ,_pdftrapped / unknown .code:n =
2163     {
2164         \msg_warning:nxxx { hyp } { unknown-choice }
2165         { pdftrapped }
2166         { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2167         { \exp_not:n {#1} }
2168     }
2169 }

```

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

```

2170 \keys_define:nn { hyp / setup }
2171 {
2172     pdfinfo .code:n =
2173     {
2174         \keys_set:nn { hyp / info } { #1 }
2175     }
2176 }

```

Now we set some default values

```

2177 \keys_set:nn { hyp / setup } {pdfcreator = LaTeX-with-hyperref}
2178 \keys_set:nn { hyp / setup } {pdfauthor = }
2179 \keys_set:nn { hyp / setup } {pdftitle = }
2180 \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:

```
2181 \clist_map_inline:nn
2182 {
2183   ,pdfcopyright
2184   ,pdftype
2185   ,pdflicenseurl
2186   ,pdfauthortitle
2187   ,pdfcaptionwriter
2188   ,pdfmetalang
2189   ,pdfapart
2190   ,pdfaconformance
2191   ,pdfuapart
2192   ,pdfxstandard
2193   ,pdfsource
2194   ,pdfdocumentid
2195   ,pdfinstanceid
2196   ,pdfversionid
2197   ,pdfrendition
2198   ,pdfpublication
2199   ,pdfpubtype
2200   ,pdfbytes
2201   ,pdfnumpages
2202   ,pdfissn
2203   ,pdfeissn
2204   ,pdfisbn
2205   ,pdfbookedition
2206   ,pdfpublisher
2207   ,pdfvolumenum
2208   ,pdfissuenum
2209   ,pdfpagerange
2210   ,pdfdoi
2211   ,pdfurl
2212   ,pdfidentifier
2213   ,pdfsubtitle
2214   ,pdfpubstatus
2215   ,pdfcontactaddress
2216   ,pdfcontactcity
2217   ,pdfcontactregion
2218   ,pdfcontactpostcode
2219   ,pdfcontactcountry
2220   ,pdfcontactphone
2221   ,pdfcontactemail
2222   ,pdfcontacturl
2223   ,pdfdate
2224 }
2225 {
2226   \keys_define:nn { hyp / setup }
2227   {
2228     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2229   }
2230 }
```


12.5 Transitions

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

```
2231 \keys_define:nn { hyp / setup }
2232 {
2233   pdfpageduration .code:n =
2234   {
2235     \tl_if_blank:nTF { #1 }
2236     {
2237       \pdfmanagement_remove:nn {Page}{Dur}
2238     }
2239     {
2240       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2241     }
2242   }
2243 }
```

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 W 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

The keys allowed in the argument of pdfpagetransition are

```
2244 \keys_define:nn { hyp / setup }
2245 {
2246   pdfpagetransition .code:n =
2247   {
2248     \tl_if_blank:nTF {#1}
2249     {
2250       \pdfmanagement_remove:nn {Page}{Trans}
2251     }
2252     {
```

```

2253         \group_begin:
2254         \keys_set:nn { hyp / trans }{style=R,#1}
2255         \pdf_object_unnamed_write:nx { dict }
2256         {
2257             \pdfdict_use:n {l__hyp_page/Trans}
2258         }
2259         \pdfmanagement_add:nnx {Page}{Trans}{\pdf_object_ref_last:}
2260         \group_end:
2261     }
2262 }
2263 }
2264 \keys_define:nn { hyp / trans }
2265 {
2266     ,style .choices:nn =
2267     {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2268     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2269     ,style / unknown .code:n =
2270     {
2271         \msg_warning:nnxxx { hyp } { unknown-choice }
2272         { trans / style }
2273         { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2274         { \exp_not:n {#1} }
2275     }
2276     ,duration .code:n =
2277     {
2278         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}
2279     }
2280     ,direction .choices:nn =
2281     {H,V}
2282     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2283     ,direction .choices:nn =
2284     {0,90,180,270,315}
2285     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ #1 } }
2286     ,direction / None .code:n =
2287     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ /None } }
2288     ,direction / unknown .code:n =
2289     {
2290         \msg_warning:nnxxx { hyp } { unknown-choice }
2291         { trans / direction }
2292         {
2293             H~(horizontal,~only~Split,~Blinds),
2294             V~(vertical,~only~Split,~Blinds),
2295             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2296             90~(bottom~to~top,~only~Wipe),
2297             180~(right~to~left,~only~Wipe),
2298             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2299             315~(top~left~to~bottom,~only~Glitter),
2300             None~(only~Fly)
2301         }
2302         { \exp_not:n {#1} }
2303     }
2304     ,motion .choices:nn =
2305     {I,0}
2306     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }

```

```

2307 ,motion / unknown .code:n =
2308 {
2309     \msg_warning:nnxxx { hyp } { unknown-choice }
2310     { trans / motion }
2311     { I~(inwards) , O~(outwards) }
2312     { \exp_not:n {#1} }
2313 }
2314 ,scale .code:n =
2315 { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2316 ,opaque .choices:nn = {true,false}
2317 { \pdfdict_put:nnn { l__hyp_page/Trans }{ B } { #1 } }
2318 ,opaque / unknown .code:n =
2319 {
2320     \msg_warning:nnxxx { hyp } { unknown-choice }
2321     { trans / B }
2322     { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2323     { \exp_not:n {#1} }
2324 }
2325 % try to set unknown keys as style
2326 ,unknown .code:n =
2327 {
2328     % warning ...
2329     \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2330 }
2331 }

```

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

```

2332 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}

```

Unfinished Form field code

```

2333 \NewDocumentCommand \MakeFieldObject { m m }
2334 {
2335     \pdfxform_new:nnn { #2 }{ } { #1 }
2336 }
2337
2338
2339 \prop_new:N    \g__hyp_AcroForm_CoFields_prop
2340 \prop_new:N    \g__hyp_AcroForm_Fields_prop
2341
2342 \let\HyField@afields\ltx@empty
2343 \let\HyField@cofields\ltx@empty
2344 %% UF test for old pdftex removed
2345 %\let\HyField@AuxAddToFields\ltx@gobble
2346 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2347 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2348
2349 % the value doesn't matter, but with a prop we avoid duplicates and it is
2350 % clearly faster than removing them from a sequence
2351 \def\HyField@AuxAddToFields#1
2352 {
2353     \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2354 }%
2355
2356 %fields with empty key get a value too -- lets hope that
2357 %this give the expected behaviour

```

```

2358 \def\HyField@AuxAddToCoFields #1 #2
2359 {
2360   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2361 }
2362
2363 \Hy@AtBeginDocument
2364 {
2365   \if@filesw
2366     \immediate\write\@mainaux{%
2367       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2368     }%
2369     \immediate\write\@mainaux{%
2370       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2371     }%
2372   \fi
2373   \let\HyField@AfterAuxOpen\@firstofone
2374 }%
2375
2376 \def\HyField@AddToFields
2377 {
2378   \exp_args:Nx\HyField__hypAddToFields
2379   {
2380     \pdfannot_box_ref_last:
2381   }
2382   \ifx\Fld@calculate@code\ltx@empty
2383   \else
2384     \begingroup
2385     \Hy@safe@activetrue
2386     \edef\Hy@temp{%
2387       \endgroup
2388       \if@filesw
2389         \write\@mainaux
2390         {
2391           \string\HyField@AuxAddToCoFields
2392           {
2393             \Fld@calculate@sortkey
2394           }
2395           {
2396             \pdfannot_box_ref_last:
2397           }
2398         }
2399       \fi
2400     }%
2401     \Hy@temp
2402   \fi
2403 }%
2404
2405 \def\HyField__hypAddToFields#1{
2406   \HyField@AfterAuxOpen{%
2407     \if@filesw
2408       \write\@mainaux{%
2409         \string\HyField@AuxAddToFields{#1}%
2410       }%
2411     \fi

```

```

2412     }%
2413 }%
2414
2415 \ExplSyntaxOff
2416 \ExplSyntaxOn
2417 \tl_new:N \l__hyp_CheckmarkYes_tl
2418 \tl_set:Nn \l__hyp_CheckmarkYes_tl { __hyp_xform_CheckMarkYes }
2419 \tl_new:N \l__hyp_CheckmarkOff_tl
2420 \tl_set:Nn \l__hyp_CheckmarkOff_tl { __hyp_xform_CheckMarkOff }
2421
2422 \def\@Form[#1]
2423 {
2424   \@ifundefined{textcolor}{\let\textcolor\@gobble}{ }
2425   \kvsetkeys{Form}{#1}
2426   \pdf@ifdraftmode{ }
2427   {
2428     \Hy@FormObjects
2429     \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2430     {
2431       \pdfmanagement_add:nnx { Catalog / AcroForm } { Fields }{##1}
2432       %\pdfmanagement_show:n { Catalog / AcroForm }
2433     }
2434     \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2435     {
2436       \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2437       {
2438         \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2439       }
2440       \seq_sort:Nn \l__hyp_tmpa_seq
2441       {
2442         \int_compare:nNnTF { \pdf@strcmp{##1}{##2} } > { 0 }
2443         { \sort_return_swapped: }
2444         { \sort_return_same: }
2445       }
2446       \seq_map_inline:Nn \l__hyp_tmpa_seq
2447       {
2448         \pdfmanagement_add:nnx { Catalog / AcroForm }
2449         { CO }
2450         {
2451           \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2452         }
2453       }
2454     }
2455     \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2456     {ZaDb} {\pdf_object_ref:n {l__hyp_font_zapfdingbats_obj} }
2457     \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2458     {Helv} {\pdf_object_ref:n {l__hyp_font_helvetica_obj} }
2459     \pdfmanagement_add:nnx {Catalog /AcroForm}
2460     {DA}{(/Helv~10~Tf~0~g)}
2461     \pdfmeta_standard_verify:nTF {form_no_NeedAppearance}
2462     {
2463       \legacy_if:nT { HyField@NeedAppearances }
2464       {
2465         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}

```

```

2466         }
2467     }
2468     {
2469         \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2470     }
2471 }
2472 \MakeFieldObject
2473 {
2474     \group_begin:
2475     \fontfamily{pzd}
2476     \fontencoding{U}
2477     \fontseries{m}
2478     \fontshape{n}
2479     \selectfont
2480     \char123
2481     \group_end:
2482 }
2483 {__hyp_xform_Ding}
2484 \MakeFieldObject
2485 {
2486     \group_begin:
2487     \fontfamily{pzd}
2488     \fontencoding{U}
2489     \fontseries{m}
2490     \fontshape{n}
2491     \selectfont
2492     \phantom{\char123}
2493     \group_end:
2494 }
2495 {__hyp_xform_DingOff}
2496 \MakeFieldObject
2497 {
2498     \group_begin:
2499     \fontfamily{pzd}
2500     \fontencoding{U}
2501     \fontseries{m}
2502     \fontshape{n}
2503     \selectfont
2504     \char51
2505     \group_end:
2506 }
2507 {__hyp_xform_CheckMarkYes}
2508 \MakeFieldObject
2509 {
2510     \group_begin:
2511     \fontfamily{pzd}
2512     \fontencoding{U}
2513     \fontseries{m}
2514     \fontshape{n}
2515     \selectfont
2516     \phantom{\char51} %perhaps xetex needs some small glyph ..
2517     \group_end:
2518 }
2519 {__hyp_xform_CheckMarkOff}

```

```

2520 \MakeFieldObject
2521 {
2522   \fbox{\textcolor{yellow}{\textsf{Submit}}}% %color?
2523 }
2524 {__hyp_xform_Submit}
2525 \MakeFieldObject
2526 {
2527   \fbox{\textcolor{yellow}{\textsf{SubmitP}}}% %color?
2528 }
2529 {__hyp_xform_SubmitP}
2530 }
2531 \ExplSyntaxOff
2532 \let\@endForm\ltx@empty
2533 \let\HyAnn@AbsPageLabel\ltx@empty
2534 \let\Fld@pageobjref\ltx@empty
2535
2536 \ExplSyntaxOn
2537 \newcount\HyAnn@Count
2538 \HyAnn@Count=\ltx@zero
2539 \def\HyAnn@AbsPageLabel
2540 {
2541   \global\advance\HyAnn@Count by\ltx@one
2542   %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2543   %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2544   %\zref@refused{HyAnn@\the\HyAnn@Count}%
2545   \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
2546   \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
2547 }%
2548 \def\Fld@pageobjref
2549 {
2550   \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
2551   {
2552     /P~\pdf_pageobject_ref:n
2553     {
2554       \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
2555     }
2556   }
2557 }
2558 \ExplSyntaxOff
2559 \ExplSyntaxOn
2560 %% check if the attr should be set through
2561 %% hooks.
2562 %% check if options are missing.
2563 \def\@TextField[#1]#2{% parameters, label
2564   \def\Fld@name{#2}%
2565   \let\Fld@default\ltx@empty
2566   \let\Fld@value\@empty
2567   \def\Fld@width{\DefaultWidthofText}%
2568   \def\Fld@height{%
2569     \ifFld@multiline
2570       \DefaultHeightofTextMultiline
2571     \else
2572       \DefaultHeightofText
2573     \fi

```

```

2574 }%
2575 \begingroup
2576   \expandafter\HyField@SetKeys\expandafter{%
2577     \DefaultOptionsofText,#1%
2578   }%
2579   \PDFForm@Name
2580   \HyField@FlagsText
2581   \ifFld@hidden\def\Fld@width{1sp}\fi
2582   \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2583   \LayoutTextField{#2}{%
2584     \leavevmode
2585     \HyAnn@AbsPageLabel
2586     \Hy@escapeform\PDFForm@Text
2587     \pdfannot_box:nnnn
2588       {\Fld@width}
2589       {\Fld@height}
2590       {Opt} %is this correct?
2591       {\PDFForm@Text}
2592     \MakeTextField{\Fld@width}{\Fld@height}
2593     \HyField@AddToFields
2594   }%
2595 \endgroup
2596 }
2597 \providecommand\@curropt{}
2598 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2599   \def\Fld@name{#2}
2600   \let\Fld@default\relax
2601   \let\Fld@value\relax
2602   \def\Fld@width{\DefaultWidthofChoiceMenu}
2603   \def\Fld@height{\DefaultHeightofChoiceMenu}
2604   \begingroup
2605     \Fld@menulength=0 %
2606     \@tempdima\z@
2607     \clist_map_variable:nNn { #3 } \@curropt
2608     %\@for\@curropt:=#3\do
2609     {%
2610       \expandafter\Fld@checkequals\@curropt==\\%
2611       \Hy@StepCount\Fld@menulength
2612       \settowidth{\@tempdimb}{\@currDisplay}%
2613       \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2614     }%
2615   \advance\@tempdima by~15\p@
2616   \begingroup
2617     \HyField@SetKeys{#1}
2618   \edef\x{\endgroup
2619     \noexpand\expandafter
2620     \noexpand\HyField@SetKeys
2621     \noexpand\expandafter{%
2622       \expandafter\noexpand\csname DefaultOptionsof%
2623         \ifFld@radio
2624           Radio%
2625         \else
2626           \ifFld@combo
2627             \ifFld@popdown

```



```

2628         PopdownBox%
2629     \else
2630         ComboBox%
2631     \fi
2632 \else
2633     ListBox%
2634 \fi
2635 \fi
2636 \endcsname
2637 }%
2638 }\x
2639 \HyField@SetKeys{#1}%
2640 \PDFForm@Name
2641 \ifFld@hidden\def\Fld@width{1sp}\fi
2642 \ifx\Fld@value\relax
2643     \let\Fld@value\Fld@default
2644 \fi
2645 \LayoutChoiceField{#2}{%
2646     \ifFld@radio
2647         \HyField@FlagsRadioButton
2648         \__hypRadio{#3}%
2649     \else
2650         \begingroup
2651         \HyField@FlagsChoice
2652         \ifdim\Fld@width<\@tempdima
2653             \ifdim\@tempdima<1cm\@tempdima1cm\fi
2654             \edef\Fld@width{\the\@tempdima}%
2655         \fi
2656         \ifFld@combo
2657         \else
2658             \@tempdima=\the\Fld@menulength\Fld@charsize
2659             \advance\@tempdima by~\Fld@borderwidth bp %
2660             \advance\@tempdima by~\Fld@borderwidth bp %
2661             \edef\Fld@height{\the\@tempdima}%
2662         \fi
2663         \__hypListbox{#3}%
2664     \endgroup
2665 \fi
2666 }%
2667 \endgroup
2668 }
2669 \tl_new:N \l__hyp_RadioYes_tl
2670 \tl_set:Nn \l__hyp_RadioYes_tl { __hyp_xform_Ding }
2671 \def\__hypRadio#1{%
2672     \Fld@listcount=0-%
2673     %\show\Fld@default
2674     \EdefEscapeName\Fld@default{\Fld@default}%
2675     \clist_map_variable:nNn { #1 } \@curropt
2676     %\@for\@curropt:=#1\do
2677     {%
2678         \expandafter\Fld@checkequals\@curropt==\\%
2679         \EdefEscapeName\@currValue{\@currValue}%
2680         \Hy@StepCount\Fld@listcount
2681         \@currDisplay\space

```

```

2682 \leavevmode
2683 \HyAnn@AbsPageLabel
2684 \Hy@escapeform\PDFForm@Radio
2685 \pdfannot_box:nnnn
2686 {\Fld@width}
2687 {\Fld@height}
2688 {Opt} %is this correct?
2689 {
2690 \PDFForm@Radio
2691 /AP
2692 <<
2693 /N
2694 <<
2695 /\currValue\c_space_tl \pdfxform_ref:o {__hyp_xform_Ding}
2696 %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2697 >>
2698 >>
2699 }
2700 {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2701 \int_compare:nNtT { \Fld@listcount} = { 1 }
2702 { \HyField@AddToFields }
2703 \c_space_tl % deliberate space between radio buttons
2704 % to do: --> should be configurable
2705 }%
2706 }
2707
2708 \newcount\Fld@listcount
2709 \def\__hypListbox#1
2710 {
2711 \HyField@PDFChoices{#1}
2712 \mode_leave_vertical:
2713 \HyAnn@AbsPageLabel
2714 \Hy@escapeform\PDFForm@List
2715 \pdf_link_user:nnn
2716 {widget} %perhaps we need more types??
2717 {\PDFForm@List}
2718 {\MakeChoiceField{\Fld@width}{\Fld@height}}
2719 \HyField@AddToFields
2720 }
2721
2722
2723 \def\@PushButton[#1]#2{% parameters, label
2724 \def\Fld@name{#2}%
2725 \group_begin:
2726 \exp_args:No\HyField@SetKeys
2727 {
2728 \DefaultOptionsofPushButton,#1
2729 }
2730 \PDFForm@Name
2731 \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2732 {
2733 \HyField@FlagsPushButton
2734 \legacy_if:nT {Fld@hidden}
2735 {

```

```

2736         \def\Fld@width{1sp}
2737     }
2738     \LayoutPushButtonField
2739     {
2740         \mode_leave_vertical:
2741         \HyAnn@AbsPageLabel
2742         \Hy@escapeform\PDFForm@Push
2743         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2744         \pdfannot_box:nnnn
2745         {\box_wd:N\l_tmpa_box}
2746         {\box_ht:N\l_tmpa_box}
2747         {\box_dp:N\l_tmpa_box} %is this correct?
2748         {\PDFForm@Push}
2749         {\box_use:N\l_tmpa_box}
2750         \HyField@AddToFields
2751     }
2752 }
2753 {
2754     \msg_error:nn { hyp }{ pdfa-no-push-button }
2755     \LayoutPushButtonField
2756     {
2757         \mode_leave_vertical:
2758         \MakeButtonField{#2}
2759     }
2760 }
2761 \group_end:
2762 }
2763
2764 \def\@Submit[#1]#2
2765 {
2766     \def\Fld@width {\DefaultWidthofSubmit}
2767     \def\Fld@height{\DefaultHeightofSubmit}
2768     \group_begin:
2769     \exp_args:No\HyField@SetKeys
2770     {
2771         \DefaultOptionsofSubmit,#1
2772     }
2773     \HyField@FlagsPushButton
2774     \HyField@FlagsSubmit
2775     \legacy_if:nT { Fld@hidden }
2776     {
2777         \def\Fld@width{1sp}
2778     }
2779     \mode_leave_vertical:
2780     \HyAnn@AbsPageLabel
2781     \Hy@escapeform\PDFForm@Submit
2782     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2783     \pdfannot_box:nnnn
2784     {\box_wd:N\l_tmpa_box}
2785     {\box_ht:N\l_tmpa_box}
2786     {\box_dp:N\l_tmpa_box} %is this correct?
2787     {
2788         \PDFForm@Submit
2789         /AP<<

```

```

2790         /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2791         /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2792     >>
2793     }
2794     \HyField@AddToFields
2795     \box_use:N\l_tmpa_box
2796
2797     \group_end:
2798 }
2799
2800 \def\@Reset[#1]#2
2801 {
2802     \def\Fld@width {\DefaultWidthofReset}
2803     \def\Fld@height{\DefaultHeightofReset}
2804     \group_begin:
2805         \exp_args:No\HyField@SetKeys
2806         {
2807             \DefaultOptionsofReset,#1
2808         }
2809     \mode_leave_vertical:
2810     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2811     {
2812         \HyField@FlagsPushButton
2813         \legacy_if:nT { Fld@hidden }
2814         { \def\Fld@width{1sp} }
2815         \HyAnn@AbsPageLabel
2816         \Hy@escapeform\PDFForm@Reset
2817         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2818         \pdfannot_box:nnnn
2819         {\box_wd:N\l_tmpa_box}
2820         {\box_ht:N\l_tmpa_box}
2821         {\box_dp:N\l_tmpa_box} %is this correct?
2822         { \PDFForm@Reset }
2823         \HyField@AddToFields
2824         \box_use:N \l_tmpa_box
2825     }
2826     {
2827         \msg_error:nn { hyp }{ pdfa-no-reset-button }
2828         \MakeButtonField{#2}
2829     }
2830     \group_end:
2831 }
2832
2833 \def\@CheckBox[#1]#2
2834 {% parameters, label
2835     \def\Fld@name{#2}
2836     \def\Fld@default{0}
2837     \group_begin:
2838         \def\Fld@width {\DefaultWidthofCheckBox}
2839         \def\Fld@height{\DefaultHeightofCheckBox}
2840         \exp_args:No\HyField@SetKeys
2841         {
2842             \DefaultOptionsofCheckBox,#1
2843         }

```

```

2844 \PDFForm@Name
2845 \HyField@FlagsCheckBox
2846 \legacy_if:nT { Fld@hidden }
2847 {
2848   \def\Fld@width{1sp}
2849 }
2850 \LayoutCheckField{#2}
2851 {
2852   \mode_leave_vertical:
2853   \HyAnn@AbsPageLabel
2854   \Hy@escapeform\PDFForm@Check
2855   \pdfannot_box:nnnn
2856     {\Fld@width}
2857     {\Fld@height}
2858     {Opt} %is this correct?
2859     {\PDFForm@Check}
2860   \HyField@AddToFields %check if this works with xelatex ...
2861 }
2862 \group_end:
2863 }
2864 \ExplSyntaxOff
2865
2866 %hm. Should a luatex driver use type1 fonts in fields????
2867 \ExplSyntaxOn
2868 \def\Hy@FormObjects
2869 {
2870   \pdf_object_new:nn {l__hyp_encoding_pdffdoc_obj } { dict }
2871   \pdf_object_new:nn {l__hyp_font_zapfdingbats_obj } { dict }
2872   \pdf_object_new:nn {l__hyp_font_helvetica_obj } { dict }
2873   \pdf_object_write:nx {l__hyp_encoding_pdffdoc_obj }
2874   {
2875     /Type/Encoding
2876     /Differences[
2877       24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2878       /ring/tilde
2879       \c_space_tl
2880       39/quotesingle
2881       \c_space_tl
2882       96/grave %
2883       \iow_newline:
2884       128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2885       /fraction/guilsinglleft/guilsinglright/minus/perthousand
2886       /quotedblbase/quotedblleft/quotedblright/quoteleft
2887       /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2888       /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2889       \iow_newline:
2890       164/currency
2891       \c_space_tl
2892       166/brokenbar
2893       \c_space_tl
2894       168/dieresis/copyright/ordfeminine
2895       \c_space_tl
2896       172/logicalnot/.notdef/registered/macron/degree/plusminus
2897       /twosuperior/threesuperior/acute/mu

```

```

2898         \c_space_tl
2899         183/periodcentered/cedilla/onesuperior/ordmasculine
2900         \c_space_tl
2901         188/onequarter/onehalf/threequarters
2902         \iow_newline:
2903         192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
2904         /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
2905         /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
2906         /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
2907         /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
2908         /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
2909         /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
2910         /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
2911         /ocircumflex/otilde/odieresis/divide/oslash/ugrave
2912         /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
2913     ]
2914 }
2915 \pdf_object_write:nn {l__hyp_font_zapfdingbats_obj }
2916 {
2917     /Type/Font
2918     /Subtype/Type1
2919     /Name/ZaDb
2920     /BaseFont/ZapfDingbats
2921 }
2922 \pdf_object_write:nx {l__hyp_font_helvetica_obj }
2923 {
2924     /Type/Font
2925     /Subtype/Type1
2926     /Name/Helv
2927     /BaseFont/Helvetica
2928     /Encoding~\pdf_object_ref:n { l__hyp_encoding_pdfdoc_obj }
2929 }
2930 \global\let\Hy@FormObjects\relax
2931 }
2932 \ExplSyntaxOff
2933 \providecommand*{\Fld@pageobjref}{\}
2934 \ifcsname pdf@escapestring\endcsname
2935     \def\Hy@escapeform#1{%
2936         \ifHy@pdfescapeform
2937             \let\Hy@escapestring\pdfescapestring
2938         \else
2939             \let\Hy@escapestring\@firstofone
2940         \fi
2941     }%
2942     \Hy@escapeform{}%
2943 \else
2944     \let\Hy@escapestring\@firstofone
2945     \def\Hy@escapeform#1{%
2946         \ifHy@pdfescapeform
2947             \def\Hy@escapestring##1{%
2948                 \noexpand\Hy@escapestring{\noexpand##1}%
2949             }%
2950             \edef\Hy@temp{#1}%
2951             \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{\}\@nil

```

```

2952     \def\Hy@escapestring##1{%
2953         \ifundefined{Hy@esc@\string##1}{%
2954             ##1%
2955             \ThisShouldNotHappen
2956         }{%
2957             \csname Hy@esc@\string##1\endcsname
2958         }%
2959     }%
2960     \else
2961         \let\Hy@escapestring\@firstofone
2962     \fi
2963 }%
2964 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
2965     \ifx\#3\%
2966     \else
2967         \expandafter
2968         \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
2969         \ltx@ReturnAfterFi{%
2970             \Hy__hypescapeform#3\@nil
2971         }%
2972     \fi
2973 }%
2974 \fi
2975 \def\PDFForm@Name{%
2976     \PDFForm__hypName\Fld@name
2977     \ifx\Fld@altname\relax
2978     \else
2979         \PDFForm__hypName\Fld@altname
2980     \fi
2981     \ifx\Fld@mappingname\relax
2982     \else
2983         \PDFForm__hypName\Fld@mappingname
2984     \fi
2985 }
2986 \def\PDFForm__hypName#1{%
2987     \begingroup
2988         \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
2989             \ifHy@unicode
2990                 \Hy@unicodedefalse
2991             \fi
2992             \fi
2993         \pdfstringdef\Hy@gtemp#1%
2994     \endgroup
2995     \let#1\Hy@gtemp
2996 }
2997 \def\Fld@X@additionalactions{%
2998     \ifx\Fld@keystroke@code\@empty
2999     \else
3000         /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3001     \fi
3002     \ifx\Fld@format@code\@empty
3003     \else
3004         /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3005     \fi

```

```

3006 \ifx\Fld@validate@code\@empty
3007 \else
3008   /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3009 \fi
3010 \ifx\Fld@calculate@code\@empty
3011 \else
3012   /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3013 \fi
3014 \ifx\Fld@onfocus@code\@empty
3015 \else
3016   /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3017 \fi
3018 \ifx\Fld@onblur@code\@empty
3019 \else
3020   /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3021 \fi
3022 \ifx\Fld@onmousedown@code\@empty
3023 \else
3024   /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3025 \fi
3026 \ifx\Fld@onmouseup@code\@empty
3027 \else
3028   /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3029 \fi
3030 \ifx\Fld@onenter@code\@empty
3031 \else
3032   /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3033 \fi
3034 \ifx\Fld@onexit@code\@empty
3035 \else
3036   /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3037 \fi
3038 }
3039 \ExplSyntaxOn
3040 \def\Fld@additionalactions
3041 {%
3042   \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{\}
3043   {
3044     \pdfmeta_standard_verify:nT {\annot_widget_no_AA}
3045     {\AA<<\Fld@X@additionalactions>>}
3046   }
3047 }
3048 \ExplSyntaxOff
3049 \def\Fld@annotnames{%
3050   /T(\Fld@name)%
3051   \ifx\Fld@altname\relax
3052   \else
3053     /TU(\Fld@altname)%
3054   \fi
3055   \ifx\Fld@mappingname\relax
3056   \else
3057     /TM(\Fld@mappingname)%
3058   \fi
3059 }

```



```

3060 \ExplSyntaxOn
3061 \def\PDFForm@Check
3062 {
3063   /Subtype/Widget
3064   ~\Fld@annotflags
3065   ~\Fld@pageobjref
3066   ~\Fld@annotnames
3067   /FT/Btn
3068   \Fld@flags
3069   /Q~\Fld@align
3070   /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3071   /AP
3072   <<
3073   /N
3074   <<
3075     /Yes~\pdfxform_ref:o{\l__hyp_CheckmarkYes_tl}
3076     /Off~\pdfxform_ref:o{\l__hyp_CheckmarkOff_tl}
3077   >>
3078   >>
3079   /MK<<
3080     \int_compare:nNf {\Fld@rotation}={0}
3081     {
3082       /R~\Fld@rotation
3083     }
3084     \tl_if_empty:NF\Fld@bordercolor
3085     {
3086       /BC[\Fld@bordercolor]
3087     }
3088     \tl_if_empty:NF\Fld@bcolor
3089     {
3090       /BG[\Fld@bcolor]
3091     }
3092     /CA(\Hy@escapestring{\Fld@cbsymbol})%
3093   >>
3094   /DA
3095   (
3096     /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3097     \tl_if_empty:NF \Fld@color
3098     {
3099       \c_space_tl \Fld@color
3100     }
3101   )
3102   /H/P
3103   \legacy_if:nTF {\Fld@checked}
3104   {
3105     /V/Yes /AS/Yes
3106   }
3107   {
3108     /V/Off /AS/Off
3109   }
3110   \Fld@additionalactions
3111 }
3112 \ExplSyntaxOff
3113 \ExplSyntaxOn

```

```

3114 \def\PDFForm@Push
3115 {
3116     /Subtype/Widget
3117     ~\Fld@annotflags
3118     ~\Fld@pageobjref
3119     ~\Fld@annotnames
3120     /FT/Btn
3121     ~\Fld@flags
3122     /H/P
3123     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3124     \bool_if:nT
3125     {
3126         !\int_compare_p:nNn {\Fld@rotation} = {0}
3127         ||
3128         \tl_if_exist_p:N \Fld@bordercolor
3129     }
3130     {
3131         /MK
3132         <<
3133             \int_compare:nNf {\Fld@rotation} = {0}
3134             {
3135                 /R~\Fld@rotation
3136             }
3137             \tl_if_exist:NT \Fld@bordercolor
3138             {
3139                 /BC[\Fld@bordercolor]
3140             }
3141         >>
3142     }
3143     /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3144     \Fld@additionalactions
3145 }
3146
3147 \ExplSyntaxOff
3148 \def\PDFForm@List{%
3149     /Subtype/Widget%
3150     \Fld@annotflags
3151     \Fld@pageobjref
3152     \Fld@annotnames
3153     /FT/Ch%
3154     \Fld@flags
3155     /Q \Fld@align
3156     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3157     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3158         \ifx\Fld@bordercolor\relax\else 1\fi
3159         \ifx\Fld@bcolor\relax \else 1\fi
3160         \space
3161     \else
3162         /MK<<%
3163         \ifnum\Fld@rotation=\z@
3164         \else
3165             /R \Fld@rotation
3166         \fi
3167         \ifx\Fld@bordercolor\relax

```

```

3168         \else
3169             /BC[\Fld@bordercolor]%
3170         \fi
3171         \ifx\Fld@bcolor\relax
3172         \else
3173             /BG[\Fld@bcolor]%
3174         \fi
3175     >>%
3176 \fi
3177 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3178     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3179 \Fld@choices
3180 \Fld@additionalactions
3181 }
3182 \ExplSyntaxOn
3183 \def\PDFForm@Radio
3184 {
3185     /Subtype/Widget
3186     ~\Fld@annotflags
3187     ~\Fld@pageobjref
3188     ~\Fld@annotnames
3189     /FT/Btn
3190     \Fld@flags
3191     /H/P
3192     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3193     /MK<<
3194     \ifnum\Fld@rotation=\z@
3195     \else
3196         /R~\Fld@rotation
3197     \fi
3198     \ifx\Fld@bordercolor\relax
3199     \else
3200         /BC[\Fld@bordercolor]%
3201     \fi
3202     \ifx\Fld@bcolor\relax
3203     \else
3204         /BG[\Fld@bcolor]%
3205     \fi
3206     /CA(\Hy@escapestring{\Fld@radiosymbol})%
3207     >>
3208     /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3209         \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3210 \ifx\Fld@default\@empty
3211     /V/Off%
3212     /DV/Off%
3213 \else
3214     /V/\Fld@default
3215     /DV/\Fld@default
3216 \fi
3217 \Fld@additionalactions
3218 }
3219 \ExplSyntaxOff
3220 \ExplSyntaxOn
3221 % Does an appeareance dict make sense here?

```

```

3222 \def\PDFForm@Text
3223 {
3224     /Subtype/Widget
3225     ~\Fld@annotflags
3226     ~\Fld@pageobjref
3227     ~\Fld@annotnames
3228     /FT/Tx
3229     ~\Fld@flags
3230     /Q~\Fld@align
3231     /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3232     \bool_if:nT
3233     {
3234         !\int_compare_p:nNn {\Fld@rotation} = {0}
3235         ||
3236         \tl_if_exist_p:N \Fld@bordercolor
3237         ||
3238         \tl_if_exist_p:N \Fld@bcolor
3239     }
3240     {
3241         /MK
3242         <<
3243             \int_compare:nNf {\Fld@rotation} = {0}
3244             {
3245                 /R~\Fld@rotation
3246             }
3247             \tl_if_exist:NT \Fld@bordercolor
3248             {
3249                 /BC[\Fld@bordercolor]
3250             }
3251             \tl_if_exist:NT \Fld@bcolor
3252             {
3253                 /BG[\Fld@bcolor]
3254             }
3255             >>
3256     }
3257     /DA
3258     (
3259         /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3260         \tl_if_empty:NF {\c_space_tl\Fld@color}
3261     )
3262     /DV(\Hy@escapestring{\Fld@default})
3263     /V(\Hy@escapestring{\Fld@value})
3264     ~\Fld@additionalactions
3265     \int_compare:nNt { \Fld@maxlen}>{0}
3266     {
3267         /MaxLen~\Fld@maxlen
3268     }
3269 }
3270 \ExplSyntaxOff
3271
3272 \def\PDFForm@Submit{%
3273     /Subtype/Widget%
3274     \Fld@annotflags
3275     \Fld@pageobjref

```

```

3276 \Fld@annotnames
3277 /FT/Btn%
3278 \Fld@flags
3279 /H/P%
3280 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3281 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3282 \ifx\Fld@bordercolor\relax\else 1\fi
3283 \space
3284 \else
3285 /MK<<%
3286 \ifnum\Fld@rotation=\z@
3287 \else
3288 /R \Fld@rotation
3289 \fi
3290 \ifx\Fld@bordercolor\relax
3291 \else
3292 /BC[\Fld@bordercolor]%
3293 \fi
3294 >>%
3295 \fi
3296 /A<<%
3297 /S/SubmitForm%
3298 /F<<%
3299 /FS/URL%
3300 /F(\Hy@escapestring{\Form@action})%
3301 >>%
3302 \Fld@submitflags
3303 >>%
3304 \Fld@additionalactions
3305 }
3306 \ExplSyntaxOn
3307 \def\PDFForm@Reset{%
3308 /Subtype/Widget%
3309 \Fld@annotflags
3310 \Fld@pageobjref
3311 \Fld@annotnames
3312 /FT/Btn%
3313 \Fld@flags
3314 /H/P%
3315 /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3316 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3317 \ifx\Fld@bordercolor\relax\else 1\fi
3318 \space
3319 \else
3320 /MK<<%
3321 \ifnum\Fld@rotation=\z@
3322 \else
3323 /R~\Fld@rotation
3324 \fi
3325 \ifx\Fld@bordercolor\relax
3326 \else
3327 /BC[\Fld@bordercolor]%
3328 \fi
3329 >>%

```

```

3330         \fi
3331         /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3332         /A<</S/ResetForm>>%
3333         \Fld@additionalactions
3334     }%
3335
3336
3337 %these patterns are used in hyperref checks.
3338 %it is unclear if they are really useful and if a backend support is
3339 %needed.
3340 \str_case:VnF \c_sys_backend_str
3341 {
3342     { pdfmode }
3343     {
3344         \def\HyPat@ObjRef
3345         {
3346             [0-9]*[1-9][0-9]*~0~R
3347         }
3348     }
3349     { dvipdfmx }
3350     {
3351         \def\HyPat@ObjRef
3352         {
3353             @[^~]+
3354         }
3355     }
3356     { xdvipdfmx }
3357     {
3358         \def\HyPat@ObjRef
3359         {
3360             @[^~]+
3361         }
3362     }
3363 }
3364 { %also set in hyperref sty, so probably not needed.
3365     \def\HyPat@ObjRef/{.+}
3366 }
3367
3368
3369 \ExplSyntaxOff
3370 % UF: removed Hy@writebookmark
3371 %     \Hy@currentbookmarklevel{0}
3372 %     \Hy@numberline
3373 %     \__hypwritetorep
3374 %     counter{bookmark@seq@number}
3375 % removed \HyPsd@SanitizeForOutFile, not needed
3376 % removed \currentpdfbookmark, defined by bookmark,
3377 % should use \newcommand there
3378 % removed \subpdfbookmark, defined by bookmark,
3379 % should use \newcommand there
3380 % removed \belowpdfbookmark, defined by bookmark,
3381 % should use \newcommand there
3382 % removed \pdfbookmark, defined by bookmark,
3383 % \BOOKMARK

```

```

3384 % \@BOOKMARK
3385 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3386 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3387 %% removed \ReadBookmarks / unneeded with bookmark.
3388 %% removed \Hy@OutlineName
3389 %% removed \check@bm@number
3390 %% removed \calc@bm@number
3391
3392 \ifHy@implicit
3393 \else
3394   \expandafter\endinput
3395 \fi
3396 \newlength\Hy@SectionHShift
3397 \def\Hy@SectionAnchorHref#1{%
3398   \ifx\protect\@typeset@protect
3399     \Hy__hypSectionAnchor{#1}%
3400   \fi
3401 }
3402 \DeclareRobustCommand*\Hy__hypSectionAnchor{[1]{%
3403   \leavevmode
3404   \hbox to 0pt{%
3405     \kern-\Hy@SectionHShift
3406     \Hy@raisedlink{%
3407       \hyper@anchorstart{#1}\hyper@anchorend
3408     }%
3409     \hss
3410   }%
3411 }
3412 \let\H@old@ssect\@ssect
3413 \def\@ssect#1#2#3#4#5{%
3414   \Hy@MakeCurrentHrefAuto{section*}%
3415   \setlength{\Hy@SectionHShift}{#1}%
3416   \begingroup
3417     \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3418     \toks\tw@\expandafter{%
3419       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3420       #5%
3421     }%
3422   \edef\x{\endgroup
3423     \the\toks@{\the\toks\tw@}%
3424   }\x
3425 }
3426 \let\H@old@schapter\@schapter
3427 \def\@schapter#1{%
3428   \begingroup
3429     \let\@mkboth\@gobbletwo
3430     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3431     \Hy@raisedlink{%
3432       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3433     }%
3434   \endgroup
3435   \H@old@schapter{#1}%
3436 }
3437 \ltx@ifundefined{chapter}{-}{%

```

```

3438 \let\Hy@org@chapter\@chapter
3439 \def\@chapter{%
3440   \def\Hy@next{%
3441     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3442     \Hy@raisedlink{%
3443       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3444     }%
3445   }%
3446   \ifnum\c@secnumdepth>\m@ne
3447     \ltx@ifundefined{if@mainmatter}%
3448     \iftrue{\csname if@mainmatter\endcsname}%
3449     \let\Hy@next\relax
3450   \fi
3451 \fi
3452 \Hy@next
3453 \Hy@org@chapter
3454 }%
3455 }
3456 \let\H@old@part\@part
3457 \begingroup\expandafter\expandafter\expandafter\endgroup
3458 \expandafter\ifx\csname chapter\endcsname\relax
3459   \let\Hy@secnum@part\z@
3460 \else
3461   \let\Hy@secnum@part\m@ne
3462 \fi
3463 \def\@part{%
3464   \ifnum\Hy@secnum@part>\c@secnumdepth
3465     \phantomsection
3466   \fi
3467   \H@old@part
3468 }
3469 \let\H@old@spart\@spart
3470 \def\@spart#1{%
3471   \Hy@MakeCurrentHrefAuto{part*}%
3472   \Hy@raisedlink{%
3473     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3474   }%
3475   \H@old@spart{#1}%
3476 }
3477 \let\H@old@sect\@sect
3478 \def\@sect#1#2#3#4#5#6[#7]#8{%
3479   \ifnum #2>\c@secnumdepth
3480     \expandafter\@firstoftwo
3481   \else
3482     \expandafter\@secondoftwo
3483   \fi
3484   {%
3485     \Hy@MakeCurrentHrefAuto{section*}%
3486     \setlength{\Hy@SectionHShift}{#3}%
3487     \begingroup
3488       \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3489       \toks\tw@\expandafter{%
3490         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3491         #8%

```



```

3492     }%
3493     \edef\x{\endgroup
3494       \the\toks@{\the\toks\tw@}%
3495     }\x
3496   }{%
3497     \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8}%
3498   }%
3499 }
3500 \expandafter\def\csname Parent-4\endcsname{}
3501 \expandafter\def\csname Parent-3\endcsname{}
3502 \expandafter\def\csname Parent-2\endcsname{}
3503 \expandafter\def\csname Parent-1\endcsname{}
3504 \expandafter\def\csname Parent0\endcsname{}
3505 \expandafter\def\csname Parent1\endcsname{}
3506 \expandafter\def\csname Parent2\endcsname{}
3507 \expandafter\def\csname Parent3\endcsname{}
3508 \expandafter\def\csname Parent4\endcsname{}
3509 %%
3510 %% End of file 'hgeneric-testphase.def'.
3511 \</package>
3512 \*colorscheme>
3513 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3514 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3515 % linkcolor=
3516 %,filecolor=
3517 %,urlcolor=
3518 %,menucolor=
3519 %,runcolor=
3520 %,linkbordercolor=
3521 %,filebordercolor=
3522 %,urlbordercolor=
3523 %,menubordercolor=
3524 %,runbordercolor=
3525
3526 \prop_const_from_keyval:Nn \c__hyp_colorscheme_original_prop
3527 {
3528   linkcolor      = [rgb]{1,0,0}, %red
3529   filecolor      = [rgb]{0,1,1}, %cyan
3530   urlcolor       = [rgb]{1,0,1}, %magenta
3531   menucolor      = [rgb]{1, 0, 0}, %red
3532   runcolor       = [rgb]{0,1,1}, %cyan
3533   %-----
3534   linkbordercolor = [rgb]{1, 0 ,0 },
3535   filebordercolor = [rgb]{0, .5, .5},
3536   urlbordercolor  = [rgb]{0, 1, 1},
3537   menubordercolor = [rgb]{1, 0, 0},
3538   runbordercolor  = [rgb]{0, .7, .7}
3539 }
3540
3541 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3542 {
3543   linkcolor      = [rgb]{0,0.2,0.6},
3544   filecolor      = [rgb]{0.8,0,0.8},
3545   urlcolor       = [rgb]{0.8,0,0.8},

```

```

3546     menucolor      = [rgb]{0,0.2,0.6},
3547     runcolor       = [rgb]{0.8,0,0.8},
3548     %-----      %-----
3549     linkbordercolor = [rgb]{0,0.2,0.6},
3550     filebordercolor = [rgb]{0.8,0,0.8},
3551     urlbordercolor  = [rgb]{0.8,0,0.8},
3552     menubordercolor = [rgb]{0,0.2,0.6},
3553     runbordercolor  = [rgb]{0.8,0,0.8}
3554 }
3555
3556 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3557 { %two colors: intern/extern
3558     linkcolor      = [rgb]{0.79216, 0, 0.12549},
3559     filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3560     urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3561     menucolor      = [rgb]{0.79216, 0, 0.12549 },
3562     runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3563     %-----      %-----
3564     linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3565     filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3566     urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3567     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3568     runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3569 }
3570
3571 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3572 { %all darkgray
3573     linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3574     filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3575     urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3576     menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3577     runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3578     %-----      %-----
3579     linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3580     filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3581     urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3582     menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3583     runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3584 }
3585
3586 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3587 { %dvipsnam.def
3588     linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3589     filecolor      = [rgb]{1, 0, 0}, %Red
3590     urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3591     menucolor      = [rgb]{1, 0, 0}, %Red
3592     runcolor       = [rgb]{1, 0, 0}, %Red
3593     %-----      %-----
3594     linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3595     filebordercolor = [rgb]{1, 0, 0}, %Red
3596     urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3597     menubordercolor = [rgb]{1, 0, 0}, %Red
3598     runbordercolor  = [rgb]{1, 0, 0} %Red
3599 }

```

```

3600
3601 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3602 { %dvipsnam.def
3603   linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3604   filecolor      = [rgb]{0, 1, 0},   %Green
3605   urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3606   menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3607   runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3608   %----- %-----
3609   linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3610   filebordercolor = [rgb]{0, 1, 0},   %Green
3611   urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3612   menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3613   runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3614 }
3615
3616
3617 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3618 {
3619   linkcolor      = [rgb]{0.50196, 0, 0.02353},
3620   filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3621   urlcolor       = [rgb]{0.54118, 0, 0.52941},
3622   menucolor      = [rgb]{0.44706, 0.45882, 0},
3623   runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3624   %----- %-----
3625   linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3626   filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3627   urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3628   menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3629   runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3630 }
3631
3632 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3633 {
3634   linkcolor      = [rgb]{0,0,0},
3635   filecolor      = [rgb]{0,0,0},
3636   urlcolor       = [rgb]{0,0,0},
3637   menucolor      = [rgb]{0,0,0},
3638   runcolor       = [rgb]{0,0,0},
3639   %----- %-----
3640   linkbordercolor = [rgb]{0,0,0},
3641   filebordercolor = [rgb]{0,0,0},
3642   urlbordercolor  = [rgb]{0,0,0},
3643   menubordercolor = [rgb]{0,0,0},
3644   runbordercolor  = [rgb]{0,0,0}
3645 }
3646 </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	
\#	638
\%	639
\.	456, 460, 462
@curropt commands:	
\@curropt:	2608, 2676
\\	18, 19, 20, 27, 31, 41, 61, 68, 75, 82, 89, 104, 109, 110, 118, 119, 128, 129, 130, 131, 132, 139, 768, 2610, 2678, 2965
_	454, 456, 460, 462
A	
\A	454
\Acrobatmenu	17, 156
\addcontentsline	12
\AddToDocumentProperties	330
\AddToHook	355
\advance	2541, 2615, 2659, 2660
allcolor (hypersetup key)	897
\author	2
B	
\b	458
\begingroup	184, 238, 274, 2384, 2575, 2604, 2616, 2650, 2987, 3416, 3428, 3457, 3487
\belowpdfbookmark	3380
\bgroup	248
\BOOKMARK	3383
bookmarkstype (hypersetup key)	12
bool commands:	
\bool_if:N \mathcal{TF}	241, 256, 304, 587, 602, 613, 626, 634, 664, 722, 779, 867, 877, 1117, 1128
\bool_if:n \mathcal{TF}	734, 1820, 1845, 3124, 3232
\bool_lazy_and:nn \mathcal{TF}	350
\bool_new:N	178, 179, 440, 444, 448
\bool_set_true:N	449
bordercolormodel (hypersetup key)	12, 918
box commands:	
\box_dp:N	530, 2747, 2786, 2821
\box_ht:N	529, 2746, 2785, 2820
\box_new:N	389, 451
\box_set_dp:Nn	2017
\box_set_ht:Nn	2024
\box_use:N	1134, 2749, 2795, 2824
\box_use_drop:N	1139
\box_wd:N	528, 2745, 2784, 2819
\l_tmpa_box	2743, 2745, 2746, 2747, 2749, 2782, 2784, 2785, 2786, 2795, 2817, 2819, 2820, 2821, 2824
C	
\char	2480, 2492, 2504, 2516
\chardef	137
\cite	30
clist commands:	
\clist_map_function:nN	111, 120
\clist_map_inline:nn	2181
\clist_map_variable:nNn	2607, 2675
color commands:	
\color_export:nnN	817
\color_export:nnnN	823
\color_select:n	834, 870, 1138
\color_select:nn	840
\color_set:nn	2, 335, 851
\color_set:nnn	2, 334, 857
color names:	
hyp/annot/file	437
hyp/annot/link	437
hyp/annot/menu	437
hyp/annot/run	437
hyp/annot/url	437
colorfile (hypersetup key)	897
colorlink (hypersetup key)	897
colorlinks (hypersetup key)	883
colormenu (hypersetup key)	897
colorrun (hypersetup key)	897
colorscheme (hypersetup key)	1, 1301
colorurl (hypersetup key)	897
cs commands:	
\cs_generate_variant:Nn	135, 136, 332, 368, 383, 388, 492, 826, 843, 860
\cs_gset:Npn	1109
\cs_if_exist_p:N	351
\cs_new:Npn	104, 384
\cs_new_protected:Npn	184, 265, 280, 293, 314, 327, 362, 371, 468, 472, 476, 480, 493, 495, 499, 541, 547, 554, 558, 580, 585, 600, 611, 624, 662, 720, 777, 810, 821, 827, 838, 844, 855, 1053, 2052, 2108
\cs_set_eq:NN	140, 149, 150, 190, 232, 638, 639, 1378

<code>\cs_set_protected:Npn</code>	138, 163, 509, 768	<code>\else</code>	2383, 2571, 2625, 2629, 2632, 2649, 2657, 2938, 2943, 2960, 2966, 2978, 2982, 2999, 3003, 3007, 3011, 3015, 3019, 3023, 3027, 3031, 3035, 3052, 3056, 3157, 3158, 3159, 3161, 3164, 3168, 3172, 3178, 3195, 3199, 3203, 3209, 3213, 3281, 3282, 3284, 3287, 3291, 3316, 3317, 3319, 3322, 3326, 3393, 3460, 3481
<code>\csname</code>	2622, 2957, 2968, 3448, 3458, 3500, 3501, 3502, 3503, 3504, 3505, 3506, 3507, 3508	<code>\endcsname</code>	2636, 2934, 2957, 2968, 3448, 3458, 3500, 3501, 3502, 3503, 3504, 3505, 3506, 3507, 3508
<code>\currentpdfbookmark</code>	3376	<code>\endgroup</code> ..	282, 2387, 2595, 2618, 2664, 2667, 2994, 3422, 3434, 3457, 3493
D			
<code>\d</code>	456, 460, 462	<code>\endinput</code>	3394
<code>debug</code> (hypersetup key)	<u>1325</u>	exp commands:	
<code>\DeclareDocumentMetadata</code>	3, 10, 13, 14, 17, 20, 28, 34	<code>\exp_args:Ne</code>	3042
<code>\DeclareRobustCommand</code>	237, 252, 272, 285, 299, 3402	<code>\exp_args:Nne</code>	2146
<code>\def</code>	144, 504, 2347, 2351, 2358, 2376, 2405, 2422, 2539, 2548, 2563, 2564, 2567, 2568, 2581, 2582, 2598, 2599, 2602, 2603, 2641, 2671, 2709, 2723, 2724, 2736, 2764, 2766, 2767, 2777, 2800, 2802, 2803, 2814, 2833, 2835, 2836, 2838, 2839, 2848, 2868, 2935, 2945, 2947, 2952, 2964, 2975, 2986, 2997, 3040, 3049, 3061, 3114, 3148, 3183, 3222, 3272, 3307, 3344, 3351, 3358, 3365, 3397, 3413, 3427, 3439, 3440, 3463, 3470, 3478, 3500, 3501, 3502, 3503, 3504, 3505, 3506, 3507, 3508	<code>\exp_args:Nno</code>	282, 295
<code>\DefaultHeightofCheckBox</code>	2839	<code>\exp_args:Nno</code>	267, 316, 2094
<code>\DefaultHeightofChoiceMenu</code>	2603	<code>\exp_args:Nnox</code>	672
<code>\DefaultHeightofReset</code>	2803	<code>\exp_args:NNV</code>	1805, 1836
<code>\DefaultHeightofSubmit</code>	2767	<code>\exp_args:NnV</code>	543, 550
<code>\DefaultHeightofText</code>	2572	<code>\exp_args:Nnx</code>	2329
<code>\DefaultHeightofTextMultiline</code> ...	2570	<code>\exp_args:No</code>	572, 576, 2726, 2769, 2805, 2840
<code>\DefaultOptionsofCheckBox</code>	2842	<code>\exp_args:NV</code>	526, 533
<code>\DefaultOptionsofPushButton</code>	2728	<code>\exp_args:Nx</code> ...	668, 2087, 2135, 2378
<code>\DefaultOptionsofReset</code>	2807	<code>\exp_not:N</code>	768
<code>\DefaultOptionsofSubmit</code>	2771	<code>\exp_not:n</code>	1240, 1274, 1385, 1445, 1466, 1503, 1523, 1546, 1590, 1623, 1648, 1699, 1724, 1751, 1800, 1877, 1905, 1932, 1952, 2167, 2274, 2302, 2312, 2323
<code>\DefaultOptionsofText</code>	2577	<code>\expandafter</code>	2576, 2610, 2619, 2621, 2622, 2678, 2951, 2967, 3394, 3418, 3419, 3457, 3458, 3480, 3482, 3489, 3490, 3500, 3501, 3502, 3503, 3504, 3505, 3506, 3507, 3508
<code>\DefaultWidthofCheckBox</code>	2838	<code>\ExplSyntaxOff</code>	10, 2415, 2531, 2558, 2864, 2932, 3048, 3112, 3147, 3219, 3270, 3369
<code>\DefaultWidthofChoiceMenu</code>	2602	<code>\ExplSyntaxOn</code>	8, 11, 2416, 2536, 2559, 2867, 3039, 3060, 3113, 3182, 3220, 3306
<code>\DefaultWidthofReset</code>	2802	<code>extension</code> (hypersetup key)	<u>12</u> , <u>1347</u>
<code>\DefaultWidthofSubmit</code>	2766	F	
<code>\DefaultWidthofText</code>	2567	<code>\fbox</code>	2522, 2527, 2700
<code>destlabel</code> (hypersetup key)	<u>12</u>	<code>\fi</code>	2372, 2399, 2402, 2411, 2573, 2581, 2582, 2613, 2631, 2634, 2635, 2641, 2644, 2653, 2655, 2662, 2665, 2940, 2962, 2972, 2974, 2980, 2984, 2991, 2992,
dim commands:			
<code>\dim_to_decimal_in_bp:n</code>	<u>11</u> , 140		
<code>\do</code>	2608, 2676		
<code>\documentclass</code>	<u>21</u>		
<code>draft</code> (hypersetup key)	<u>1325</u>		
E			
<code>\edef</code>	2386, 2618, 2654, 2661, 2950, 3422, 3493		
<code>\EdefEscapeName</code>	2674, 2679		

__hyp_colormodel_select_aux:wn .	\l__hyp_href_url_protocol_tl . . .
..... 827, 831, 838 180, 191, 230, 267, 318
__hyp_colormodel_set_aux:nwn . . .	__hyp_link_goto_begin:nw
..... 844, 848, 855 558, 592, 607
\c__hyp_colorscheme_daleif_prop 3541	__hyp_link_goto_end: . . 580, 593, 615
\c__hyp_colorscheme_henryford_-	\c__hyp_map_annot_hyp_prop 401
prop 3632	\c__hyp_map_hyp_annot_prop . . 401,
\c__hyp_colorscheme_julian_prop 3556	861, 925, 961, 988, 996, 1007, 1034,
\c__hyp_colorscheme_original_-	1042, 1111, 1215, 1251, 1261, 1393
prop 3526	__hyp_ocg_init: 1053, 1053, 1109, 1119
\c__hyp_colorscheme_phelype_prop	__hyp_PageLabels_gpsh: 499, 499, 512
..... 3617	\l__hyp_para_tmpa_tl 397, 733, 736, 745
\c__hyp_colorscheme_szabolcsA_-	\l__hyp_para_tmpa_tl_uuu\l__hyp_-
prop 3586	text_tmpa_str_uuu\g__hyp_text_-
\c__hyp_colorscheme_szabolcsB_-	tmpa_str 394
prop 3601	\l__hyp_RadioYes_tl 2669, 2670
\c__hyp_colorscheme_tivv_prop . . 3571	__hyp_ref_check:nn 361, 371, 383, 2546
\l__hyp_dest_box 29,	__hyp_ref_if_exist:nn . 361, 369, 370
451, 528, 529, 530, 2001, 2017, 2024	__hyp_ref_if_exist:nnTF . . 373, 2550
\l__hyp_dest_name_tmpa_tl . . 394,	__hyp_ref_label:nn 361, 362, 368, 2545
561, 562, 568, 572, 574, 577, 685, 698	__hyp_ref_value:nn 361, 384, 388, 2554
\l__hyp_dest_pdfremotestartview_-	__hyp_secondoftwoewithopt:wnn . . .
tl 420, 693, 1808, 1812 323, 324, 325, 326
\g__hyp_dest_pdfstartpage_tl . . .	__hyp_setup_info_date_key:nn . . .
... 420, 1819, 1821, 1828, 1846, 1853 2108, 2140, 2141
\g__hyp_dest_pdfstartview_tl . . .	__hyp_setup_info_key:nn . . 2052,
420, 1821, 1828, 1839, 1843, 1846, 1853	2101, 2102, 2103, 2104, 2106, 2107
\l__hyp_dest_pdfview_tl	__hyp_store_metadata:nn
..... 436, 544, 551, 327, 332, 2049,
1970, 1973, 1980, 1983, 1984, 1985,	2069, 2087, 2121, 2135, 2156, 2228
1986, 1987, 1988, 1993, 1997, 2033	__hyp_text_cleanup:N . 472, 472, 486
\c__hyp_dest_startview_regex . . .	\l__hyp_text_enc_dest_print_tl . .
..... 452, 1806, 1837 423, 684
\c__hyp_dest_undefined_tl	\l__hyp_text_enc_dest_tl
..... 400, 567, 568 29, 423, 522, 573
__hyp_destination:nn	\l__hyp_text_enc_file_print_tl . .
..... 29, 515, 515, 544, 551 423, 727
\l__hyp_filename_tmpa_tl . . . 394,	\l__hyp_text_enc_info_print_tl . .
667, 669, 675, 676, 681, 728, 729, 745 423, 497
__hyp_href_pdf_aux:nn 277, 280	\l__hyp_text_enc_para_print_tl . .
\l__hyp_href_pdf_destination_tl 423, 732
..... 181, 192, 231, 282	\l__hyp_text_enc_uri_print_tl . . .
\l__hyp_href_pdf_page_tl 182, 200, 692 243,
__hyp_href_run_aux:nn 290, 293	246, 258, 261, 306, 309, 423, 631, 1404
\l__hyp_href_run_parameter_tl . . .	__hyp_text_pdfstring:nnN
..... 183, 204, 295 480, 480, 492, 494,
__hyp_href_url_aux:n 311, 314	497, 520, 571, 629, 682, 725, 730, 1404
__hyp_href_url_aux:nn 263, 265	__hyp_text_pdfstring_info:nN . . .
\l__hyp_href_url_encode_bool 495, 495, 2063, 2081, 2091
..... 178, 189, 227, 241, 256, 304	__hyp_text_purify:nN . 468, 468, 485
__hyp_href_url_format	__hyp_text_string_from_unicode:nN
..... 184, 190, 232, 317 476, 476, 487
\l__hyp_href_url_ismap_bool	\g__hyp_text_tmpa_str . . 399, 488, 490
..... 179, 203, 634	

\l__hyp_text_tmpa_str	linkfileprefix	1347
..... 398, 485, 486, 487, 488	linktoc	1347
\l__hyp_tmpa_box 389, 1121, 1134, 1139	linktocpage	1347
\l__hyp_tmpa_int	localanchorname	1347
..... 389	menu	9, 1393
\l__hyp_tmpa_seq	menuborder	13
1808, 1837, 1839, 1960, 1961, 1965,	menuborderstyle	13, 1007
1967, 1991, 1998, 1999, 2000, 2009,	menucolor	897
2011, 2021, 2026, 2438, 2440, 2446	naturalnames	1347
\l__hyp_tmpa_str	nesting	13
..... 389, 2063, 2064, 2066,	ocgcolorfile	1147
2081, 2082, 2084, 2091, 2092, 2096	ocgcolorlink	1147
\l__hyp_tmpa_tl 389, 523, 527, 535,	ocgcolorlinks	1147
938, 942, 1404, 1405, 1410, 1804,	ocgcolormenu	1147
1806, 1835, 1837, 1967, 1968, 1975	ocgcolorrun	1147
\l__hyp_uri_tmpa_tl ... 394, 632, 633	ocgcolorurl	1147
hyp/annot/file (color name)	pageanchor	1347
..... 437	pdfauthor	2052
hyp/annot/link (color name)	pdfborder	13
..... 437	pdfborderstyle	13, 1007
hyp/annot/menu (color name)	pdfcreationdate	13, 2108
..... 437	pdfcreator	2052
hyp/annot/run (color name)	pdfencoding	1315
..... 437	pdfinfo	2170
hyp/annot/url (color name)	pdfkeywords	2052
..... 437	pdflang	13, 2038
hyp/text/pdfstring	pdflinkmargin	13
..... 467	pdfmoddate	13, 2108
\hypercalcbp	pdfproducer	2052
..... 11, 17, 140	pdfremotestartview	11
\HyperDestNameFilter	pdfstartview	11
..... 521, 572	pdfsubject	2052
\hypersetup 1, 2, 5, 8, 9, 12, 17, 64, 96, 163	pdftitle	2052
\hypersetup keys:	pdftrapped	2142
allcolor	pdfversion	1315
..... 897	pdfview	11, 1956
bookmarkstyle	plainpages	1347
..... 12	run	9, 1393
bordercolormodel	runborder	13
..... 12, 918	runborderstyle	13, 1007
colorfile	runcolor	897
..... 897	unicode	1315
colorlink	url	9, 1393
..... 897	urlborder	13
colorlinks	urlborderstyle	13, 1007
..... 883	urlcolor	897
colormenu	verbose	1325
..... 897	hypertextnames (hypersetup key)	1347
colorrun		
..... 897		
colorscheme		
..... 1, 1301		
colorurl		
..... 897		
debug		
..... 1325		
destlabel		
..... 12		
draft		
..... 1325		
extension		
..... 12, 1347		
file		
..... 9, 1393		
fileborderstyle		
..... 13, 1007		
filecolor		
..... 897		
final		
..... 1325		
hidefile		
..... 1277		
hidelink		
..... 1277		
hidelinks		
..... 1277		
hidemenu		
..... 1277		
hiderun		
..... 1277		
hideurl		
..... 1277		
hypertextnames		
..... 1347		
link		
..... 9, 1393		
linkborder		
..... 13		
linkborderstyle		
..... 13, 1007		
linkcolor		
..... 897		

I

\ifcase	3157, 3281, 3316
\ifcsname	2934
\ifdim	2613, 2652, 2653

<code>\ifnum</code> . . .	2988, 3157, 3163, 3194, 3281, 3286, 3316, 3321, 3446, 3464, 3479
<code>\iftrue</code>	3448
<code>\ifx</code>	2382, 2582, 2642, 2965, 2977, 2981, 2998, 3002, 3006, 3010, 3014, 3018, 3022, 3026, 3030, 3034, 3051, 3055, 3158, 3159, 3167, 3171, 3178, 3198, 3202, 3209, 3210, 3282, 3290, 3317, 3325, 3398, 3458
<code>\immediate</code>	2366, 2369
int commands:	
<code>\int_compare:nNnTF</code> . . .	1965, 1991, 2442, 2701, 3080, 3133, 3243, 3265
<code>\int_compare_p:nNn</code>	3126, 3234
<code>\int_eval:n</code>	691
<code>\int_max:nn</code>	692
<code>\int_new:N</code>	392
int internal commands:	
<code>__hypListbox</code>	2663, 2709
<code>__hypRadio</code>	2648, 2671
<code>__hypwritetorep</code>	3373
iow commands:	
<code>\iow_newline:</code>	2883, 2889, 2902
K	
<code>\kern</code>	3405
keys commands:	
<code>\keys_define:nn</code> 169, 187, 225, 336, 883, 899, 906, 918, 927, 948, 963, 982, 1009, 1028, 1149, 1161, 1175, 1196, 1209, 1217, 1246, 1277, 1290, 1301, 1315, 1325, 1334, 1347, 1374, 1395, 1400, 1956, 2038, 2054, 2072, 2110, 2124, 2142, 2170, 2226, 2231, 2244, 2264
<code>\l_keys_key_str</code>	174, 2096, 2329
<code>\keys_set:nn</code> 166, 195, 214, 239, 254, 275, 288, 302, 344, 1307, 1314, 2147, 2174, 2177, 2178, 2179, 2180, 2254, 2329
<code>\keys_set_known:nn</code>	2332
<code>\kvsetkeys</code>	165, 2425
L	
<code>\label</code>	8, 12
<code>\LayoutCheckField</code>	2850
<code>\LayoutChoiceField</code>	2645
<code>\LayoutPushButtonField</code>	2738, 2755
<code>\LayoutTextField</code>	2583
<code>\leavevmode</code>	2584, 2682, 3403
legacy commands:	
<code>\legacy_if:nTF</code>	151, 353, 357, 507, 2463, 2734, 2775, 2813, 2846, 3103
<code>\let</code>	147, 240, 255, 276, 289, 303, 324, 325, 326, 597, 647, 654, 708, 714, 758, 764, 807, 2342, 2343, 2345, 2346, 2373, 2424, 2532, 2533, 2534, 2565, 2566, 2600, 2601, 2643, 2930, 2937, 2939, 2944, 2961, 2995, 3412, 3426, 3429, 3438, 3449, 3456, 3459, 3461, 3469, 3477
<code>link</code> (hypersetup key)	9, 1393
<code>linkborder</code> (hypersetup key)	13
<code>linkborderstyle</code> (hypersetup key)	13, 1007
<code>linkcolor</code> (hypersetup key)	897
<code>linkfileprefix</code> (hypersetup key)	1347
<code>linktoc</code> (hypersetup key)	1347
<code>linktocpage</code> (hypersetup key)	1347
<code>localanchorname</code> (hypersetup key)	1347
M	
<code>\MakeButtonField</code> 2743, 2758, 2782, 2817, 2828
<code>\MakeChoiceField</code>	2718
<code>\MakeFieldObject</code>	2333, 2472, 2484, 2496, 2508, 2520, 2525
<code>\MakeRadioField</code>	2700
<code>\MakeTextField</code>	2592
<code>\mbox</code>	1131
<code>menu</code> (hypersetup key)	9, 1393
<code>menuborder</code> (hypersetup key)	13
<code>menuborderstyle</code> (hypersetup key)	13, 1007
<code>menucolor</code> (hypersetup key)	897
mode commands:	
<code>\mode_if_horizontal:TF</code>	517, 539
<code>\mode_leave_vertical:</code> 560, 641, 700, 747, 784, 2712, 2740, 2757, 2779, 2809, 2852
msg commands:	
<code>\msg_error:nn</code>	352, 2754, 2827
<code>\msg_line_context:</code>	69
<code>\g_msg_module_name_prop</code>	12
<code>\msg_new:nnn</code>	37, 44, 49, 53, 57, 64, 71, 78, 85, 92, 98, 105, 114, 124
<code>\msg_new:nnnn</code>	13, 23
<code>\msg_warning:nn</code>	153, 1321
<code>\msg_warning:nnn</code>	162, 564, 802
<code>\msg_warning:nnnn</code> 1153, 1200, 1443, 1487, 1521, 1544, 1606, 1635, 1669, 1684, 1697, 1710, 1735, 1768, 1784, 1811, 1842, 1875, 1889, 1916, 1950, 2032
<code>\msg_warning:nnnnn</code>	173, 1237, 1271, 1382, 1463, 1500, 1587, 1620, 1645, 1721, 1748, 1797, 1902, 1929, 2164, 2271, 2290, 2309, 2320

N	
naturalnames (hypersetup key)	1347
nesting (hypersetup key)	13
\newcommand	143 , 3377 , 3379 , 3381
\newcount	2537 , 2708
\NewDocumentCommand	2333
\NewExpandableDocumentCommand	323
\newlength	3396
\noexpand	2619 , 2620 , 2621 , 2622 , 2948
\nolinkurl	3 , 4
O	
ocgcolorfile (hypersetup key)	1147
ocgcolorlink (hypersetup key)	1147
ocgcolorlinks (hypersetup key)	1147
ocgcolormenu (hypersetup key)	1147
ocgcolorrun (hypersetup key)	1147
ocgcolorurl (hypersetup key)	1147
P	
page/Trans internal commands:	
l_hyp_page/Trans	465
pageanchor (hypersetup key)	1347
\paperwidth	3
\PassOptionsToPackage	359 , 1339 , 1344
pdf commands:	
\pdf_bdcobject:nn	1133 , 1136
\pdf_destination:nn	24 , 135 , 534
\pdf_destination:nnnn	527
\pdf_emc:	1135 , 1141
\pdf_link_user:nnn	2715
\pdf_name_from_unicode_e:n	220 , 671 , 786
\pdf_object_if_exist:nTF	669
\pdf_object_new:nn	1055 , 1056 , 1057 , 1058 , 2870 , 2871 , 2872
\pdf_object_ref:n	136 , 681 , 1061 , 1063 , 1085 , 1086 , 1089 , 1093 , 1098 , 1103 , 1108 , 2456 , 2458 , 2928
\pdf_object_ref_last:	702 , 2259
\pdf_object_unnamed_write:nn	701 , 2255
\pdf_object_write:nn	1059 , 1065 , 1075 , 1087 , 2873 , 2915 , 2922
\pdf_pageobject_ref:n	1828 , 1853 , 2552
\pdf_string_from_unicode:nnN	478
\pdf_version:	1491 , 1610 , 1639 , 1673 , 1688 , 1714 , 1739 , 1772 , 1788 , 1893 , 1920
\pdf_version_compare:NnTF	1147 , 1194 , 1481 , 1572 , 1594 , 1630 , 1663 , 1678 , 1704 , 1729 , 1755 , 1778 , 1883 , 1910
\pdf_version_compare_p:Nn	738
\pdf_version_major:	150 , 1156 , 1204
\pdf_version_minor:	149 , 1156 , 1204
pdfannot commands:	
\pdfannot_box:nnnn	2587 , 2685 , 2744 , 2783 , 2818 , 2855
\pdfannot_box_ref_last:	2380 , 2396
\pdfannot_dict_put:nnn	642 , 702 , 748 , 787 , 939 , 974 , 998 , 1020 , 1044 , 1222 , 1253
\pdfannot_dict_remove:nn	933 , 969 , 990 , 1015 , 1036 , 1230 , 1263
\pdfannot_link:nnn	643 , 703 , 749 , 788
\pdfannot_link_goto_begin:nw	577
\pdfannot_link_goto_end:	582
\pdfannot_link_margin:n	8 , 1525
\c_pdfannot_link_types_seq	446
pdfauthor (hypersetup key)	2052
\pdfbookmark	3382
pdfborder (hypersetup key)	13
pdfborderstyle (hypersetup key)	13 , 1007
pdfcreationdate (hypersetup key)	13 , 2108
pdfcreator (hypersetup key)	2052
\pdfdest	11
pdfdict commands:	
\pdfdict_new:n	465 , 620 , 657 , 716 , 773
\pdfdict_put:nnn	212 , 219 , 466 , 621 , 622 , 633 , 636 , 658 , 659 , 671 , 678 , 688 , 698 , 717 , 718 , 729 , 744 , 774 , 775 , 785 , 1551 , 1552 , 1556 , 1557 , 2268 , 2278 , 2282 , 2285 , 2287 , 2306 , 2315 , 2317
\pdfdict_remove:nn	209 , 741 , 1561 , 1562
\pdfdict_use:n	642 , 701 , 748 , 754 , 787 , 792 , 2257
pdfencoding (hypersetup key)	1315
\pdfescapestring	2937
pdffile commands:	
\pdffile_embed_file:nnn	673
PDFForm internal commands:	
\PDFForm_hypName	2976 , 2979 , 2983 , 2986
pdfinfo (hypersetup key)	2170
pdfkeywords (hypersetup key)	2052
pdflang (hypersetup key)	13 , 2038
pdflinkmargin (hypersetup key)	13
pdfmanagement commands:	
\pdfmanagement_add:nn	1534
\pdfmanagement_add:nnn	501 , 1085 , 1086 , 1108 , 1410 , 1435 , 1451 , 1455 , 1475 , 1483 , 1512 , 1567 , 1577 , 1601 , 1615 , 1627 , 1632 , 1657 , 1665 , 1680 , 1706 , 1731 , 1763 , 1780 , 1826 , 1851 , 1865 , 1885 , 1912 , 1941 , 2047 , 2066 , 2084 ,

2095, 2119, 2133, 2151, 2240, 2259, 2431, 2448, 2455, 2457, 2459, 2465	\providecommand 141, 142, 2367, 2370, 2597, 2933
\pdfmanagement_if_active_p: . . . 351	\ProvidesFile 3
\pdfmanagement_remove:nn	
. . . 1407, 1431, 1439, 1459, 1471, 1496, 1508, 1516, 1530, 1539, 1583, 1598, 1617, 1654, 1693, 1718, 1744, 1759, 1793, 1823, 1848, 1861, 1870, 1898, 1925, 1937, 1946, 2044, 2060, 2078, 2116, 2130, 2237, 2250, 2469	
\pdfmanagement_show:n 2432	
pdfmeta commands:	
\pdfmeta_standard_verify:nnTF . . .	
. 782, 2731, 2810	
\pdfmeta_standard_verify:nTF . . .	
. 2461, 3044	
pdfmoddate (hypersetup key) 13, 2108	
pdfproducer (hypersetup key) 2052	
pdfremotestartview (hypersetup key) . . 11	
pdfstartview (hypersetup key) 11	
\pdfstringdef 8, 9, 17, 26, 2993	
\pdfstringdefDisableCommands	
. 324, 325, 326	
pdfsubject (hypersetup key) 2052	
pdftitle (hypersetup key) 2052	
pdftrapped (hypersetup key) 2142	
pdfversion (hypersetup key) 1315	
pdfview (hypersetup key) 11, 1956	
pdfxform commands:	
\pdfxform_new:nnn 2335	
\pdfxform_ref:n	
. . . 2695, 2696, 2790, 2791, 3075, 3076	
\phantom 2492, 2516	
\phantomsection 3465	
plainpages (hypersetup key) 1347	
prg commands:	
\prg_generate_conditional_-	
variant:Nnn 370	
\prg_new_eq_conditional:NNn . . . 369	
prop commands:	
\prop_const_from_keyval:Nn	
. 403, 411, 3526, 3541, 3556, 3571, 3586, 3601, 3617, 3632	
\prop_gput:Nnn 12, 2353, 2360	
\prop_if_empty:NTF 2434	
\prop_item:Nn 2451	
\prop_map_inline:Nn	
. 861, 925, 961, 988, 996, 1007, 1034, 1042, 1111, 1215, 1251, 1261, 1305, 1393, 2429, 2436	
\prop_new:N 2339, 2340	
\protect 240, 255, 276, 289, 303, 375, 597, 647, 654, 708, 714, 758, 764, 807, 3398	
	\ReadBookmarks 3387
	ref commands:
	\ref_if_exist:nn 369
	\ref_label:nn 365
	\ref_value:nn 22, 386
	regex commands:
	\regex_const:Nn 452
	\regex_extract_once:NnNTF 1806, 1837
	\relax 597, 647, 654, 708, 714, 758, 764, 807, 2600, 2601, 2642, 2930, 2977, 2981, 3051, 3055, 3158, 3159, 3167, 3171, 3198, 3202, 3282, 3290, 3317, 3325, 3449, 3458
	\RemoveFromHook 1415
	\RenewDocumentCommand 156
	\RequirePackage 7, 19, 355, 3385
	run (hypersetup key) 9, 1393
	runborder (hypersetup key) 13
	runborderstyle (hypersetup key) . 13, 1007
	runcolor (hypersetup key) 897
	S
	\selectfont 2479, 2491, 2503, 2515
	seq commands:
	\seq_const_from_clist:Nn 401
	\seq_count:N 1965, 1991
	\seq_get_right:NN 1967
	\seq_item:Nn 1808, 1839, 1961, 2009, 2011, 2021, 2026
	\seq_map_inline:Nn
	. 438, 442, 446, 897, 1192, 1288, 2446
	\seq_new:N 391
	\seq_put_right:Nn
 1998, 1999, 2000, 2438
	\seq_set_split:Nnn 1960
	\seq_sort:Nn 2440
	\setlength 3415, 3486
	\setpdflinkmargin 8
	\settowidth 2612
	\show 2673
	sort commands:
	\sort_return_same: 2444
	\sort_return_swapped: 2443
	\space 379, 2681, 3160, 3177, 3178, 3208, 3209, 3283, 3315, 3318
	\spacefactor 517, 539
	\stockwidth 3
	str commands:
	\c_colon_str 768

<code>\c_hash_str</code>	638	<code>\ifundefined</code>	2424, 2953
<code>\c_percent_str</code>	639	<code>\@latex@warning</code>	376
<code>\str_case:nnTF</code>	3340	<code>\@mainaux</code>	2366, 2369, 2389, 2408
<code>\str_case_e:nnTF</code>	1961	<code>\@mkboth</code>	3429
<code>\str_gset:Nn</code>	922	<code>\@nil</code>	2951, 2964, 2970
<code>\str_gset_eq:NN</code>	488	<code>\@part</code>	3456, 3463
<code>\str_head:n</code>	2153, 2158	<code>\@pdfauthor</code>	21
<code>\str_if_eq:nnTF</code>	524, 2064, 2082, 2092, 3042	<code>\@pdfborder</code>	141
<code>\str_if_eq_p:nn</code>	736	<code>\@pdfborderstyle</code>	142
<code>\str_lowercase:n</code>	1961, 2087, 2135, 2154, 2159	<code>\@savsf</code>	517, 539
<code>\str_new:N</code>	393, 398, 399, 437	<code>\@schapter</code>	3426, 3427
<code>\str_set:Nn</code>	470	<code>\@secondoftwo</code>	3482
<code>\str_set_convert:Nnnn</code>	8	<code>\@sect</code>	3477, 3478
<code>\str_set_eq:NN</code>	490	<code>\@spart</code>	3469, 3470
<code>\str_tail:n</code>	2154, 2159	<code>\@ssect</code>	3412, 3413
<code>\str_uppercase:n</code>	2147, 2153, 2158	<code>\@tempdima</code>	2606, 2613, 2615, 2652, 2653, 2654, 2658, 2659, 2660, 2661
<code>\string</code>	2367, 2370, 2391, 2409, 2953, 2957, 2968	<code>\@tempdimb</code>	2612, 2613
<code>\subpdfbookmark</code>	3378	<code>\@typeset@protect</code>	3398
sys commands:		<code>\@unexpandable@protect</code>	240, 255, 276, 289, 303
<code>\c_sys_backend_str</code>	3340	<code>\c@secnumdepth</code>	3446, 3464, 3479
T			
TeX and L ^A T _E X 2 _ε commands:			
<code>\@BOOKMARK</code>	3384	<code>\calc@bm@number</code>	3390
<code>\@CheckBox</code>	2833	<code>\check@bm@number</code>	3389
<code>\@ChoiceMenu</code>	2598	<code>\Fld@additionalactions</code>	3040, 3110, 3144, 3180, 3217, 3264, 3304, 3333
<code>\@Form</code>	2422	<code>\Fld@align</code>	3069, 3155, 3230
<code>\@PushButton</code>	2723	<code>\Fld@altname</code> ..	2977, 2979, 3051, 3053
<code>\@Reset</code>	2800	<code>\Fld@annotflags</code>	3064, 3117, 3150, 3186, 3225, 3274, 3309
<code>\@Submit</code>	2764	<code>\Fld@annotnames</code>	3049, 3066, 3119, 3152, 3188, 3227, 3276, 3311
<code>\@TextField</code>	2563	<code>\Fld@bcolor</code>	3088, 3090, 3171, 3173, 3202, 3204, 3238, 3251, 3253
<code>\@bookmarksopenlevel</code>	1426	<code>\fld@bcolor</code>	3159
<code>\@bsphack</code>	364	<code>\Fld@bordercolor</code>	3084, 3086, 3128, 3137, 3139, 3158, 3167, 3169, 3198, 3200, 3236, 3247, 3249, 3282, 3290, 3292, 3317, 3325, 3327
<code>\@chapter</code>	3438, 3439	<code>\Fld@borderstyle</code>	3070, 3123, 3156, 3192, 3231, 3280, 3331
<code>\@currDisplay</code>	2612, 2681	<code>\Fld@borderwidth</code> ..	2659, 2660, 3070, 3123, 3156, 3192, 3231, 3280, 3331
<code>\@currValue</code>	2679, 2695	<code>\Fld@calculate@code</code> ..	2382, 3010, 3012
<code>\@currentHref</code>	3419, 3432, 3443, 3473, 3490	<code>\Fld@calculate@sortkey</code>	2393
<code>\@curropt</code> ..	2597, 2607, 2610, 2675, 2678	<code>\Fld@cbsymbol</code>	3092
<code>\@empty</code>	2566, 2582, 2998, 3002, 3006, 3010, 3014, 3018, 3022, 3026, 3030, 3034, 3178, 3209, 3210	<code>\Fld@charsize</code>	2658, 3096, 3177, 3208, 3259, 3315
<code>\@endForm</code>	2532	<code>\Fld@checkequals</code>	2610, 2678
<code>\@esphack</code>	366	<code>\Fld@choices</code>	3179
<code>\@firstofone</code> ..	2373, 2939, 2944, 2961	<code>\Fld@color</code> ..	3097, 3099, 3178, 3209, 3260
<code>\@firstoftwo</code>	3480	<code>\Fld@default</code>	2565, 2582, 2600, 2643, 2673, 2674, 2836, 3210, 3214, 3215, 3262
<code>\@for</code>	2608, 2676		
<code>\@gobble</code>	2424		
<code>\@gobbletwo</code>	3429		
<code>\@hyper@launch</code>	34, 768		
<code>\@ifnextchar</code>	248		

\Fld@flags	3068,	\Hy@bookmarkstype	1427
3121, 3154, 3190, 3229, 3278, 3313		\Hy@chapapp	3430, 3441
\Fld@format@code	3002, 3004	\Hy@colorlink	31
\Fld@height		\Hy@currentbookmarklevel	3371
2568, 2589, 2592, 2603, 2661, 2687,		\Hy@DisableOption	148, 155
2700, 2718, 2767, 2803, 2839, 2857		\Hy@drafttrue	1338
\Fld@keystroke@code	2998, 3000	\Hy@escapeform	
\Fld@listcount	2672, 2680, 2701, 2708		2586, 2684, 2714, 2742,
\Fld@mappingname	2981, 2983, 3055, 3057	2781, 2816, 2854, 2935, 2942, 2945	
\Fld@maxlen	3265, 3267	\Hy@escapestring	2937,
\Fld@menulength	2605, 2611, 2658	2939, 2944, 2947, 2948, 2951, 2952,	
\Fld@name		2961, 2964, 3000, 3004, 3008, 3012,	
2564, 2599, 2724, 2835, 2976, 3050		3016, 3020, 3024, 3028, 3032, 3036,	
\Fld@onblur@code	3018, 3020	3092, 3143, 3206, 3262, 3263, 3300	
\Fld@onclick@code	3143	\Hy@finaltrue	1343
\Fld@onenter@code	3030, 3032	\Hy@FormObjects	2428, 2868, 2930
\Fld@onexit@code	3034, 3036	\Hy@gttemp	2993, 2995
\Fld@onfocus@code	3014, 3016	\Hy@href	248
\Fld@onmousedown@code	3022, 3024	\Hy@href@nextactionraw	213
\Fld@onmouseup@code	3026, 3028	\Hy@href@page	201
\Fld@pageobjref		\Hy@hypertextnamesfalse	1353
2534, 2548, 2933, 3065,		\Hy@hypertextnametrue	1352
3118, 3151, 3187, 3226, 3275, 3310		\Hy@linkfileprefix	1355
\Fld@radiosymbol	3206	\Hy@linktoc	1378
\Fld@rotation	3080,	\Hy@localanchornamefalse	1358
3082, 3126, 3133, 3135, 3157, 3163,		\Hy@localanchorname>true	1357
3165, 3194, 3196, 3234, 3243, 3245,		\Hy@MakeCurrentHrefAuto	
3281, 3286, 3288, 3316, 3321, 3323			3414, 3430, 3441, 3471, 3485
\Fld@submitflags	3302	\Hy@naturalnamesfalse	1362
\Fld@validate@code	3006, 3008	\Hy@naturalnametrue	1361
\Fld@value		\Hy@next	3440, 3449, 3452
2566, 2582, 2601, 2642, 2643, 3263		\Hy@numberline	144, 3372
\Fld@width	2567,	\Hy@org@chapter	3438, 3453
2581, 2588, 2592, 2602, 2641, 2652,		\Hy@OutlineName	3388
2654, 2686, 2700, 2718, 2736, 2766,		\Hy@OutlineRerunCheck	3386
2777, 2802, 2814, 2838, 2848, 2856		\Hy@pageanchorfalse	1366
\Fld@X@additionalactions		\Hy@pageanchortrue	1365
2997, 3042, 3045		\Hy@pdfmajorversion	150
\Form@action	3300	\Hy@pdfminorversion	149
\G@refundefinedtrue	375	\Hy@pdfstringtrue	27, 483
\H@old@part	3456, 3467	\Hy@pdfversion	2988
\H@old@chapter	3426, 3435	\Hy@plainpagesfalse	1370
\H@old@ssect	3477, 3488, 3497	\Hy@plainpagetrue	1369
\H@old@spart	3469, 3475	\Hy@pstringdef	493, 2968
\H@old@sssect	3412, 3417	\Hy@PutCatalog	499
\href@	248	\Hy@raisedlink	3406, 3431, 3442, 3472
\Hy@abspage	511	\Hy@RestoreLastskip	538
\Hy@activeanchorfalse	556	\Hy@safe@activetrue	640, 2385
\Hy@activeanchortrue	549	\Hy@SaveLastskip	518
\Hy@AtBeginDocument	2347, 2363	\Hy@secnum@part	3459, 3461, 3464
\Hy@bookmarksnumberedfalse	1419	\Hy@SectionAnchorHref	3397, 3419, 3490
\Hy@bookmarksnumberedtrue	1420	\Hy@SectionHShift	
\Hy@bookmarksopenfalse	1423		3396, 3405, 3415, 3486
\Hy@bookmarksopentrue	1424	\Hy@StepCount	2611, 2680

\Hy@temp	2386, 2401, 2950, 2951	\HyPsd@pdfencoding@unicode	147
\Hy@unicodedefalse	2990	\HyPsd@SanitizeForOutFile	3375
\Hy@unicodetrue	146	\if@filesw	2365, 2388, 2407
\Hy@VerboseAnchor	519	\ifFld@combo	2626, 2656
\Hy@verbosefalse	1329	\ifFld@hidden	2581, 2641
\Hy@VerboseLinkStart	589, 604	\ifFld@multiline	2569
\Hy@VerboseLinkStop	595, 617, 650, 710, 760, 798	\ifFld@popdown	2627
\Hy@verbosetrue	1328	\ifFld@radio	2623, 2646
\Hy@VersionChecked	137	\ifHy@implicit	3392
\Hy@WrapperDef	515	\ifHy@pdfescapeform	2936, 2946
\Hy@xspace@end	592, 649, 709, 759, 797	\ifHy@unicode	2989
\HyAnn@AbsPageLabel	2533, 2539, 2585,	\kv@set@family@handler	161
2683, 2713, 2741, 2780, 2815, 2853		\ltx@empty	2342,
\HyAnn@Count	2537, 2538, 2541, 2542,	2343, 2382, 2532, 2533, 2534, 2565	
2543, 2544, 2545, 2546, 2550, 2554		\ltx@gobble	2345
\HyField@AddToFields	2376, 2593,	\ltx@gobbletwo	2346
2702, 2719, 2750, 2794, 2823, 2860		\ltx@ifUndefined	3437, 3447
\HyField@afields	2342	\ltx@one	2541
\HyField@AfterAuxOpen	2347, 2373, 2406	\ltx@ReturnAfterFi	2969
\HyField@AuxAddToCoFields	2346, 2358, 2370, 2391	\ltx@zero	2538
\HyField@AuxAddToFields	2345, 2351, 2367, 2409	\m@ne	3446, 3461
\HyField@cofields	2343	\OBJ@OCG@view	143
\HyField@FlagsCheckBox	2845	\p@	2615
\HyField@FlagsChoice	2651	\pdf@ifdraftmode	2426
\HyField@FlagsPushButton	2733, 2773, 2812	\PDF@SetupDoc	138
\HyField@FlagsRadioButton	2647	\pdf@strcmp	2442
\HyField@FlagsSubmit	2774	\PDFForm@Check	2854, 2859, 3061
\HyField@FlagsText	2580	\PDFForm@List	2714, 2717, 3148
\HyField@PDFChoices	2711	\PDFForm@Name	2579, 2640, 2730, 2844, 2975
\HyField@SetKeys	2576, 2617,	\PDFForm@Push	2742, 2748, 3114
2620, 2639, 2726, 2769, 2805, 2840		\PDFForm@Radio	2684, 2690, 3183
\HyPat@ObjRef	3344, 3351, 3358, 3365	\PDFForm@Reset	2816, 2822, 3307
\hyper@anchor	541	\PDFForm@Submit	2781, 2788, 3272
\hyper@anchorend	541, 3407, 3432, 3443, 3473	\PDFForm@Text	2586, 2591, 3222
\hyper@anchorstart	541, 3407, 3432, 3443, 3473	\protected@edef	561
\hyper@link	31, 31, 585	\strip@pt	3096, 3177, 3208, 3259, 3315
\hyper@linkend	30, 31, 611	\toks@	3417, 3423, 3488, 3494
\hyper@linkfile	282, 662	\tw@	3418, 3423, 3489, 3494
\hyper@linklaunch	34, 295, 720, 771	\XR@ext	1349
\hyper@linknamed	35, 158, 777	\z@	2606, 3157, 3163,
\hyper@linkstart	30, 31, 600	3194, 3281, 3286, 3316, 3321, 3459	
\hyper@linkurl	267, 317, 624	\zref@labelbylist	2543
\hyper@normalise	248, 263, 277, 290, 311	\zref@labelbyprops	2542
\HyPL@Labels	501, 511	\zref@refused	2544
\HyPL@StorePageLabel	499	\texorpdfstring	27
\HyPsd@LoadUnicode	145	text commands:	
\HyPsd@pdfencoding	147	\text_expand:n	667
		\text_purify:n	8, 470
		\textcolor	2424, 2522, 2527
		\textsf	2522, 2527
		\the	511,
		2542, 2543, 2544, 2545, 2546, 2550,	
		2554, 2654, 2658, 2661, 3423, 3494	

