

The hyperref-generic module

A generic driver for hyperref

The L^AT_EX Project*

Version 0.95e, released 2021-06-14

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

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

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

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

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

1 Avoiding transition problems

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

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

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

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

2 Bookmarks / outlines

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

3 “Metadata”

“Metadata”, informations about the document, are stored in a PDF in two places: The `/Info` dictionary and the XMP-metadata. `hyperref` only handles the `/Info` dictionary. The XMP-metadata can be added with like `pdfx` and `hyperxmp`.

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

The current handling of the metadata is problematic:

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

To resolve some of this problem the driver will

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

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

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

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

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

- `pdflang` is deprecated. Instead `\DeclareDocumentMetadata` should be used:

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

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

4 Dates

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

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

5 PDF page size (mediabox)

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

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

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

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

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

6 Commands to create “external” references

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

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

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

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

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

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

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

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

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

6.1 Special problem: links to files

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

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

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

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

6.2 Splits

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

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

For this reason three new commands have been create:

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

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

6.3 Options

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

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

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

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

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

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

Some main changes are

- The default colors have been changed.
- `citecolor` is no longer set with `allcolors` and it is not part of the color scheme, it is only supported for compability.
- a number of color schemes have been predefined.

7.1 Background information

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

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

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

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

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

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

7.2 New Keys

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

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

- The boolean keys `url`, `link`, `run`, `menu`, `file` allow to deactivate locally the link types.

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

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

7.3 Public interfaces

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

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

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

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

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

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

7.4 Changed behaviour

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

Color keys accept the following input syntax:

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

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

frenchlinks The option `frenchlinks` does nothing at all.

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

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

8 PDF strings

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

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

The new method is under heavy development!

Important differences here are

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

9 Package options from `hyperref`

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

Options that still must be set as package options are

- `backref`
- `CJKbookmarks` this key should not be used anymore. At some time it will be removed.
- `destlabels` (destination names are taken from `\label` if possible)
- `encap`

- **hyperfigures** (according to the **hyperref** manual it makes figures hyper links, but actually is a no-op for most drivers, and it does nothing with this driver either.)
- **hyperfootnotes**
- **hyperindex**
- **implicit** (redefine L^AT_EX internals)
- **nesting** unneeded key, see comment below in 14. At some time it will be either removed or extended (if some use can be found).
- **pagebackref**
- **pdfpagelabels** (set PDF page labels)
- **psdextra** this loads some extra definitions used by `\pdfstringdef`. The new driver uses `\pdfstringdef` only for the bookmarks, for other strings it is not relevant.

Options that can be without problems set as package options are

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

Ignored options:

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

10 Disabling links

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

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

url (*setup key*)

file (*setup key*)

run (*setup key*)

menu (*setup key*)

11 Draftmode

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

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

12 Dropped options

A number of options are ignored by this driver

pdfversion The `pdfversion` should be set in `\DeclareDocumentMetadata`

setpagesize The key is ignored and the PDF page size is not set. Load `color` or `graphicx` or use a class which sets the PDF page size.

breaklinks The option does nothing sensible anyway (apart from triggering a warning). Currently with latex+dvips links can't be broken. But there is work in progress to change this.

unicode This is always true.

pdfa If this option is set to true `hyperref` normally checks and sets a small number of requirements for the PDF standard PDF/A. The key is ignored with this driver. Instead the wanted standard should be declared in `\DeclareDocumentMetadata`:

```
\DeclareDocumentMetadata{pdfstandard=A-2b}
```

Currently A-1b, A-2b, A-3b can be set. The support for various requirements is still incomplete, but the parts that `hyperref` checked are implemented:

- The `/F` key is added to links and `Print` is activated, `Hidden`, `Invisible`, `NoView` are deactivated.
- `/NeedAppearances` is suppressed
- `PushButtons`, which use the action `/S/JavaScript` are suppressed.
- `ResetButtons`, which use the action `/S/ResetForm` are suppressed.
- In widget annotations, the `/AA` dictionary is suppressed.

13 Destinations

Destinations (sometimes call anchors in the `hyperref` documentation) are the places a link jumped too. Unlike the name may suggest they don't described an exact location in the PDF. Instead a destination contains a reference to a page along with an instruction how to display this page. The normally used "`XYZ top left zoom`" for example instructs the viewer to show the page with the given *zoom* and the top left corner at the *top left* coordinates—which then gives the impression that there is an anchor at this position.

From these instructions two (`Fit` and `FitB`) don't take an argument. All others take one (`FitH`, `FitV`, `FitBH`, `FitBV`) or more (`XYZ`, `FitR`) arguments. These arguments

are normally coordinates, `XYZ` takes also a zoom factor. The coordinates are absolute coordinates in `bp` relative to the lower left corner of the PDF.

With the primitive command `\pdfdest` of `pdftex` almost all instructions are created with a keyword only: The needed coordinate is calculated automatically from the location the `\pdfdest` command is issued. So to get a specific coordinate one has to move the command to the right place. E.g.

```
\AddToHookNext{shipout/background}
{\put(0,-\pdfpageheight+100bp){\pdfdest name{destA} FitH\relax}}
```

Exceptions are the `XYZ` instruction, where `pdftex` accepts a keyword `zoom` followed by a zoom factor, and the `FitR` instruction which understands the keywords `width`, `height` and `depth` followed by a dimension, which is then used to calculate a rectangle relative to the current location. If no keywords are given the dimensions are taken from the surrounding box—which can also lead to zero sized areas.

The manual of `hyperref` gives a bit the impression as if this coordinates can be set manually by the user but as described above this is mostly wrong: It is for normal destination only possible with a dvi-backend like `dvips` which make use of `pdfmark.def`. `pdftex` and `luatex` can use manual coordinates only for `pdfstartview` and `pdfremotestartview`. As `dvips` was the first driver of `hyperref` the option `pdfview` was at first developed for it and then adapted to `pdftex`. But this had the effect that the handling of the option `pdfview` is inconsequent across the backend and engines: For example with `pdfview=FitH 100` `pdftex` ignores the number and calculates its own, while `dvips` sets the coordinate to the absolute 100. The zoom factor of `XYZ` is not supported by the `pdftex` driver at all, and `FitR` only partially.

The generic driver consolidate this but tries to stay compatible with the other drivers as far as possible. It also takes into account that `pdfview` and `pdfstartview` and `pdfremotestartview` have different requirements: While for the first relative coordinates are fine, for the two others absolute coordinates are more sensible.

`pdfview` (setup key) So with this driver the options `pdfview`, `pdfstartview` and `pdfremotestartview`
`pdfstartview` (setup key) take the following options:

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

- `pdfstartview` and `pdfremotestartview` will pass the optional numbers or keyword after expansion as absolute coordinates and zoom. Missing numbers will be filled up with `null`.

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

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

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

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

14 Assorted key descriptions

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

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

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

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

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

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

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

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

will create

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

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

`nesting` (setup key)

`pdfborder` (setup key)

`linkborder` (setup key)

`urlborder` (setup key)

`runborder` (setup key)

`menuborder` (setup key)

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

`pdfborderstyle` (setup key)

The value of this key is the content of the BS dictionary. As an example

`linkborderstyle` (setup key)

```
/Type/Border /W 1 /S/U /D[3 2]
```

`urlborderstyle` (setup key)

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

`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}[2021-06-14 v0.95e %
4   generic Hyperref driver for the LaTeX PDF management testphase bundle]
5
6 \RequirePackage{etoolbox} %why?
7 \ExplSyntaxOn
```

```

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

```

1 messages

Redirect the message name:

```

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

```

At first a message for the testing of the resource management

```

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

```

The pdfversion should be set in \DeclareDocumentMetadata

```

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

```

A generic message for ignored keys.

```

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

```

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

```

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

```

pdf/A message for not allowed Named actions

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

A message if the destination name is empty.

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

A message if the destination check fails

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

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

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

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

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

```

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

```

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

```

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

```

2 Variants

```

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

```

3 Overwriting/providing commands from hyperref

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

```

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

```

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

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

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

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

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

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

```
144 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
145 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:
146 \legacy_if:nT { Hy@setpdfversion }
147 {
148   \msg_warning:nn { hyp }{ pdfversion-disabled }
149 }
150 \Hy@DisableOption{pdfversion}
```

`\Acrobatmenu` should use the new internal link command

```
151 \RenewDocumentCommand \Acrobatmenu { m m }
152 {
153   \hyper@linknamed {#1} {#2}
154 }
```

`\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!

```
155 % TODO should go at some time ...
156 % \kv@set@family@handler{Hyp}
157 % { \msg_warning:nnx {hyp}{unknown-key-to-Hyp}{#1} }
158 \cs_set_protected:Npn \hypersetup #1
159 {
160   %\kvsetkeys{Hyp} {#1}
161   \keys_set:nn { hyp / setup }{ #1 }
162 }
163 % TODO for now unknown keys should only give warnings.
164 \keys_define:nn { hyp / setup }
165 {
166   unknown .code:n =
167   {
168     \msg_warning:nnxxx { hyp } { unknown-key }
169     { hyp~/~setup }{ \l_keys_key_str } { #1 }
170   }
171 }
```

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`)

```
172
173 \bool_new:N \l__hyp_href_url_encode_bool
174 \bool_new:N \l__hyp_href_url_ismap_bool
```

```

175 \tl_new:N \l__hyp_href_url_protocol_tl
176 \tl_new:N \l__hyp_href_pdf_destination_tl
177 \tl_new:N \l__hyp_href_pdf_page_tl
178 \tl_new:N \l__hyp_href_run_parameter_tl
179 \cs_new_protected:Npn \__hyp_href_url_format {\begingroup\Url}
180
181
182 \keys_define:nn { hyp / href }
183 {
184   ,urlencode .bool_set:N = \l__hyp_href_url_encode_bool
185   ,format .code:n = { \cs_set_eq:NN \__hyp_href_url_format #1 },
186   ,protocol .tl_set:N = \l__hyp_href_url_protocol_tl
187   ,destination .tl_set:N = \l__hyp_href_pdf_destination_tl
188   ,pdfremotestartview .code:n =
189   {
190     \keys_set:nn { hyp / setup }
191     { pdfremotestartview = #1 }
192   }
193   ,page .code:n =
194   {
195     \tl_set:Nn \l__hyp_href_pdf_page_tl {#1}
196     \tl_set:Nn \Hy@href@page {#1}
197   }
198   ,ismap .bool_set:N = \l__hyp_href_url_ismap_bool
199   ,run-parameter .tl_set:N = \l__hyp_href_run_parameter_tl
200   ,nextactionraw .code:n =
201   { %perhaps some safety match later, see hyperref code
202     \tl_if_empty:nTF {#1}
203     {
204       \pdfdict_remove:nn{l_hyp/annot/A}{Next}
205     }
206     {
207       \pdfdict_put:nnn{l_hyp/annot/A}{Next}{#1}
208       \tl_set:Nn \Hy@href@nextactionraw {/Next-#1}
209       \keys_set:nn {hyp / setup }{ pdfnewwindow = true}
210     }
211   }
212   ,afrelationship .code:n =
213   {
214     \pdfdict_put:nnx
215     { l_pdffile/Filespec}{AFRelationship}{ \pdf_name_from_unicode_e:n {#1}}
216   }
217 }
218
219
220 \keys_define:nn { hyp / setup }
221 {
222   ,href / urlencode .bool_set:N = \l__hyp_href_url_encode_bool
223   ,href / urlencode .default:n = {true}
224   ,href / urlencode .initial:n = {false}
225   ,href / protocol .tl_set:N = \l__hyp_href_url_protocol_tl
226   ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
227   ,href / format .code:n = { \cs_set_eq:NN \__hyp_href_url_format #1 }
228 }

```

```

229
230 \hook_new_pair:nn{cmd/href/before}{cmd/href/after}
231
232 \DeclareRobustCommand*{\href}[1][]{%
233   \mode_leave_vertical:
234   \hook_use:n{cmd/href/before}
235   \group_begin:
236   \keys_set:nn { hyp / href } {#1}
237   \let\protect\@unexpandable@protect
238   \bool_if:NTF \l__hyp_href_url_encode_bool
239     {
240       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
241     }
242     {
243       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
244     }
245   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
246 }
247
248 \begingroup
249 \catcode'\$=6 %
250 \catcode'\#=12 %
251 \gdef\href@\$1{\expandafter\href@split$1##\}%
252 \gdef\href@split$1#$2#$3\\$4{%
253   \hyper@link{$1}{$2}{$4}%<---__hyp-docstrip doubling!
254   \endgroup
255   \hook_use:n{cmd/href/after}
256 }%
257 \endgroup
258
259 \hook_new_pair:nn{cmd/hrefurl/before}{cmd/hrefurl/after}
260
261 \DeclareRobustCommand*{\hrefurl}[1][]{
262 {
263   \mode_leave_vertical:
264   \hook_use:n{cmd/href/before}
265   \group_begin:
266   \keys_set:nn { hyp / href } {#1}
267   \let\protect\@unexpandable@protect
268   \bool_if:NTF \l__hyp_href_url_encode_bool
269     {
270       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
271     }
272     {
273       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
274     }
275   \hyper@normalise\__hyp_href_url_aux:nn}
276
277 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
278 {
279   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
280   \group_end:
281   \hook_use:n{cmd/href/after}
282 }

```

```

283
284 \hook_new_pair:nn{cmd/hrefpdf/before}{cmd/hrefpdf/after}
285 \DeclareRobustCommand*{\hrefpdf}[1] []
286 {
287   \mode_leave_vertical:
288   \hook_use:n{cmd/hrefpdf/before}
289   \group_begin:
290   \keys_set:nn { hyp / href } {#1}
291   \let\protect\@unexpandable@protect
292   \hyper@normalise\__hyp_href_pdf_aux:nn
293 }
294
295 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
296 {
297   \exp_args:Nnno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
298   \group_end:
299   \hook_use:n{cmd/hrefpdf/after}
300 }
301
302 \hook_new_pair:nn{cmd/hrefrun/before}{cmd/hrefrun/after}
303 \DeclareRobustCommand*{\hrefrun}[1] []
304 {
305   \mode_leave_vertical:
306   \hook_use:n{cmd/hrefrun/before}
307   \group_begin:
308   \keys_set:nn { hyp / href } {#1}
309   \let\protect\@unexpandable@protect
310   \hyper@normalise\__hyp_href_run_aux:nn
311 }
312
313 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
314 {
315   \exp_args:Nnno\hyper@linklaunch{#2}{#1}{\l__hyp_href_run_parameter_tl}
316   \group_end:
317   \hook_use:n{cmd/hrefrun/after}
318 }
319
320
321 \hook_new_pair:nn{cmd/url/before}{cmd/url/after}
322
323 \DeclareRobustCommand*{\url}[1] []
324 {
325   \mode_leave_vertical:
326   \hook_use:n{cmd/url/before}
327   \group_begin:
328   \keys_set:nn { hyp / href } {#1}
329   \let\protect\@unexpandable@protect
330   \bool_if:NTF \l__hyp_href_url_encode_bool
331   {
332     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
333   }
334   {
335     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
336   }

```

```

337 \hyper@normalise\__hyp_href_url_aux:n
338 }
339
340 \cs_new_protected:Npn \__hyp_href_url_aux:n #1
341 {
342   \exp_args:Nno
343     \hyper@linkurl{\__hyp_href_url_format {#1}}
344     {\l__hyp_href_url_protocol_tl#1}
345   \group_end:
346   \hook_use:n{cmd/url/after}
347 }
348

```

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

```

349
350 \def\urldef#1#2{\begingroup\def\url{\hyper@normalise\url@}\setbox\z@\hbox\bgroup
351   \def\Url@HyperHook##1\endgroup{\Url@def{#1}{#2}}}%
352   % Because hyperref breaks \urldef and does not define its own (Grrrr!)...
353   \def\url@##1{\egroup\endgroup\DeclareRobustCommand#1{#2{##1}}}%
354   #2}
355

```

make the new commands compatible with `\pdfstringdef`:

```

356 \NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
357 \pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
358 \pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}
359 \pdfstringdefDisableCommands{\let\hrefrun\__hyp_secondoftwowithopt:wnn}

```

4 Compatibility commands

4.1 Metadata

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

```

360 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
361 {
362   %\tl_set:cn {@#1}{#2}
363   \AddToDocumentProperties[hyperref]{#1}{#2}
364 }
365 \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.

```

366 \hook_new:n{hyp/link/cite}
367 \color_set:nmn {hyp/color/cite}{HTML}{2E7E2A}
368 \color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
369 \keys_define:nn { hyp / setup }
370 {
371   ,citecolor .code:n = {\__hyp_color_set:ne {hyp/color/cite}{#1}}
372   ,citebordercolor

```

```

373     .code:n = {\__hyp_color_set:ne {hyp/color/citeborder}}{#1}}
374   }
375   \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
376   {
377     \keys_set:nn { hyp / setup }
378     {
379       ,linkbordercolor= hyp/color/citeborder
380       ,linkcolor      = hyp/color/cite
381     }
382   }

```

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

```

383   \bool_lazy_and:nnF
384   { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
385   { \msg_error:nn { hyp}{ missing-resource-management } }

```

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

```

386   \legacy_if:nT { Hy@bookmarks }
387   {
388     \AddToHook{begindocument/before}[hyperref/bookmark]{\RequirePackage{bookmark}}
389   }
390   \legacy_if:nT { Hy@draft }
391   {
392     \PassOptionsToPackage{draft}{bookmark}
393   }

```

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 394 %
\__hyp_ref_check:nn    395 \cs_new_protected:Npn \__hyp_ref_label:nn #1 #2 %label/attributes
\__hyp_ref_value:nn    396 {
397   \@bsphack
398   \ref_label:nn{#1}{#2}
399   \@esphack
400 }
401 \cs_generate_variant:Nn \__hyp_ref_label:nn {en}

```

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

```

402 \prg_new_eq_conditional:Nn \__hyp_ref_if_exist:nn \ref_if_exist:nn { p , T , F , TF }
403 \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.

```

404 \cs_new_protected:Npn \__hyp_ref_check:nn #1 #2 %label/attribute
405 {
406   \__hyp_ref_if_exist:nnF {#1}{#2}
407   {
408     \protect\G@refundefinedtrue
409     \@latex@warning
410     {
411       Reference~'\tl_to_str:n {#1}'~with~attribute~'\tl_to_str:n {#2}'~
412       on~page~\thepage~\space undefined
413     }
414   }
415 }
416 \cs_generate_variant:Nn \__hyp_ref_check:nn {en}

```

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

```

417 \cs_new:Npn \__hyp_ref_value:nn #1 #2
418 {
419   \ref_value:nn{#1}{#2}
420 }
421 \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 422 \box_new:N \l__hyp_tmpa_box
\l__hyp_tmpa_int 423 \tl_new:N \l__hyp_tmpa_tl
\l__hyp_tmpa_box 424 \seq_new:N \l__hyp_tmpa_seq
\l__hyp_tmpa_str 425 \int_new:N \l__hyp_tmpa_int
426 \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  TODO: document and check use!
\l__hyp_uri_tmpa_tl      427 \tl_new:N \l__hyp_dest_name_tmpa_tl
\l__hyp_filename_tmpa_tl 428 \tl_new:N \l__hyp_uri_tmpa_tl
\__hyp_text_tmpa_str_uuuu 429 \tl_new:N \l__hyp_filename_tmpa_tl
                          430 \tl_new:N \l__hyp_para_tmpa_tl
                          431 \str_new:N \l__hyp_text_tmpa_str
                          432 \str_new:N \g__hyp_text_tmpa_str

```

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

7.2 Constants

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

```
433 \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`

```
434 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
435 {url,link,file,menu,run}
436 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
437 {
438   URI    = url,
439   GoTo    = link,
440   GoToR   = file,
441   Named   = menu,
442   Launch  = run
443 }
444 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
445 {
446   url     = URI,
447   link    = GoTo,
448   file    = GoToR,
449   menu    = Named,
450   run     = Launch
451 }
452
```

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

7.3 Variables

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

```
453 \tl_new:N \g__hyp_dest_pdfstartpage_tl
454 \tl_new:N \g__hyp_dest_pdfstartview_tl
455 \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
456 \tl_new:N \l__hyp_text_enc_uri_print_tl
457 \tl_new:N \l__hyp_text_enc_info_print_tl
458 \tl_new:N \l__hyp_text_enc_dest_tl
459 \tl_new:N \l__hyp_text_enc_dest_print_tl
460 \tl_new:N \l__hyp_text_enc_file_print_tl
461 \tl_new:N \l__hyp_text_enc_para_print_tl
```

```

462
463 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
464 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
465 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
466 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
467 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
468 \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.

```

469 \tl_new:N \l__hyp_dest_pdfview_tl

```

(End definition for `\l__hyp_dest_pdfview_tl`.)

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

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

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

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

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

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

```

470 \str_new:N \g__hyp_bordercolormodel_str

```

(End definition for `\g__hyp_bordercolormodel_str`.)

7.4 Booleans

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

`\l_hyp_annot_colorurl_bool`

`\l_hyp_annot_colorfile_bool`

`\l_hyp_annot_colorrun_bool`

`\l_hyp_annot_colormenu_bool`

```

471 \seq_map_inline:Nn \c__hyp_annot_types_seq
472 {
473   \bool_new:c {l_hyp_annot_color#1_bool}
474 }

```

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

`\l_hyp_annot_ocgcolorlink_bool`

`\l_hyp_annot_ocgcolorurl_bool`

`\l_hyp_annot_ocgcolorfile_bool`

`\l_hyp_annot_ocgcolorrun_bool`

`\l_hyp_annot_ocgcolormenu_bool`

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

```

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

```

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

`\l_hyp_annot_Named_bool`

`\l_hyp_annot_Launch_bool`

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

```

479 \seq_map_inline:Nn \c_pdfannot_link_types_seq
480 {
481   \bool_new:c {l_hyp_annot_#1_bool}
482   \bool_set_true:c {l_hyp_annot_#1_bool}
483 }

```

(End definition for `\l_hyp_annot_GoTo_bool` `\l_hyp_annot_URI_bool` `\l_hyp_annot_GoToR_bool` `\l_hyp_annot_Named_bool` `\l_hyp_annot_Launch_bool`.)

7.5 Boxes

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

```
484 \box_new:N \l__hyp_dest_box
```

(End definition for `\l__hyp_dest_box`.)

7.6 Regex

`\c__hyp_dest_startview_regex` This regex is used to extract the right arguments pdfstartview and pdfremotestartview. Their values is filled up with null and then the start extracted.

```
485 \regex_const:Nn \c__hyp_dest_startview_regex
486 {
487   \A\ *
488   (?:
489     (?:XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
490     |
491     (?:Fit\b|FitB\b)
492     |
493     (?:\b(?:FitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1})
494     |
495     (?:FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
496   )
497 }
```

(End definition for `\c__hyp_dest_startview_regex`.)

7.7 PDF dictionaries

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

```
498 \pdfdict_new:n {l__hyp_page/Trans}
499 \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`

```
500 \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`

```

501 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
502 {
503     \str_set:Nx #2 {\text_purify:n { #1 } }
504 }

```

(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`

```

505 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
506 {
507
508 }

```

(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`

```

509 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
510 {
511     \pdf_string_from_unicode:nVN { #1 } #2 #2
512 }

```

(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`

```

513 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
514 {
515     \group_begin:
516     \Hy@pdfstringtrue
517     \hook_use:n {hyp/text/pdfstring}
518     \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
519     \__hyp_text_cleanup:N \l__hyp_text_tmpa_str
520     \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
521     \str_gset_eq:NN \g__hyp_text_tmpa_str \l__hyp_text_tmpa_str
522     \group_end:
523     \str_set_eq:NN #3 \g__hyp_text_tmpa_str
524 }
525 \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
526 \cs_new_protected:Npn \Hy@pstringdef #1 #2
527 { \_hyp\_text\_pdfstring:xnN {#2} {utf8/string-raw}#1 }
This is a special version for info keys:
```

```
\_hyp\_text\_pdfstring\_info:nN
528 \cs_new_protected:Npn \_hyp\_text\_pdfstring\_info:nN #1 #2
529 {
530   \_hyp\_text\_pdfstring:nonN { #1 } { \l\_hyp\_text\_enc\_info\_print\_tl } #2
531 }
```

```
(End definition for \_hyp\_text\_pdfstring\_info:nN.)
```

9 Pagelabels

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

The current implementation in hyperref/hyperref drivers:

xetex: hxdetex.def, line 80-110

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

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

dvipdfm: identical to xetex

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

luatex identical to pdftex

The code in hyperref inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-28 is correctly split. If the counter can not be identified hyperref generates only /P entries with the whole content.

The new implementation makes use of the pdf management: The relevant entry in the catalog is continuously updated and pushed out at the end of the document. This works (hopefully ...) with all drivers.

We do not try to avoid the (in hyperref’s wording) “useless” pagelabel entry `/PageLabels <</Nums[0<</S/D>>]>>` (but it would be possible), we also don’t test for empty `\thepage`, hyperref seems to handle this fine and the pdf is valid.

The code has to define `\Hy@PutCatalog` as we can’t yet change code in hyperref. The switch for draftmode has been removed.

```
\_hyp\_PageLabels\_gpush:
  \Hy@PutCatalog
  \HyPL@StorePageLabel
532 \cs_new_protected:Npn \_hyp\_PageLabels\_gpush:
533 {
534   \pdfmanagement\_add:nnx {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
```

```

535 }
536
537 \def\Hy@PutCatalog #1 {}
538
539
540 \legacy_if:nT { Hy@pdfpagelabels }
541 {
542   \cs_set_protected:Npn \HyPL@StorePageLabel #1
543   {
544     \tl_gput_right:Nx \HyPL@Labels { \the\Hy@abspage<<#1>> }
545     \__hyp_PageLabels_gppush:
546   }
547 }

```

(End definition for `__hyp_PageLabels_gppush:`, `\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

548 \Hy@WrapperDef \__hyp_destination:nn #1 #2
549 {
550   \mode_if_horizontal:T { \@savs\spacefactor }
551   \Hy@SaveLastskip      %defined in hyperref
552   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
553   \__hyp_text_pdfstring:xoN
554   { \HyperDestNameFilter{#1} }

```

```

555     { \l__hyp_text_enc_dest_tl }
556     \l__hyp_tmpa_tl
557 \str_if_eq:nnTF {#2} {fitrbox}
558 {
559     \exp_args:NV
560     \pdf_destination:nnnn \l__hyp_tmpa_tl
561     { \box_wd:N \l__hyp_dest_box }
562     { \box_ht:N \l__hyp_dest_box }
563     { \box_dp:N \l__hyp_dest_box }
564 }
565 {
566     \exp_args:NV
567     \pdf_destination:nf
568     { \l__hyp_tmpa_tl }
569     { #2 }
570 }
571 \Hy@RestoreLastskip %defined in hyperref
572 \mode_if_horizontal:T { \spacefactor\@savsf }
573 }

```

(End definition for `__hyp_destination:nn`.)

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

```

\hyper@anchor
\hyper@anchorstart
\hyper@anchorend
574 \cs_new_protected:Npn \hyper@anchor #1
575 {
576     \exp_args:NnV
577     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
578 }
579
580 \cs_new_protected:Npn \hyper@anchorstart #1
581 {
582     \Hy@activeanchortrue
583     \exp_args:NnV
584     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
585 }
586
587 \cs_new_protected:Npn \hyper@anchorend
588 {
589     \Hy@activeanchorfalse
590 }

```

(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:

```

591 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
592 {
593   \mode_leave_vertical:
594   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
595   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
596   {
597     \msg_warning:nnx
598     { hyp }
599     { empty-destination-name }
600     { \c__hyp_dest_undefined_tl }
601     \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
602   }
603   {
604     \__hyp_text_pdfstring:xoN
605     { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
606     { \l__hyp_text_enc_dest_tl }
607     \l__hyp_dest_name_tmpa_tl
608   }
609   \exp_args:No
610   \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
611 }
612
613 \cs_new_protected:Npn \__hyp_link_goto_end:
614 {
615   \pdfannot_link_goto_end:
616 }

```

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.

```

617
618 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
619 {

```

```

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

```

10.3 URI links

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

```

653 \pdfdict_new:n {l_hyp/annot/A/URI}
654 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
655 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
656
657 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
658 {
659   \bool_if:NTF \l__hyp_annot_URI_bool
660   {
661     \group_begin:
662     \__hyp_text_pdfstring:xoN
663     { #2}
664     { \l__hyp_text_enc_uri_print_tl }
665     \l__hyp_uri_tmpa_tl
666     \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
667     \bool_if:NT \l__hyp_href_url_ismap_bool
668     {
669       \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}

```

```

670     }
671     \cs_set_eq:NN \# \c_hash_str
672     \cs_set_eq:NN \% \c_percent_str
673     \Hy@safe@activestrue
674     \mode_leave_vertical:
675     \pdfannot_dict_put:nnx {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
676     \pdfannot_link:nxn { URI }
677     {
678     }
679     {
680         \let\protect\relax
681         #1
682         \Hy@xspace@end
683         \Hy@VerboseLinkStop %where is the start??
684     }
685     \group_end:
686 }
687 {\let\protect\relax#1}}
688 }
689

```

10.4 GoToR Links files

```

690 \pdfdict_new:n {l_hyp/annot/A/GoToR}
691 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
692 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
693
694
695 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
696 {
697     \bool_if:NTF \l__hyp_annot_GoToR_bool
698     {
699         \group_begin:
700         \tl_set:Nx \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
701         \exp_args:Nx
702         \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
703         {
704             \pdfdict_put:nnx { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n {ap
705             \exp_args:Nnox
706             \pdffile_embed_file:nnn
707             {}
708             {\l__hyp_filename_tmpa_tl }
709             {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
710         }
711         \pdfdict_put:nnx
712         {l_hyp/annot/A/GoToR}
713         {F}
714         {\pdf_object_ref:e {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
715         \__hyp_text_pdfstring:nnN
716         { #3 }
717         { \l__hyp_text_enc_dest_print_tl }
718         \l__hyp_dest_name_tmpa_tl
719         \tl_if_blank:eTF {#3}
720         {

```

```

721 \pdfdict_put:nnx {l_hyp/annot/A/GoToR}{D}
722 {
723   [
724     \int_eval:n
725     { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
726     /\l__hyp_dest_pdfremotestartview_tl
727   ]
728 }
729 }
730 {
731   \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
732 }
733 \mode_leave_vertical:
734 \pdf_object_unnamed_write:nx{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
735 \pdfannot_dict_put:nnx {link/GoToR}{A}{\pdf_object_ref_last:}
736 \pdfannot_link:nxn %expansion??
737 { GoToR }
738 {
739 }
740 {
741   \let\protect\relax
742   #1\Hy@xspace@end
743   \Hy@VerboseLinkStop %where is the start??
744 }
745 \group_end:
746 }
747 {{\let\protect\relax#1}}
748 }

```

10.5 Launch links

We define \hyper@linklaunch for naming consistency

```

749 \pdfdict_new:n {l_hyp/annot/A/Launch}
750 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
751 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
752
753 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
754 {
755   \bool_if:NTF \l__hyp_annot_Launch_bool
756   {
757     \group_begin:
758     \__hyp_text_pdfstring:nnN
759     { #1 }
760     { \l__hyp_text_enc_file_print_tl }
761     \l__hyp_filename_tmpa_tl
762     \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
763     \__hyp_text_pdfstring:noN
764     { #3 }
765     { \l__hyp_text_enc_para_print_tl }
766     \l__hyp_para_tmpa_tl
767     \bool_if:NTF
768     {
769       \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
770       ||

```

```

771         \pdf_version_compare_p:Nn > {1.9}
772     }
773     {
774         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
775     }
776     {
777         \pdfdict_put:nnx {l_hyp/annot/A/Launch}{Win}
778         {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
779     }
780     \mode_leave_vertical:
781     \pdfannot_dict_put:nnx {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}>>}
782     \pdfannot_link:nnx
783     { Launch }
784     {
785         % /A
786         % <<
787         % \pdfdict_use:n {l_hyp/annot/A/Launch}
788         % >>
789     }
790     {
791         \let\protect\relax
792         #2\Hy@xspace@end
793         \Hy@VerboseLinkStop %where is the start??
794     }
795     \group_end:
796 }
797 {\let\protect\relax#2}}
798 }

```

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

```

799 \use:x
800 { % filename, anchor text, linkname
801     \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str ##1 \exp_not:N \ \ ##2 #
802 }
803 {
804     \hyper@linklaunch {#1}{#2}{#3}
805 }

```

10.6 Named links (menu)

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

```

806 \pdfdict_new:n {l_hyp/annot/A/Named}
807 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
808 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
809
810 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
811 {
812     \bool_if:NTF \l__hyp_annot_Named_bool
813     {
814         \group_begin:
815         \pdfmeta_standard_verify:nnTF {named_actions}{#1}
816         {
817             \mode_leave_vertical:

```

```

818         \pdfdict_put:nnx {l_hyp/annot/A/Named}{N}
819         {\pdf_name_from_unicode_e:n{#1}}
820         \pdfannot_dict_put:nnx {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}>>}
821         \pdfannot_link:nxn { Named }
822         {
823             % /A
824             % <<
825             % \pdfdict_use:n { l_hyp/annot/A/Named }
826             % >>
827         }
828         {
829             #2
830             \Hy@xspace@end
831             \Hy@VerboseLinkStop
832         }
833     }
834     {
835         \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
836         #2
837     }
838     \group_end:
839 }
840 {{\let\protect\relax#2}}
841 }
842

```

11 Link decorations

11.1 Functions to export and select colors

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

```

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

```

This exports a color as space separated numbers as needed in in /C. $\{\langle color \rangle\}$ 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).

```

843 \cs_new_protected:Npn \__hyp_color_export:nnN #1 #2 #3
844 {
845     \tl_if_head_eq_charcode:nNTF {#1}[ %]
846     {
847         \__hyp_color_export_aux:wnnN #1 {#2} #3
848     }
849     {
850         \color_export:nnN {#1} {#2} #3
851     }
852 }
853

```

```

854 \cs_new_protected:Npn \__hyp_color_export_aux:wnnN [#1] #2 #3 #4
855 {
856   \color_export:nnnN {#1}{#2}{#3}#4
857 }
858
859 \cs_generate_variant:Nn \__hyp_color_export:nnN {xVN}

```

(End definition for __hyp_color_export:nnN and __hyp_color_export_aux:wnnN.)

```

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

```

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

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

```

860 \cs_new_protected:Npn \__hyp_color_select:n #1
861 {
862   \tl_if_head_eq_charcode:nNTF {#1}[ %]
863   {
864     \__hyp_color_select_aux:wn #1
865   }
866   {
867     \color_select:n {#1}
868   }
869 }
870
871 \cs_new_protected:Npn \__hyp_color_select_aux:wn [#1] #2
872 {
873   \color_select:nn {#1}{#2}
874 }
875
876 \cs_generate_variant:Nn \__hyp_color_select:n {e}

```

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

```

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

```

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

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

```

877 \cs_new_protected:Npn \__hyp_color_set:nn #1 #2
878 {
879   \tl_if_head_eq_charcode:nNTF {#2}[ %]
880   {
881     \__hyp_color_set_aux:nwn { #1 } #2
882   }
883   {
884     \color_set:nn {#1} {#2}
885   }
886 }
887

```

```

888 \cs_new_protected:Npn \__hyp_color_set_aux:nwn #1 [#2] #3
889 {
890     \color_set:nnn {#1}{#2}{#3}
891 }
892
893 \cs_generate_variant:Nn \__hyp_color_set:nn {ne}

```

(End definition for `__hyp_color_set:nn` and `__hyp_color_set_aux:nwn`.)

11.2 Textcolor of links

colors are added in the hooks. This means that they can also be removed if needed. They add a group—this isn’t needed with `hyperref` code, but could be relevant with low-level annotations.

```

894 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
895 {
896     \hook_gput_code:nnn
897     {pdfannot/link/#2/begin}
898     {hyp/color}
899     {
900         \bool_if:cT { l_hyp_annot_color#1_bool }
901         {
902             \group_begin:
903             \color_select:n { hyp/color/#1}
904         }
905     }
906     \hook_gput_code:nnn
907     {pdfannot/link/#2/end}
908     {hyp/color}
909     {
910         \bool_if:cT { l_hyp_annot_color#1_bool }
911         {
912             \group_end:
913         }
914     }
915 }

```

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

```

916 \keys_define:nn { hyp / setup }
917 {
918     ,colorlinks .meta:n =
919     {
920         ,pdfborder={0~0~0}
921         ,pdfborderstyle=
922         ,colorurl =#1
923         ,colorlink =#1
924         ,colorrun =#1
925         ,colormenu =#1
926         ,colorfile =#1
927     }
928     ,colorlinks .default:n = {true}
929 }

```

```

colorurl (setup key)
colorlink (setup key) 930 \seq_map_inline:Nn \c__hyp_annot_types_seq
colorrun (setup key) 931 {
colormenu (setup key) 932 \keys_define:nn { hyp / setup }
colorfile (setup key) 933 {
  urlcolor (setup key) 934 ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
linkcolor (setup key) 935 ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
  runcolor (setup key) 936 }
  menucolor (setup key) 937 }
  filecolor (setup key) 938
allcolor (setup key) 939 \keys_define:nn { hyp / setup }
 940 {
 941 ,allcolors .meta:n =
 942 {
 943 ,urlcolor=#1
 944 ,linkcolor=#1
 945 ,runcolor=#1
 946 ,filecolor=#1
 947 ,menucolor=#1
 948 }
 949 ,allcolors .value_required:n = true
 950 }

```

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)`

```

951 \keys_define:nn { hyp / setup }
952 {
953 ,bordercolormodel .choices:nn =
954 {rgb,cmyk}
955 { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
956 ,bordercolormodel .initial:n = {rgb}
957 }

958 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
959 {
960 \keys_define:nn { hyp / setup }
961 {
962 #1bordercolor .code:n =
963 {
964 \tl_if_empty:nTF { ##1 }
965 {
966 \pdfannot_dict_remove:nn
967 {link/#2}
968 { C }
969 }
970 {
971 \__hyp_color_export:xVN {##1}\g__hyp_bordercolormodel_str \l__hyp_tmpa_tl
972 \pdfannot_dict_put:nnx

```

```

973         {link/#2}
974         { C }
975         { [\l__hyp_tmpa_tl] }
976     }
977 }
978 }
979 }
980
981 \keys_define:nn { hyp / setup }
982 {
983     ,allbordercolors .meta:n =
984     {
985         ,linkbordercolor=#1
986         ,urlbordercolor =#1
987         ,filebordercolor=#1
988         ,menubordercolor=#1
989         ,runbordercolor =#1
990     }
991     ,allbordercolors .value_required:n = true
992 }
993

```

11.3.2 Borderwidth and -arc

```

994 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
995 {
996     \keys_define:nn { hyp / setup }
997     {
998         #1border .code:n =
999         {
1000             \tl_if_empty:nTF { ##1 }
1001             {
1002                 \pdfannot_dict_remove:nn
1003                 {link/#2}
1004                 { Border }
1005             }
1006             {
1007                 \pdfannot_dict_put:nnn
1008                 {link/#2}
1009                 { Border }
1010                 { [##1] }
1011             }
1012         }
1013     }
1014 }
1015 \keys_define:nn { hyp / setup }
1016 {
1017     ,pdfborder .code:n =
1018     {
1019         \tl_if_empty:nTF { #1 }
1020         {
1021             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1022             {
1023                 \pdfannot_dict_remove:nn

```

```

1024         {link/##2}
1025         { Border }
1026     }
1027 }
1028 {
1029     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1030     {
1031         \pdfannot_dict_put:nnn
1032         {link/##2}
1033         { Border }
1034         { [#1] }
1035     }
1036 }
1037 }
1038 ,pdfborder .initial:n = {0~0~1},
1039 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1040 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1041 {
runborderstyle (setup key) 1042     \keys_define:nn { hyp / setup }
fileborderstyle (setup key) 1043     {
menuborderstyle (setup key) 1044         #1borderstyle .code:n =
1045         {
1046             \tl_if_empty:nTF { ##1 }
1047             {
1048                 \pdfannot_dict_remove:nn
1049                 {link/#2}
1050                 { BS }
1051             }
1052             {
1053                 \pdfannot_dict_put:nnn
1054                 {link/#2}
1055                 { BS }
1056                 { <<##1>> }
1057             }
1058         }
1059     }
1060 }
1061 \keys_define:nn { hyp / setup }
1062 {
1063     ,pdfborderstyle .code:n =
1064     {
1065         \tl_if_empty:nTF { #1 }
1066         {
1067             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1068             {
1069                 \pdfannot_dict_remove:nn
1070                 {link/##2}
1071                 { BS }
1072             }

```

```

1073     }
1074     {
1075         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1076         {
1077             \pdfannot_dict_put:nnn
1078             {link/##2}
1079             { BS }
1080             { <<#1>> }
1081         }
1082     }
1083 }
1084 ,pdfborderstyle .initial:n = {},
1085 }

```

11.4 ocolorlinks

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

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

```

1086 \cs_new_protected:Npn \__hyp_ocg_init:
1087 {
1088     \pdf_object_new:nn { __hyp/OCG/View } { dict }
1089     \pdf_object_new:nn { __hyp/OCG/Print } { dict }
1090     \pdf_object_new:nn { __hyp/OCG/config } { dict }
1091     \pdf_object_new:nn { __hyp/OCG/refarray } { array }
1092     \pdf_object_write:nx { __hyp/OCG/refarray }
1093     {
1094         \pdf_object_ref:n { __hyp/OCG/View }
1095         \c_space_tl
1096         \pdf_object_ref:n { __hyp/OCG/Print }
1097     }
1098     \pdf_object_write:nn { __hyp/OCG/View }
1099     {
1100         /Type/OCG
1101         /Name(View)
1102         /Usage
1103         <<
1104             /Print <</PrintState/OFF>>~
1105             /View <</ViewState/ON >>~
1106         >>
1107     }
1108     \pdf_object_write:nn { __hyp/OCG/Print }
1109     {
1110         /Type/OCG
1111         /Name(Print)
1112         /Usage
1113         <<
1114             /Print <</PrintState/ON>>~
1115             /View <</ViewState/OFF>>~
1116         >>
1117     }
1118     \pdfmanagement_add:nnx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OC

```

```

1119 \pdfmanagement_add:nnx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG
1120 \pdf_object_write:nx { __hyp/OCG/config }
1121 {
1122   /OFF[\pdf_object_ref:n { __hyp/OCG/Print }]
1123   /AS[
1124     <<
1125     /Event/View
1126     /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1127     /Category[/View]
1128     >>
1129     <<
1130     /Event/Print
1131     /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1132     /Category[/Print]
1133     >>
1134     <<
1135     /Event/Export
1136     /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1137     /Category[/Print]
1138     >>
1139     ]
1140   }
1141 \pdfmanagement_add:nnx { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG
1142 \cs_gset:Npn \__hyp_ocg_init: {}
1143 }

```

(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.

```

1144 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1145 {
1146   \hook_gput_code:nnn
1147   {pdfannot/link/#2/begin}
1148   {hyp/ocg}
1149   {
1150     \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1151     {
1152       \__hyp_ocg_init:
1153       \group_begin:
1154       \hbox_set:Nw \l__hyp_tmpa_box
1155     }
1156   }
1157   \hook_gput_code:nnn
1158   {pdfannot/link/#2/end}
1159   {hyp/ocg}
1160   {
1161     \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1162     {
1163       \hbox_set_end:
1164       \mbox
1165       {
1166         \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1167         \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1168         \pdf_emc:

```

```

1169         \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1170     \group_begin:
1171     \color_select:n { hyp/color/#1 }
1172     \box_use_drop:N \l__hyp_tmpa_box
1173     \group_end:
1174     \pdf_emc:
1175 }
1176 \group_end:
1177 }
1178 }
1179 }

```

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

```

ocgcolorlink (setup key) 1180 \bool_lazy_or:nnTF
ocgcolorurl (setup key) 1181 { \pdf_version_compare_p:Nn > {1.4} }
ocgcolorfile (setup key) 1182 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
ocgcolormenu (setup key) 1183 {
1184     \keys_define:nn { hyp / setup }
1185     {
1186         ,_ocgcolorlinks .meta:n =
1187         {
1188             ocgcolorlink=#1,
1189             ocgcolorurl=#1,
1190             ocgcolorfile=#1,
1191             ocgcolorruncolor=#1,
1192             ocgcolormenu=#1
1193         }
1194         ,_ocgcolorlinks .default:n = true
1195     }
1196 }
1197 {
1198     \keys_define:nn { hyp / setup }
1199     {
1200         ,_ocgcolorlinks .code:n =
1201         {
1202             \msg_warning:nnxx
1203             { hyp }
1204             { ignore-deprecated-or-unknown-option-in-pdf-version }
1205             { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1206         }
1207     }
1208 }
1209
1210 \keys_define:nn { hyp / setup }
1211 {
1212     ,ocgcolorlinks .choice:
1213     ,ocgcolorlinks / true .meta:n =
1214     {
1215         pdfborder      = {0~0~0},
1216         pdfborderstyle = {},
1217         colorlinks      = false,
1218         _ocgcolorlinks = true
1219     }
1220     ,ocgcolorlinks / false .meta:n =

```

```

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

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1252 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1253 {
1254     \keys_define:nn { hyp / setup }
1255     {
1256         ,#1highlight .choices:nn =
1257         { /I, /N, /O, /P }
1258         {
1259             \pdfannot_dict_put:nnn
1260             {link/#2}
1261             { H }
1262             { ##1 }
1263         }
1264     }
1265     ,#1highlight / .code:n =
1266     {
1267         \pdfannot_dict_remove:nn
1268         {link/#2}
1269         { H }
1270     }

```

```

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

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1314 \keys_define:nn { hyp / setup }
hideurl (setup key) 1315 {
hidefile (setup key) 1316     hidelinks .meta:n =
hiderun (setup key) 1317     {
hidemenu (setup key) 1318         ,colorlinks = false

```

```

1319         ,ocgcolorlinks = false
1320         ,pdfborder      = { 0~0~0 }
1321         ,pdfborderstyle=
1322     }
1323 }
1324
1325 \seq_map_inline:Nn \c__hyp_annot_types_seq
1326 {
1327     \keys_define:nn { hyp / setup }
1328     {
1329         hide#1 .meta:n =
1330         {
1331             ,color#1      = false
1332             ,ocgcolor#1   = false
1333             ,#1border     = { 0~0~0 }
1334             ,#1borderstyle =
1335         }
1336     }
1337 }

```

11.7 color schemes and settings

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

`colorscheme` (*setup key*)

```

1338 \keys_define:nn { hyp / setup }
1339 {
1340     colorscheme .code:n =
1341     {
1342         \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1343         {
1344             \keys_set:nn { hyp /setup }
1345             {
1346                 ##1 = ##2
1347             }
1348         }
1349     }
1350 }
1351 \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*)

`pdfversion` (*setup key*)

```

1352 \keys_define:nn { hyp / setup }
1353 {
1354     ,unicode      .code:n = {}
1355     ,pdfencoding  .code:n = {}
1356     ,pdfversion  .code:n =
1357     {

```

```

1358         \msg_warning:nn { hyp }{ pdfversion-disabled }
1359     }
1360 }
1361 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
debug (setup key) 1362 \keys_define:nn { hyp / setup }
draft (setup key) 1363 {
final (setup key) 1364 ,verbose .choice:
1365 ,verbose / true .code:n = { \Hy@verbosetrue}
1366 ,verbose / false .code:n = { \Hy@verbosefalse}
1367 ,verbose .default:n = {true}
1368 ,debug .meta:n = {verbose=#1}
1369 ,debug .default:n = {true}
1370 }
1371 \keys_define:nn { hyp / setup }
1372 {
1373 ,draft .code:n =
1374 {
1375 \Hy@drafttrue
1376 \PassOptionsToPackage{draft}{bookmark}
1377 }
1378 ,final .code:n =
1379 {
1380 \Hy@finaltrue
1381 \PassOptionsToPackage{final}{bookmark}
1382 }
1383 }

extension (setup key)
hypertextnames (setup key) 1384 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1385 {
pageanchor (setup key) 1386 ,extension .tl_set:N = \XR@ext
linktoc (setup key) 1387 ,extension .initial:n= pdf
linktocpage (setup key) 1388 ,hypertextnames .choice:
plainpages (setup key) 1389 ,hypertextnames / true .code:n = { \Hy@hypertextnamestrue}
localanchorname (setup key) 1390 ,hypertextnames / false .code:n = { \Hy@hypertextnamesfalse}
linkfileprefix (setup key) 1391 ,hypertextnames .default:n = {true}
1392 ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1393 ,localanchorname .choice:
1394 ,localanchorname / true .code:n = { \Hy@localanchornamestrue }
1395 ,localanchorname / false .code:n = { \Hy@localanchornamefalse }
1396 ,localanchorname .default:n = {true}
1397 ,naturalnames .choice:
1398 ,naturalnames / true .code:n = { \Hy@naturalnamestrue}
1399 ,naturalnames / false .code:n = { \Hy@naturalnamesfalse}
1400 ,naturalnames .default:n = {true}
1401 ,pageanchor .choice:
1402 ,pageanchor / true .code:n = { \Hy@pageanchortrue}
1403 ,pageanchor / false .code:n = { \Hy@pageanchorfalse}

```

```

1404 ,pageanchor .default:n = {true}
1405 ,plainpages .choice:
1406 ,plainpages / true .code:n = { \Hy@plainpagetrue}
1407 ,plainpages / false .code:n = { \Hy@plainpagesfalse}
1408 ,plainpages .default:n = {true}
1409 }
1410
1411 \keys_define:nn { hyp / setup }
1412 {
1413 ,linktoc .choices:nn = { none, section, all, page }
1414 {
1415 \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1416 }
1417 ,linktoc / unknown .code:n =
1418 {
1419 \msg_warning:nxxx { hyp } { unknown-choice }
1420 { linktoc }
1421 { none, section, all, page }
1422 { \exp_not:n {#1} }
1423 }
1424 ,linktocpage .choice:
1425 ,linktocpage / true .meta:n = {linktoc=page}
1426 ,linktocpage / false .meta:n = {linktoc=section}
1427 ,linktocpage .default:n = true
1428 }
1429

```

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

```

url (setup key) 1430 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1431 {
menu (setup key) 1432 \keys_define:nn { hyp / setup }
run (setup key) 1433 {
1434 ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1435 }
1436 }

```

```

1437 \keys_define:nn { hyp / setup }
1438 {
1439 ,baseurl .code:n =
1440 {
1441 \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1442 \tl_if_empty:NTF \l__hyp_tmpa_tl
1443 {
1444 \pdfmanagement_remove:nn {Catalog} { URI }
1445 }
1446 {
1447 \pdfmanagement_add:nxx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1448 }
1449 }
1450 %only false does something ...
1451 ,bookmarks .choice:
1452 ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1453 ,bookmarks / true .code:n = {}
1454 ,bookmarks .default:n = {true}

```

```

1455 ,bookmarksnumbered .choice:
1456 ,bookmarksnumbered / false .code:n = { \Hy@bookmarksnumberedfalse }
1457 ,bookmarksnumbered / true .code:n = { \Hy@bookmarksnumberedtrue }
1458 ,bookmarksnumbered .default:n = {true}
1459 ,bookmarksopen .choice:
1460 ,bookmarksopen / false .code:n = { \Hy@bookmarksopenfalse }
1461 ,bookmarksopen / true .code:n = { \Hy@bookmarksopentruetrue }
1462 ,bookmarksopen .default:n = {true}
1463 ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1464 ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1465 ,pdfcenterwindow .choice:
1466 ,pdfcenterwindow / false .code:n =
1467 {
1468   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1469 }
1470 ,pdfcenterwindow / true .code:n =
1471 {
1472   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1473 }
1474 ,pdfcenterwindow / .code:n =
1475 {
1476   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1477 }
1478 ,pdfcenterwindow / unknown .code:n =
1479 {
1480   \msg_warning:nnxx { hyp } { no-bool }
1481   { pdfcenterwindow }
1482   { \exp_not:n {#1} }
1483 }
1484 ,pdfcenterwindow .default:n = true
1485 ,pdfdirection .choice:
1486 ,pdfdirection / L2R .code:n =
1487 {
1488   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1489 }
1490 ,pdfdirection / R2L .code:n =
1491 {
1492   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1493 }
1494 ,pdfdirection / .code:n =
1495 {
1496   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1497 }
1498 ,pdfdirection / unknown .code:n =
1499 {
1500   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1501   { pdfdirection }
1502   { L2R , R2L }
1503   { \exp_not:n {#1} }
1504 }
1505 ,pdfdisplaydoctitle .choice:
1506 ,pdfdisplaydoctitle / false .code:n =
1507 {
1508   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }

```

```

1509     }
1510 ,pdftitle / true .code:n =
1511     {
1512         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1513     }
1514 ,pdftitle .default:n = true
1515 ,pdfduplex .choices:nn =
1516     {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1517     {
1518         \pdf_version_compare:NnTF > {1.6}
1519         {
1520             \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1521             { PrintDuplex } { /#1 }
1522         }
1523         {
1524             \msg_warning:nnxx
1525             {hyp}
1526             {ignore-deprecated-or-unknown-option-in-pdf-version}
1527             {pdfduplex}
1528             {\pdf_version:}
1529         }
1530     }%
1531 ,pdfduplex / .code:n =
1532     {
1533         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1534     }
1535 ,pdfduplex / unknown .code:n =
1536     {
1537         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1538         { pdfduplex }
1539         { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1540         { \exp_not:n {#1} }
1541     }
1542 ,pdffitwindow .choice:
1543 ,pdffitwindow / false .code:n =
1544     {
1545         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1546     }
1547 ,pdffitwindow / true .code:n =
1548     {
1549         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1550     }
1551 ,pdffitwindow / .code:n =
1552     {
1553         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1554     }
1555 ,pdffitwindow .default:n = true
1556 ,pdffitwindow / unknown .code:n =
1557     {
1558         \msg_warning:nnxx { hyp } { no-bool }
1559         { pdffitwindow }
1560         { \exp_not:n {#1} }
1561     }
1562 ,pdfmargin .code:n = { \pdfannot_link_margin:n { #1 } }

```

```

1563 ,pdflinkmargin .initial:n = {1pt}
1564 ,pdfmenubar .choice:
1565 ,pdfmenubar / true .code:n =
1566 {
1567     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1568 }
1569 ,pdfmenubar / false .code:n =
1570 {
1571     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1572     { HideMenubar } { true }
1573 }
1574 ,pdfmenubar / .code:n =
1575 {
1576     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1577 }
1578 ,pdfmenubar .default:n = true
1579 ,pdfmenubar / unknown .code:n =
1580 {
1581     \msg_warning:nnxx { hyp } { no-bool }
1582     { pdfmenubar }
1583     { \exp_not:n {#1} }
1584 }
1585 ,pdfnewwindow .choice:
1586 ,pdfnewwindow / true .code:n =
1587 {
1588     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1589     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1590 }
1591 ,pdfnewwindow / false .code:n =
1592 {
1593     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1594     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1595 }
1596 ,pdfnewwindow / .code:n =
1597 {
1598     \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1599     \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1600 }
1601 ,pdfnonfullscreenpagemode .choices:nn =
1602 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1603 {
1604     \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1605     { NonFullScreenPageMode } {/#1}
1606 }
1607 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1608 {
1609     \pdf_version_compare:NnTF < {1.6}
1610     {
1611         %message
1612     }
1613     {
1614         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1615         {NonFullScreenPageMode}{/UseAttachments}
1616     }

```

```

1617     }
1618 ,pdfnonfullscreenpagemode / .code:n =
1619 {
1620     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1621 }
1622 ,pdfnonfullscreenpagemode / unknown .code:n =
1623 {
1624     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1625     { pdfnonfullscreenpagemode }
1626     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1627     { \exp_not:n {#1} }
1628 }
1629 ,pdfnumcopies .code:n =
1630 {
1631     \pdf_version_compare:NnTF > {1.6}
1632     {
1633         \tl_if_empty:nTF {#1}
1634         {
1635             \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1636         }
1637         {
1638             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1639             {NumCopies}{#1}
1640         }
1641     }
1642     {
1643         \msg_warning:nnxx
1644         {hyp}
1645         {ignore-deprecated-or-unknown-option-in-pdf-version}
1646         {pdfnumcopies}
1647         {\pdf_version:}
1648     }
1649 }
1650 ,pdfpagelayout .choices:nn =
1651 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1652 { \pdfmanagement_add:nnx {Catalog} { PageLayout }{ /#1 } }
1653 ,pdfpagelayout / .code:n =
1654 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1655 ,pdfpagelayout / unknown .code:n =
1656 {
1657     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1658     { pdfpagelayout }
1659     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1660     { \exp_not:n {#1} }
1661 }
1662 ,pdfpagemode .choices:nn =
1663 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1664 { \pdfmanagement_add:nnx {Catalog} { PageMode }{ /#1 } }
1665 ,pdfpagemode / UseAttachments .code:n =
1666 {
1667     \pdf_version_compare:NnTF > {1.5}
1668     {
1669         \pdfmanagement_add:nnx {Catalog} { PageMode }{ /UseAttachments }
1670     }

```

```

1671     {
1672         \msg_warning:nnxx
1673         {hyp}
1674         {ignore-deprecated-or-unknown-value-in-pdf-version}
1675         {UseAttachments}
1676         {\pdf_version:}
1677     }
1678 }
1679 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1680 ,pdfpagemode / unknown .code:n =
1681 {
1682     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1683     { pdfpagemode }
1684     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1685     { \exp_not:n {#1} }
1686 }
1687 ,pdfpagescrop .code:n =
1688 {
1689     \tl_if_empty:nTF {#1} %or blank?
1690     {
1691         \pdfmanagement_remove:nn {Pages} { CropBox }
1692     }
1693     {
1694         \pdfmanagement_add:nnx {Pages} { CropBox } { [#1] }
1695     }
1696 }
1697 ,pdfpicktraybypdfsize .choice:
1698 ,pdfpicktraybypdfsize / true .code:n =
1699 {
1700     \pdf_version_compare:NnTF > {1.6}
1701     {
1702         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1703         { PickTrayByPDFSize } { true }
1704     }
1705     {
1706         \msg_warning:nnxx
1707         {hyp}
1708         {ignore-deprecated-or-unknown-option-in-pdf-version}
1709         {pdfpicktraybypdfsize}
1710         {\pdf_version:}
1711     }
1712 }
1713 ,pdfpicktraybypdfsize / false .code:n =
1714 {
1715     \pdf_version_compare:NnTF > {1.6}
1716     {
1717         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1718         { PickTrayByPDFSize } { false }
1719     }
1720     {
1721         \msg_warning:nnxx
1722         {hyp}
1723         {ignore-deprecated-or-unknown-option-in-pdf-version}
1724         {pdfpicktraybypdfsize}

```

```

1725         {\pdf_version:}
1726     }
1727 }
1728 ,pdfpicktraybypdfsize / .code:n =
1729 {
1730     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1731 }
1732 ,pdfpicktraybypdfsize / unknown .code:n =
1733 {
1734     \msg_warning:nxxx { hyp } { no-bool }
1735     { picktraybypdfsize }
1736     { \exp_not:n {#1} }
1737 }
1738 ,pdfprintarea .choices:nn =
1739 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1740 {
1741     \pdf_version_compare:NnTF < {2.0}
1742     {
1743         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1744         { PrintArea } { /#1 }
1745     }
1746     {
1747         \msg_warning:nxxx
1748         {hyp}
1749         {ignore-deprecated-or-unknown-option-in-pdf-version}
1750         {pdfprintarea}
1751         {\pdf_version:}
1752     }
1753 }%
1754 ,pdfprintarea / .code:n =
1755 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1756 ,pdfprintarea / unknown .code:n =
1757 {
1758     \msg_warning:nxxx { hyp } { unknown-choice+empty }
1759     { pdfprintarea }
1760     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1761     { \exp_not:n {#1} }
1762 }
1763 ,pdfprintclip .choices:nn =
1764 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1765 {
1766     \pdf_version_compare:NnTF < {2.0}
1767     {
1768         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1769         { PrintClip } { /#1 }
1770     }
1771     {
1772         \msg_warning:nxxx
1773         {hyp}
1774         {ignore-deprecated-or-unknown-option-in-pdf-version}
1775         {pdfprintclip}
1776         {\pdf_version:}
1777     }
1778 }%

```

```

1779 ,pdfprintclip / .code:n =
1780 {
1781     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1782 }
1783 ,pdfprintclip / unknown .code:n =
1784 {
1785     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1786     { pdfprintclip }
1787     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1788     { \exp_not:n {#1} }
1789 }
1790 ,pdfprintpagerange .code:n =
1791 {
1792     \pdf_version_compare:NnTF > {1.6}
1793     {
1794         \tl_if_empty:nTF { #1}
1795         {
1796             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1797             { PrintPageRange }
1798         }
1799         {
1800             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1801             {PrintPageRange}{[#1]}
1802         }
1803     }
1804     {
1805         \msg_warning:nnxx
1806         {hyp}
1807         {ignore-deprecated-or-unknown-option-in-pdf-version}
1808         {pdfprintpagerange}
1809         {\pdf_version:}
1810     }
1811 }
1812 ,pdfprintscaling .choices:nn =
1813 { None, AppDefault }
1814 {
1815     \pdf_version_compare:NnTF > {1.5}
1816     {
1817         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1818         { PrintScaling } { /#1 }
1819     }
1820     {
1821         \msg_warning:nnxx
1822         {hyp}
1823         {ignore-deprecated-or-unknown-option-in-pdf-version}
1824         {pdfprintscaling}
1825         {\pdf_version:}
1826     }
1827 }%
1828 ,pdfprintscaling / .code:n =
1829 {
1830     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1831 }
1832 ,pdfprintscaling / unknown .code:n =

```

```

1833     {
1834         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1835         { pdfprintarea }
1836         { None, AppDefault }
1837         { \exp_not:n {#1} }
1838     }
1839 ,pdfremotestartview .code:n =
1840 {
1841     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1842     \exp_args:NNV
1843     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1844     {
1845         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1846     }
1847     {
1848         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1849         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1850     }
1851 }
1852 ,pdfremotestartview .initial:n = {Fit}
1853 % pdfstartpage is special as it shares code with pdfstartview
1854 ,pdfstartpage .code:n =
1855 {
1856     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1857     \bool_if:nTF
1858     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1859     {
1860         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1861     }
1862     {
1863         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1864         {
1865             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1866         }
1867     }
1868 }
1869 ,pdfstartpage .initial:n =1
1870 ,pdfstartview .code:n =
1871 {
1872     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1873     \exp_args:NNV
1874     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1875     {
1876         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1877     }
1878     {
1879         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1880         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1881     }
1882     \bool_if:nTF
1883     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1884     {
1885         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1886     }

```

```

1887     {
1888         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1889         {
1890             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1891         }
1892     }
1893 }
1894 ,pdfstartview .initial:n = Fit
1895 ,pdftoolbar .choice:
1896 ,pdftoolbar / true .code:n =
1897 {
1898     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1899 }
1900 ,pdftoolbar / false .code:n =
1901 {
1902     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
1903     { HideToolbar } { true }
1904 }
1905 ,pdftoolbar / true .code:n =
1906 {
1907     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1908 }
1909 ,pdftoolbar .default:n = true
1910 ,pdftoolbar / unknown .code:n =
1911 {
1912     \msg_warning:nnxx { hyp } { no-bool }
1913     { pdftoolbar }
1914     { \exp_not:n {#1} }
1915 }
1916 % pdfview see below.
1917 ,pdfviewarea .choices:nn =
1918 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1919 {
1920     \pdf_version_compare:NnTF < {2.0}
1921     {
1922         \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
1923         { ViewArea } { /#1 }
1924     }
1925     {
1926         \msg_warning:nnxx
1927         {hyp}
1928         {ignore-deprecated-or-unknown-option-in-pdf-version}
1929         {pdfviewarea}
1930         {\pdf_version:}
1931     }
1932 }%
1933 ,pdfviewarea / .code:n =
1934 {
1935     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { ViewArea }
1936 }
1937 ,pdfviewarea / unknown .code:n =
1938 {
1939     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1940     { pdfviewarea }

```

```

1941         { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1942         { \exp_not:n {#1} }
1943     }
1944     ,pdfviewclip      .choices:nn =
1945     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1946     {
1947         \pdf_version_compare:NnTF < {2.0}
1948         {
1949             \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1950             { ViewClip } { /#1 }
1951         }
1952         {
1953             \msg_warning:nnxx
1954             {hyp}
1955             {ignore-deprecated-or-unknown-option-in-pdf-version}
1956             {pdfviewclip}
1957             {\pdf_version:}
1958         }
1959     }%
1960     ,pdfviewclip / .code:n =
1961     {
1962         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
1963     }
1964     ,pdfviewclip / unknown .code:n =
1965     {
1966         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1967         { pdfviewclip }
1968         { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1969         { \exp_not:n {#1} }
1970     }
1971     ,pdfwindowui      .choice:
1972     ,pdfwindowui / true .code:n =
1973     {
1974         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
1975     }
1976     ,pdfwindowui / false .code:n =
1977     {
1978         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1979         { HideWindowUI } { true }
1980     }
1981     ,pdfwindowui / .code:n =
1982     {
1983         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
1984     }
1985     ,pdfwindowui / unknown .code:n =
1986     {
1987         \msg_warning:nnxx { hyp } { no-bool }
1988         { pdfwindowui }
1989         { \exp_not:n {#1} }
1990     }
1991     ,pdfwindowui      .default:n = true
1992 }

```

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

```

1993 \keys_define:nn { hyp / setup }
1994 {
1995   ,pdfview .code:n =
1996   {
1997     \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
1998     \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } } {
1999       {
2000         { xyz }
2001         {
2002           \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
2003           {
2004             \seq_get_right:Nn \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2005             \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2006             {
2007               \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2008             }
2009             {
2010               \tl_set:Nx \l__hyp_dest_pdfview_tl
2011               {
2012                 \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2013               }
2014             }
2015           }
2016           {
2017             \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2018           }
2019         }
2020         { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2021         { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2022         { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2023         { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2024         { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2025         { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2026         { fitr }
2027         {
2028           \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2029           {
2030             \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2031           }
2032           {
2033             %ensure 4 values ...
2034             \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2035             \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2036             \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2037             \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2038             \hbox_set_to_wd:Nnn \l__hyp_dest_box
2039             {
2040               \fp_eval:n
2041               {
2042                 round
2043                 (
2044                   abs
2045                   (
2046                     \seq_item:Nn \l__hyp_tmpa_seq {4}

```

```

2047             -
2048             (\seq_item:Nn\l__hyp_tmpa_seq{2})
2049         ),
2050         3
2051     )
2052 }bp
2053 }{}
2054 \box_set_dp:Nn \l__hyp_dest_box
2055 {
2056     \fp_eval:n
2057     {
2058         round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2059     }bp
2060 }
2061 \box_set_ht:Nn \l__hyp_dest_box
2062 {
2063     \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2064 }
2065 }
2066 }
2067 }
2068 {
2069     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2070     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2071 }
2072 }
2073 ,pdfview .initial:n = {xyz}
2074 }

```

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.

```

2075 \keys_define:nn { hyp / setup }
2076 {
2077     ,pdflang .code:n =
2078     {
2079         \tl_if_empty:nTF { #1 }
2080         {
2081             \pdfmanagement_remove:nn {Catalog} { Lang }
2082         }
2083         {
2084             \pdfmanagement_add:nnx {Catalog} { Lang } { (#1) }
2085         }
2086         \__hyp_store_metadata:nn {pdflang}{#1}
2087     }
2088 }

```

12.3.1 “info keys”

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

pdftitle (*setup key*)

pdfcreator (*setup key*)

pdfsubject (*setup key*)

pdfproducer (*setup key*)

pdfkeywords (*setup key*)

```

2089 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2090 {
2091   \keys_define:nn { hyp / setup }
2092   {
2093     pdf#1 .code:n =
2094     {
2095       \tl_if_blank:nTF {##1}
2096       {
2097         \pdfmanagement_remove:nn {Info}{#2}
2098       }
2099       {
2100         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2101         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2102         {
2103           \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2104         }
2105       }
2106       \__hyp_store_metadata:nn {pdf#1}{##1}
2107     }
2108   }
2109   \keys_define:nn { hyp / info }
2110   {
2111     #2 .code:n =
2112     {
2113       \tl_if_blank:nTF {##1}
2114       {
2115         \pdfmanagement_remove:nn {Info}{#2}
2116       }
2117       {
2118         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2119         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2120         {
2121           \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2122         }
2123       }
2124       \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2125     }
2126     ,unknown .code:n =
2127     {
2128       \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2129       \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2130       {
2131         \exp_args:Nno
2132         \pdfmanagement_add:nnx {Info}
2133         { \l_keys_key_str } {\l__hyp_tmpa_str}
2134       }
2135     }
2136   }
2137 }
2138 \__hyp_setup_info_key:nn {author} {Author}
2139 \__hyp_setup_info_key:nn {title} {Title}
2140 \__hyp_setup_info_key:nn {producer} {Producer}
2141 \__hyp_setup_info_key:nn {creator} {Creator}
2142 % ignored key: addtopdfcreator

```

```

2143 \_hyp_setup_info_key:nn {subject} {Subject}
2144 \_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.

```

2145 \cs_new_protected:Npn \_hyp_setup_info_date_key:nn #1 #2
2146 {
2147   \keys_define:nn { hyp / setup }
2148   {
2149     pdf#1 .code:n =
2150     {
2151       \tl_if_blank:nTF {##1}
2152       {
2153         \pdfmanagement_remove:nn {Info}{#2}
2154       }
2155       {
2156         \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
2157       }
2158       \_hyp_store_metadata:nn {pdf#1}{##1}
2159     }
2160   }
2161   \keys_define:nn { hyp / info }
2162   {
2163     #2 .code:n =
2164     {
2165       \tl_if_blank:nTF {##1}
2166       {
2167         \pdfmanagement_remove:nn {Info}{#2}
2168       }
2169       {
2170         \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
2171       }
2172       \exp_args:Nx \_hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2173     }
2174   }
2175 }
2176
2177 \_hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2178 \_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 ...

```

2179 \keys_define:nn { hyp / setup }
2180 {
2181   ,pdftrapped .code:n =
2182   {
2183     \exp_args:Nne
2184     \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2185   }
2186   ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2187   {
2188     \pdfmanagement_add:nnx {Info}{Trapped}
2189     {/
2190       \str_uppercase:f { \str_head:n { #1 } }
2191       \str_lowercase:f { \str_tail:n { #1 } }

```

```

2192     }
2193     \__hyp_store_metadata:nx {pdftrapped}
2194     {
2195         \str_uppercase:f { \str_head:n { #1 } }
2196         \str_lowercase:f { \str_tail:n { #1 } }
2197     }
2198 }
2199 ,_pdftrapped / unknown .code:n =
2200 {
2201     \msg_warning:nnxxx { hyp } { unknown-choice }
2202     { pdftrapped }
2203     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2204     { \exp_not:n {#1} }
2205 }
2206 }

```

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

```

2207 \keys_define:nn { hyp / setup }
2208 {
2209     pdfinfo .code:n =
2210     {
2211         \keys_set:nn { hyp / info } { #1 }
2212     }
2213 }

```

Now we set some default values

```

2214 \keys_set:nn { hyp / setup } {pdfcreator = LaTeX-with-hyperref}
2215 \keys_set:nn { hyp / setup } {pdfauthor = }
2216 \keys_set:nn { hyp / setup } {pdftitle = }
2217 \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:

```

2218 \clist_map_inline:nn
2219 {
2220     ,pdfcopyright
2221     ,pdftype
2222     ,pdflicenseurl
2223     ,pdfauthortitle
2224     ,pdfcaptionwriter
2225     ,pdfmetalang
2226     ,pdfapart
2227     ,pdfaconformance
2228     ,pdfuapart
2229     ,pdfxstandard
2230     ,pdfsource
2231     ,pdfdocumentid
2232     ,pdfinstanceid
2233     ,pdfversionid
2234     ,pdfrendition
2235     ,pdfpublication

```

```

2236 ,pdfpubtype
2237 ,pdfbytes
2238 ,pdfnumpages
2239 ,pdfissn
2240 ,pdfeissn
2241 ,pdfisbn
2242 ,pdfbookedition
2243 ,pdfpublisher
2244 ,pdfvolumenum
2245 ,pdfissuenum
2246 ,pdfpagerange
2247 ,pdfdoi
2248 ,pdfurl
2249 ,pdfidentifier
2250 ,pdfsubtitle
2251 ,pdfpubstatus
2252 ,pdfcontactaddress
2253 ,pdfcontactcity
2254 ,pdfcontactregion
2255 ,pdfcontactpostcode
2256 ,pdfcontactcountry
2257 ,pdfcontactphone
2258 ,pdfcontactemail
2259 ,pdfcontacturl
2260 ,pdfdate
2261 }
2262 {
2263   \keys_define:nn { hyp / setup }
2264   {
2265     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2266   }
2267 }
2268
2269 \str_if_eq:eeT {A}{\pdfmeta_standard_item:n{type}}
2270 {
2271   \exp_args:Nnx
2272   \keys_set:nn {hyp / setup}
2273   {
2274     ,pdfapart = \pdfmeta_standard_item:n {level}
2275     ,pdfaconformance = \pdfmeta_standard_item:n {conformance}
2276   }
2277 }

```

12.5 Transitions

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

```

2278 \keys_define:nn { hyp / setup }
2279 {
2280   pdfpageduration .code:n =
2281   {
2282     \tl_if_blank:nTF { #1 }
2283     {
2284       \pdfmanagement_remove:nn {Page}{Dur}
2285     }

```

```

2286     {
2287         \pdfmanagement_add:nnn {Page}{Dur}{#1}
2288     }
2289 }
2290 }

```

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 duration of the transition
direction	H (horizontal, only Split, B), V (vertical, only Split, B), 0 (left to right, only Wipe), 90 (bottom to top, only Wipe), 180 (right to left, only Wipe), 270 (top to bottom, only Wipe), 315 (top left to bottom, only Wipe), None (only Fly)
motion	one of I, O, only relevant for Fly and Fly
scale	a number, only relevant for Fly
opaque	true or false, only relevant for Fly

The keys allowed in the argument of `pdfpagetransition` are

```

2291 \keys_define:nn { hyp / setup }
2292 {
2293     pdfpagetransition .code:n =
2294     {
2295         \tl_if_blank:nTF {#1}
2296         {
2297             \pdfmanagement_remove:nn {Page}{Trans}
2298         }
2299         {
2300             \group_begin:
2301             \keys_set:nn { hyp / trans }{style=R,#1}
2302             \pdf_object_unnamed_write:nx { dict }
2303             {
2304                 \pdfdict_use:n {l_hyp_page/Trans}
2305             }
2306             \pdfmanagement_add:nnx {Page}{Trans}{\pdf_object_ref_last:}
2307             \group_end:
2308         }
2309     }
2310 }

```

```

2311 \keys_define:nn { hyp / trans }
2312 {
2313   ,style .choices:nn =
2314     {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2315     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2316   ,style / unknown .code:n =
2317     {
2318       \msg_warning:nnxxx { hyp } { unknown-choice }
2319       { trans / style }
2320       { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2321       { \exp_not:n {#1} }
2322     }
2323   ,duration .code:n =
2324     {
2325       \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}
2326     }
2327   ,direction .choices:nn =
2328     {H,V}
2329     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2330   ,direction .choices:nn =
2331     {0,90,180,270,315}
2332     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ #1 } }
2333   ,direction / None .code:n =
2334     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ /None } }
2335   ,direction / unknown .code:n =
2336     {
2337       \msg_warning:nnxxx { hyp } { unknown-choice }
2338       { trans / direction }
2339       {
2340         H~(horizontal,~only~Split,~Blinds),
2341         V~(vertical,~only~Split,~Blinds),
2342         0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2343         90~(bottom~to~top,~only~Wipe),
2344         180~(right~to~left,~only~Wipe),
2345         270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2346         315~(top~left~to~bottom,~only~Glitter),
2347         None~(only~Fly)
2348       }
2349       { \exp_not:n {#1} }
2350     }
2351   ,motion .choices:nn =
2352     {I,0}
2353     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2354   ,motion / unknown .code:n =
2355     {
2356       \msg_warning:nnxxx { hyp } { unknown-choice }
2357       { trans / motion }
2358       { I~(inwards) , 0~(outwards) }
2359       { \exp_not:n {#1} }
2360     }
2361   ,scale .code:n =
2362     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2363   ,opaque .choices:nn = {true,false}
2364     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }

```

```

2365 ,opaque / unknown .code:n =
2366 {
2367     \msg_warning:nnxxx { hyp } { unknown-choice }
2368     { trans / B }
2369     { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2370     { \exp_not:n {#1} }
2371 }
2372 % try to set unknown keys as style
2373 ,unknown .code:n =
2374 {
2375     % warning ...
2376     \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2377 }
2378 }

```

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

```

2379 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}

```

Unfinished Form field code

```

2380 \NewDocumentCommand \MakeFieldObject { m m }
2381 {
2382     \pdfxform_new:nnn { #2 }{} { #1 }
2383 }
2384
2385
2386 \prop_new:N    \g__hyp_AcroForm_CoFields_prop
2387 \prop_new:N    \g__hyp_AcroForm_Fields_prop
2388
2389 \let\HyField@afields\ltx@empty
2390 \let\HyField@cofields\ltx@empty
2391 %% UF test for old pdftex removed
2392 %\let\HyField@AuxAddToFields\ltx@gobble
2393 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2394 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2395
2396 % the value doesn't matter, but with a prop we avoid duplicates and it is
2397 % clearly faster than removing them from a sequence
2398 \def\HyField@AuxAddToFields#1
2399 {
2400     \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2401 }%
2402
2403 %fields with empty key get a value too -- lets hope that
2404 %this give the expected behaviour
2405 \def\HyField@AuxAddToCoFields #1 #2
2406 {
2407     \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2408 }
2409
2410 \Hy@AtBeginDocument
2411 {
2412     \if@filesw
2413         \immediate\write\@mainaux{%
2414             \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2415         }%

```

```

2416         \immediate\write\@mainaux{%
2417             \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2418         }%
2419     \fi
2420     \let\HyField@AfterAuxOpen\@firstofone
2421 }%
2422
2423 \def\HyField@AddToFields
2424 {
2425     \exp_args:Nx\HyField__hypAddToFields
2426     {
2427         \pdfannot_box_ref_last:
2428     }
2429     \ifx\Fld@calculate@code\ltx@empty
2430     \else
2431         \begingroup
2432         \Hy@safe@activetrue
2433         \edef\Hy@temp{%
2434             \endgroup
2435             \if@filesw
2436                 \write\@mainaux
2437                 {
2438                     \string\HyField@AuxAddToCoFields
2439                     {
2440                         \Fld@calculate@sortkey
2441                     }
2442                     {
2443                         \pdfannot_box_ref_last:
2444                     }
2445                 }
2446             \fi
2447         }%
2448         \Hy@temp
2449     \fi
2450 }%
2451
2452 \def\HyField__hypAddToFields#1{
2453     \HyField@AfterAuxOpen{%
2454         \if@filesw
2455             \write\@mainaux{%
2456                 \string\HyField@AuxAddToFields{#1}%
2457             }%
2458         \fi
2459     }%
2460 }%
2461
2462 \ExplSyntaxOff
2463 \ExplSyntaxOn
2464 \tl_new:N \l__hyp_CheckmarkYes_tl
2465 \tl_set:Nn \l__hyp_CheckmarkYes_tl { __hyp_xform_CheckMarkYes }
2466 \tl_new:N \l__hyp_CheckmarkOff_tl
2467 \tl_set:Nn \l__hyp_CheckmarkOff_tl { __hyp_xform_CheckMarkOff }
2468
2469 \def\@Form[#1]

```

```

2470 {
2471   \@ifundefined{textcolor}{\let\textcolor\@gobble}{ }
2472   \kvsetkeys{Form}{#1}
2473   \pdf@ifdraftmode{
2474     {
2475       \Hy@FormObjects
2476       \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2477       {
2478         \pdfmanagement_add:nnx { Catalog / AcroForm } { Fields }{##1}
2479         %\pdfmanagement_show:n { Catalog / AcroForm }
2480       }
2481       \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2482       {
2483         \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2484         {
2485           \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2486         }
2487         \seq_sort:Nn \l__hyp_tmpa_seq
2488         {
2489           \int_compare:nNnTF { \pdf@strcmp{##1}{##2} } > { 0 }
2490           %\str_compare:nNnTF {##1} > {##2}
2491           { \sort_return_swapped: }
2492           { \sort_return_same: }
2493         }
2494         \seq_map_inline:Nn \l__hyp_tmpa_seq
2495         {
2496           \pdfmanagement_add:nnx { Catalog / AcroForm }
2497           { CO }
2498           {
2499             \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2500           }
2501         }
2502       }
2503       \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2504       {ZaDb} {\pdf_object_ref:n {\__hyp/Font/ZaDb} }
2505       \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2506       {Helv} {\pdf_object_ref:n {\__hyp/Font/Helv} }
2507       \pdfmanagement_add:nnx {Catalog /AcroForm}
2508       {DA}{(/Helv~10~Tf~0~g)}
2509       \pdfmeta_standard_verify:nTF {form_no_NeedAppearances}
2510       {
2511         \legacy_if:nT { HyField@NeedAppearances }
2512         {
2513           \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2514         }
2515       }
2516       {
2517         \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2518       }
2519     }
2520     \MakeFieldObject
2521     {
2522       \group_begin:
2523       \fontfamily{pzd}

```

```

2524         \fontencoding{U}
2525         \fontseries{m}
2526         \fontshape{n}
2527         \selectfont
2528         \char123
2529         \group_end:
2530     }
2531     {__hyp_xform_Ding}
2532 \MakeFieldObject
2533 {
2534     \group_begin:
2535     \fontfamily{pzd}
2536     \fontencoding{U}
2537     \fontseries{m}
2538     \fontshape{n}
2539     \selectfont
2540     \phantom{\char123}
2541     \group_end:
2542 }
2543 {__hyp_xform_DingOff}
2544 \MakeFieldObject
2545 {
2546     \group_begin:
2547     \fontfamily{pzd}
2548     \fontencoding{U}
2549     \fontseries{m}
2550     \fontshape{n}
2551     \selectfont
2552     \char51
2553     \group_end:
2554 }
2555 {__hyp_xform_CheckMarkYes}
2556 \MakeFieldObject
2557 {
2558     \group_begin:
2559     \fontfamily{pzd}
2560     \fontencoding{U}
2561     \fontseries{m}
2562     \fontshape{n}
2563     \selectfont
2564     \phantom{\char51} %perhaps xetex needs some small glyph ..
2565     \group_end:
2566 }
2567 {__hyp_xform_CheckMarkOff}
2568 \MakeFieldObject
2569 {
2570     \fbox{\textcolor{yellow}{\textsf{Submit}}}% %color?
2571 }
2572 {__hyp_xform_Submit}
2573 \MakeFieldObject
2574 {
2575     \fbox{\textcolor{yellow}{\textsf{SubmitP}}}% %color?
2576 }
2577 {__hyp_xform_SubmitP}

```

```

2578 }
2579 \ExplSyntaxOff
2580 \let\@endForm\ltx@empty
2581 \let\HyAnn@AbsPageLabel\ltx@empty
2582 \let\Fld@pageobjref\ltx@empty
2583
2584 \ExplSyntaxOn
2585 \newcount\HyAnn@Count
2586 \HyAnn@Count=\ltx@zero
2587 \def\HyAnn@AbsPageLabel
2588 {
2589   \global\advance\HyAnn@Count by\ltx@one
2590   %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2591   %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2592   %\zref@refused{HyAnn@\the\HyAnn@Count}%
2593   \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
2594   \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
2595 }%
2596 \def\Fld@pageobjref
2597 {
2598   \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
2599   {
2600     /P~\pdf_pageobject_ref:n
2601     {
2602       \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
2603     }
2604   }
2605 }
2606 \ExplSyntaxOff
2607 \ExplSyntaxOn
2608 %% check if the attr should be set through
2609 %% hooks.
2610 %% check if options are missing.
2611 \def\@TextField[#1]#2{% parameters, label
2612   \def\Fld@name{#2}%
2613   \let\Fld@default\ltx@empty
2614   \let\Fld@value\@empty
2615   \def\Fld@width{\DefaultWidthofText}%
2616   \def\Fld@height{%
2617     \ifFld@multiline
2618       \DefaultHeightofTextMultiline
2619     \else
2620       \DefaultHeightofText
2621     \fi
2622   }%
2623   \begingroup
2624     \expandafter\HyField@SetKeys\expandafter{%
2625       \DefaultOptionsofText,#1%
2626     }%
2627     \PDFForm@Name
2628     \HyField@FlagsText
2629     \ifFld@hidden\def\Fld@width{1sp}\fi
2630     \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2631     \LayoutTextField{#2}{%

```

```

2632     \leavevmode
2633     \HyAnn@AbsPageLabel
2634     \Hy@escapeform\PDFForm@Text
2635     \pdfannot_box:nnnn
2636         {\Fld@width}
2637         {\Fld@height}
2638         {Opt} %is this correct?
2639         {\PDFForm@Text}
2640     \MakeTextField{\Fld@width}{\Fld@height}
2641     \HyField@AddToFields
2642 }%
2643 \endgroup
2644 }
2645 \providecommand\@curropt{}
2646 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2647     \def\Fld@name{#2}
2648     \let\Fld@default\relax
2649     \let\Fld@value\relax
2650     \def\Fld@width{\DefaultWidthofChoiceMenu}
2651     \def\Fld@height{\DefaultHeightofChoiceMenu}
2652     \begingroup
2653         \Fld@menulength=0 %
2654         \@tempdima\z@
2655         \clist_map_variable:nNn { #3 } \@curropt
2656         %\@for\@curropt:=#3\do
2657         {%
2658             \expandafter\Fld@checkequals\@curropt==\\%
2659             \Hy@StepCount\Fld@menulength
2660             \settowidth{\@tempdimb}{\@currDisplay}%
2661             \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2662         }%
2663     \advance\@tempdima by~15\p@
2664     \begingroup
2665         \HyField@SetKeys{#1}
2666     \edef\x{\endgroup
2667         \noexpand\expandafter
2668         \noexpand\HyField@SetKeys
2669         \noexpand\expandafter{%
2670             \expandafter\noexpand\csname DefaultOptionsof%
2671             \ifFld@radio
2672                 Radio%
2673             \else
2674                 \ifFld@combo
2675                 \ifFld@popdown
2676                     PopdownBox%
2677                 \else
2678                     ComboBox%
2679                 \fi
2680             \else
2681                 ListBox%
2682             \fi
2683             \fi
2684         \endcsname
2685     }%

```

```

2686 } \x
2687 \HyField@SetKeys{#1}%
2688 \PDFForm@Name
2689 \ifFld@hidden\def\Fld@width{1sp}\fi
2690 \ifx\Fld@value\relax
2691   \let\Fld@value\Fld@default
2692 \fi
2693 \LayoutChoiceField{#2}{%
2694   \ifFld@radio
2695     \HyField@FlagsRadioButton
2696     \__hypRadio{#3}%
2697   \else
2698     \beginngroup
2699     \HyField@FlagsChoice
2700     \ifdim\Fld@width<\@tempdima
2701       \ifdim\@tempdima<1cm\@tempdima1cm\fi
2702       \edef\Fld@width{\the\@tempdima}%
2703     \fi
2704     \ifFld@combo
2705     \else
2706       \@tempdima=\the\Fld@menulength\Fld@charsize
2707       \advance\@tempdima by~\Fld@borderwidth bp %
2708       \advance\@tempdima by~\Fld@borderwidth bp %
2709       \edef\Fld@height{\the\@tempdima}%
2710     \fi
2711     \__hypListBox{#3}%
2712   \endgroup
2713 \fi
2714 }%
2715 \endgroup
2716 }
2717 \tl_new:N \l__hyp_RadioYes_tl
2718 \tl_set:Nn \l__hyp_RadioYes_tl { __hyp_xform_Ding }
2719 \def\__hypRadio#1{%
2720   \Fld@listcount=0~%
2721   %\show\Fld@default
2722   \EdefEscapeName\Fld@default{\Fld@default}%
2723   \clist_map_variable:nNn { #1 } \@curropt
2724   %\@for\@curropt:=#1\do
2725   {%
2726     \expandafter\Fld@checkequals\@curropt==\%
2727     \EdefEscapeName\@currValue{\@currValue}%
2728     \Hy@StepCount\Fld@listcount
2729     \@currDisplay\space
2730     \leavevmode
2731     \HyAnn@AbsPageLabel
2732     \Hy@escapeform\PDFForm@Radio
2733     \pdfannot_box:nnnn
2734       {\Fld@width}
2735       {\Fld@height}
2736       {0pt} %is this correct?
2737     {
2738       \PDFForm@Radio
2739       /AP

```

```

2740         <<
2741         /N
2742         <<
2743         /@currValue\c_space_tl \pdfxform_ref:o {__hyp_xform_Ding}
2744         %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2745         >>
2746         >>
2747     }
2748     {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2749     \int_compare:nNt { \Fld@listcount } = { 1 }
2750     { \HyField@AddToFields }
2751     \c_space_tl % deliberate space between radio buttons
2752                 % to do: --> should be configurable
2753     }%
2754 }
2755
2756 \newcount\Fld@listcount
2757 \def\__hypListbox#1
2758 {
2759     \HyField@PDFChoices{#1}
2760     \mode_leave_vertical:
2761     \HyAnn@AbsPageLabel
2762     \Hy@escapeform\PDFForm@List
2763     \pdf_link_user:nnn
2764         {widget} %perhaps we need more types??
2765         {\PDFForm@List}
2766         {\MakeChoiceField{\Fld@width}{\Fld@height}}
2767     \HyField@AddToFields
2768 }
2769
2770
2771 \def\@PushButton[#1]#2{% parameters, label
2772     \def\Fld@name{#2}%
2773     \group_begin:
2774         \exp_args:No\HyField@SetKeys
2775         {
2776             \DefaultOptionsofPushButton,#1
2777         }
2778         \PDFForm@Name
2779         \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2780         {
2781             \HyField@FlagsPushButton
2782             \legacy_if:nT {Fld@hidden}
2783             {
2784                 \def\Fld@width{1sp}
2785             }
2786             \LayoutPushButtonField
2787             {
2788                 \mode_leave_vertical:
2789                 \HyAnn@AbsPageLabel
2790                 \Hy@escapeform\PDFForm@Push
2791                 \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2792                 \pdfannot_box:nnnn
2793                     {\box_wd:N\l_tmpa_box}

```

```

2794         {\box_ht:N\l_tmpa_box}
2795         {\box_dp:N\l_tmpa_box} %is this correct?
2796         {\PDFForm@Push}
2797         {\box_use:N\l_tmpa_box}
2798         \HyField@AddToFields
2799     }
2800 }
2801 {
2802     \msg_error:nn { hyp }{ pdfa-no-push-button }
2803     \LayoutPushButtonField
2804     {
2805         \mode_leave_vertical:
2806         \MakeButtonField{#2}
2807     }
2808 }
2809 \group_end:
2810 }
2811
2812 \def\@Submit[#1]#2
2813 {
2814     \def\Fld@width {\DefaultWidthofSubmit}
2815     \def\Fld@height{\DefaultHeightofSubmit}
2816     \group_begin:
2817         \exp_args:No\HyField@SetKeys
2818         {
2819             \DefaultOptionsofSubmit,#1
2820         }
2821     \HyField@FlagsPushButton
2822     \HyField@FlagsSubmit
2823     \legacy_if:nT { Fld@hidden }
2824     {
2825         \def\Fld@width{1sp}
2826     }
2827     \mode_leave_vertical:
2828     \HyAnn@AbsPageLabel
2829     \Hy@escapeform\PDFForm@Submit
2830     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2831     \pdfannot_box:nnnn
2832     {\box_wd:N\l_tmpa_box}
2833     {\box_ht:N\l_tmpa_box}
2834     {\box_dp:N\l_tmpa_box} %is this correct?
2835     {
2836         \PDFForm@Submit
2837         /AP<<
2838             /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2839             /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2840         >>
2841     }
2842     \HyField@AddToFields
2843     \box_use:N\l_tmpa_box
2844
2845     \group_end:
2846 }
2847

```

```

2848 \def\@Reset[#1]#2
2849 {
2850   \def\Fld@width {\DefaultWidthofReset}
2851   \def\Fld@height{\DefaultHeightofReset}
2852   \group_begin:
2853     \exp_args:No\HyField@SetKeys
2854     {
2855       \DefaultOptionsofReset,#1
2856     }
2857     \mode_leave_vertical:
2858     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2859     {
2860       \HyField@FlagsPushButton
2861       \legacy_if:nT { Fld@hidden }
2862       { \def\Fld@width{1sp} }
2863       \HyAnn@AbsPageLabel
2864       \Hy@escapeform\PDFForm@Reset
2865       \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2866       \pdfannot_box:nnnn
2867       {\box_wd:N\l_tmpa_box}
2868       {\box_ht:N\l_tmpa_box}
2869       {\box_dp:N\l_tmpa_box} %is this correct?
2870       { \PDFForm@Reset }
2871       \HyField@AddToFields
2872       \box_use:N \l_tmpa_box
2873     }
2874     {
2875       \msg_error:nn { hyp }{ pdfa-no-reset-button }
2876       \MakeButtonField{#2}
2877     }
2878   \group_end:
2879 }
2880
2881 \def\@CheckBox[#1]#2
2882 {% parameters, label
2883   \def\Fld@name{#2}
2884   \def\Fld@default{0}
2885   \group_begin:
2886     \def\Fld@width {\DefaultWidthofCheckBox}
2887     \def\Fld@height{\DefaultHeightofCheckBox}
2888     \exp_args:No\HyField@SetKeys
2889     {
2890       \DefaultOptionsofCheckBox,#1
2891     }
2892     \PDFForm@Name
2893     \HyField@FlagsCheckBox
2894     \legacy_if:nT { Fld@hidden }
2895     {
2896       \def\Fld@width{1sp}
2897     }
2898     \LayoutCheckField{#2}
2899     {
2900       \mode_leave_vertical:
2901       \HyAnn@AbsPageLabel

```

```

2902         \Hy@escapeform\PDFForm@Check
2903         \pdfannot_box:nnnn
2904         {\Fld@width}
2905         {\Fld@height}
2906         {Opt} %is this correct?
2907         {\PDFForm@Check}
2908         \HyField@AddToFields %check if this works with xelatex ...
2909     }
2910 \group_end:
2911 }
2912 \ExplSyntaxOff
2913
2914 %hm. Should a luatex driver use type1 fonts in fields????
2915 \ExplSyntaxOn
2916 \def\Hy@FormObjects
2917 {
2918     \pdf_object_new:nn {__hyp/Encoding/pdfdoc } { dict }
2919     \pdf_object_new:nn {__hyp/Font/ZaDb } { dict }
2920     \pdf_object_new:nn {__hyp/Font/Helv } { dict }
2921     \pdf_object_write:nx {__hyp/Encoding/pdfdoc }
2922     {
2923         /Type/Encoding
2924         /Differences[
2925             24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2926             /ring/tilde
2927             \c_space_tl
2928             39/quotesingle
2929             \c_space_tl
2930             96/grave %
2931             \iow_newline:
2932             128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2933             /fraction/guilsinglleft/guilsinglright/minus/perthousand
2934             /quotedblbase/quotedblleft/quotedblright/quoteleft
2935             /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2936             /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2937             \iow_newline:
2938             164/currency
2939             \c_space_tl
2940             166/brokenbar
2941             \c_space_tl
2942             168/dieresis/copyright/ordfeminine
2943             \c_space_tl
2944             172/logicalnot/.notdef/registered/macron/degree/plusminus
2945             /twosuperior/threesuperior/acute/mu
2946             \c_space_tl
2947             183/periodcentered/cedilla/onesuperior/ordmasculine
2948             \c_space_tl
2949             188/onequarter/onehalf/threequarters
2950             \iow_newline:
2951             192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
2952             /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
2953             /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
2954             /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
2955             /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls

```

```

2956         /grave/aacute/acircumflex/atilde/adieresis/aring/ae
2957         /cedilla/egrave/eacute/ecircumflex/edieresis/igrave
2958         /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
2959         /ocircumflex/otilde/odieresis/divide/oslash/ugrave
2960         /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
2961     ]
2962 }
2963 \pdf_object_write:nn {__hyp/Font/ZaDb }
2964 {
2965     /Type/Font
2966     /Subtype/Type1
2967     /Name/ZaDb
2968     /BaseFont/ZapfDingbats
2969 }
2970 \pdf_object_write:nx {__hyp/Font/Helv }
2971 {
2972     /Type/Font
2973     /Subtype/Type1
2974     /Name/Helv
2975     /BaseFont/Helvetica
2976     /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
2977 }
2978 \global\let\Hy@FormObjects\relax
2979 }
2980 \ExplSyntaxOff
2981 \providecommand*{\Fld@pageobjref}{\}
2982 \ifcsname pdf@escapestring\endcsname
2983   \def\Hy@escapeform#1{%
2984     \ifHy@pdfescapeform
2985       \let\Hy@escapestring\pdfescapestring
2986     \else
2987       \let\Hy@escapestring\@firstofone
2988     \fi
2989   }%
2990   \Hy@escapeform{}\}
2991 \else
2992   \let\Hy@escapestring\@firstofone
2993   \def\Hy@escapeform#1{%
2994     \ifHy@pdfescapeform
2995       \def\Hy@escapestring##1{%
2996         \noexpand\Hy@escapestring{\noexpand##1}%
2997       }%
2998       \edef\Hy@temp{#1}%
2999       \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{\}\@nil
3000     \def\Hy@escapestring##1{%
3001       \@ifundefined{Hy@esc@\string##1}{%
3002         ##1%
3003         \ThisShouldNotHappen
3004       }{%
3005         \csname Hy@esc@\string##1\endcsname
3006       }%
3007     }%
3008   \else
3009     \let\Hy@escapestring\@firstofone

```

```

3010     \fi
3011 }%
3012 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3013     \ifx\#3\%
3014     \else
3015     \expandafter
3016     \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3017     \ltx@ReturnAfterFi{%
3018     \Hy__hypescapeform#3\@nil
3019     }%
3020 \fi
3021 }%
3022 \fi
3023 \def\PDFForm@Name{%
3024     \PDFForm__hypName\Fld@name
3025     \ifx\Fld@altname\relax
3026     \else
3027     \PDFForm__hypName\Fld@altname
3028     \fi
3029     \ifx\Fld@mappingname\relax
3030     \else
3031     \PDFForm__hypName\Fld@mappingname
3032     \fi
3033 }
3034 \def\PDFForm__hypName#1{%
3035     \begingroup
3036     \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3037     \ifHy@unicode
3038     \Hy@unicodedefalse
3039     \fi
3040     \fi
3041     \pdfstringdef\Hy@gtemp#1%
3042     \endgroup
3043     \let#1\Hy@gtemp
3044 }
3045 \def\Fld@X@additionalactions{%
3046     \ifx\Fld@keystroke@code\@empty
3047     \else
3048     /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3049     \fi
3050     \ifx\Fld@format@code\@empty
3051     \else
3052     /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3053     \fi
3054     \ifx\Fld@validate@code\@empty
3055     \else
3056     /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3057     \fi
3058     \ifx\Fld@calculate@code\@empty
3059     \else
3060     /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3061     \fi
3062     \ifx\Fld@onfocus@code\@empty
3063     \else

```

```

3064     /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3065 \fi
3066 \ifx\Fld@onblur@code\@empty
3067 \else
3068     /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3069 \fi
3070 \ifx\Fld@onmousedown@code\@empty
3071 \else
3072     /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3073 \fi
3074 \ifx\Fld@onmouseup@code\@empty
3075 \else
3076     /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3077 \fi
3078 \ifx\Fld@onenter@code\@empty
3079 \else
3080     /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3081 \fi
3082 \ifx\Fld@onexit@code\@empty
3083 \else
3084     /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3085 \fi
3086 }
3087 \ExplSyntaxOn
3088 \def\Fld@additionalactions
3089 {%
3090     \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{}
3091     {
3092         \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3093         {/AA<<\Fld@X@additionalactions>>}
3094     }
3095 }
3096 \ExplSyntaxOff
3097 \def\Fld@annotnames{%
3098     /T(\Fld@name)%
3099     \ifx\Fld@altname\relax
3100     \else
3101         /TU(\Fld@altname)%
3102     \fi
3103     \ifx\Fld@mappingname\relax
3104     \else
3105         /TM(\Fld@mappingname)%
3106     \fi
3107 }
3108 \ExplSyntaxOn
3109 \def\PDFForm@Check
3110 {
3111     /Subtype/Widget
3112     ~\Fld@annotflags
3113     ~\Fld@pageobjref
3114     ~\Fld@annotnames
3115     /FT/Btn
3116     \Fld@flags
3117     /Q~\Fld@align

```

```

3118 /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3119 /AP
3120 <<
3121 /N
3122 <<
3123 /Yes~\pdfxform_ref:o{\l__hyp_CheckmarkYes_t1}
3124 /Off~\pdfxform_ref:o{\l__hyp_CheckmarkOff_t1}
3125 >>
3126 >>
3127 /MK<<
3128 \int_compare:nNf {\Fld@rotation}={0}
3129 {
3130 /R~\Fld@rotation
3131 }
3132 \tl_if_empty:NF\Fld@bordercolor
3133 {
3134 /BC[\Fld@bordercolor]
3135 }
3136 \tl_if_empty:NF\Fld@bcolor
3137 {
3138 /BG[\Fld@bcolor]
3139 }
3140 /CA(\Hy@escapestring{\Fld@cbsymbol})%
3141 >>
3142 /DA
3143 (
3144 /ZaDb~\strip@pt\Fld@charsize\c_space_t1 Tf
3145 \tl_if_empty:NF \Fld@color
3146 {
3147 \c_space_t1 \Fld@color
3148 }
3149 )
3150 /H/P
3151 \legacy_if:nTF {\Fld@checked}
3152 {
3153 /V/Yes /AS/Yes
3154 }
3155 {
3156 /V/Off /AS/Off
3157 }
3158 \Fld@additionalactions
3159 }
3160 \ExplSyntaxOff
3161 \ExplSyntaxOn
3162 \def\PDFForm@Push
3163 {
3164 /Subtype/Widget
3165 ~\Fld@annotflags
3166 ~\Fld@pageobjref
3167 ~\Fld@annotnames
3168 /FT/Btn
3169 ~\Fld@flags
3170 /H/P
3171 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>

```

```

3172     \bool_if:nT
3173     {
3174         !\int_compare_p:nNn {\Fld@rotation} = {0}
3175         ||
3176         \tl_if_exist_p:N \Fld@bordercolor
3177     }
3178     {
3179         /MK
3180         <<
3181             \int_compare:nNnF {\Fld@rotation} = {0}
3182             {
3183                 /R~\Fld@rotation
3184             }
3185             \tl_if_exist:NT \Fld@bordercolor
3186             {
3187                 /BC[\Fld@bordercolor]
3188             }
3189         >>
3190     }
3191     /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3192     \Fld@additionalactions
3193 }
3194
3195 \ExplSyntaxOff
3196 \def\PDFForm@List{%
3197     /Subtype/Widget%
3198     \Fld@annotflags
3199     \Fld@pageobjref
3200     \Fld@annotnames
3201     /FT/Ch%
3202     \Fld@flags
3203     /Q \Fld@align
3204     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3205     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3206         \ifx\Fld@bordercolor\relax\else 1\fi
3207         \ifx\Fld@bcolor\relax \else 1\fi
3208         \space
3209     \else
3210         /MK<<%
3211         \ifnum\Fld@rotation=\z@
3212         \else
3213             /R \Fld@rotation
3214         \fi
3215         \ifx\Fld@bordercolor\relax
3216         \else
3217             /BC[\Fld@bordercolor]%
3218         \fi
3219         \ifx\Fld@bcolor\relax
3220         \else
3221             /BG[\Fld@bcolor]%
3222         \fi
3223         >>%
3224     \fi
3225     /DA(/Helv \strip@pt\Fld@charsize\space Tf%

```

```

3226     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3227 \Fld@choices
3228 \Fld@additionalactions
3229 }
3230 \ExplSyntaxOn
3231 \def\PDFForm@Radio
3232 {
3233   /Subtype/Widget
3234   ~\Fld@annotflags
3235   ~\Fld@pageobjref
3236   ~\Fld@annotnames
3237   /FT/Btn
3238   \Fld@flags
3239   /H/P
3240   /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3241   /MK<<
3242   \ifnum\Fld@rotation=\z@
3243   \else
3244     /R~\Fld@rotation
3245   \fi
3246   \ifx\Fld@bordercolor\relax
3247   \else
3248     /BC[\Fld@bordercolor]%
3249   \fi
3250   \ifx\Fld@bcolor\relax
3251   \else
3252     /BG[\Fld@bcolor]%
3253   \fi
3254   /CA(\Hy@escapestring{\Fld@radiosymbol})%
3255   >>
3256   /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3257     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3258   \ifx\Fld@default\@empty
3259     /V/Off%
3260     /DV/Off%
3261   \else
3262     /V/\Fld@default
3263     /DV/\Fld@default
3264   \fi
3265   \Fld@additionalactions
3266 }
3267 \ExplSyntaxOff
3268 \ExplSyntaxOn
3269 % Does an appeareance dict make sense here?
3270 \def\PDFForm@Text
3271 {
3272   /Subtype/Widget
3273   ~\Fld@annotflags
3274   ~\Fld@pageobjref
3275   ~\Fld@annotnames
3276   /FT/Tx
3277   ~\Fld@flags
3278   /Q~\Fld@align
3279   /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>

```

```

3280 \bool_if:nT
3281 {
3282     !\int_compare_p:nNn {\Fld@rotation} = {0}
3283     ||
3284     \tl_if_exist_p:N \Fld@bordercolor
3285     ||
3286     \tl_if_exist_p:N \Fld@bcolor
3287 }
3288 {
3289     /MK
3290     <<
3291     \int_compare:nNf {\Fld@rotation} = {0}
3292     {
3293         /R~\Fld@rotation
3294     }
3295     \tl_if_exist:NT \Fld@bordercolor
3296     {
3297         /BC[\Fld@bordercolor]
3298     }
3299     \tl_if_exist:NT \Fld@bcolor
3300     {
3301         /BG[\Fld@bcolor]
3302     }
3303     >>
3304 }
3305 /DA
3306 (
3307     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3308     \tl_if_empty:NF {\c_space_tl\Fld@color}
3309 )
3310 /DV(\Hy@escapestring{\Fld@default})
3311 /V(\Hy@escapestring{\Fld@value})
3312 ~\Fld@additionalactions
3313 \int_compare:nNt { \Fld@maxlen}>{0}
3314 {
3315     /MaxLen~\Fld@maxlen
3316 }
3317 }
3318 \ExplSyntaxOff
3319
3320 \def\PDFForm@Submit{%
3321     /Subtype/Widget%
3322     \Fld@annotflags
3323     \Fld@pageobjref
3324     \Fld@annotnames
3325     /FT/Btn%
3326     \Fld@flags
3327     /H/P%
3328     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3329     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3330         \ifx\Fld@bordercolor\relax\else 1\fi
3331         \space
3332     \else
3333         /MK<<%

```

```

3334     \ifnum\Fld@rotation=\z@
3335     \else
3336         /R \Fld@rotation
3337     \fi
3338     \ifx\Fld@bordercolor\relax
3339     \else
3340         /BC[\Fld@bordercolor]%
3341     \fi
3342     >>%
3343 \fi
3344 /A<<%
3345     /S/SubmitForm%
3346     /F<<%
3347         /FS/URL%
3348         /F(\Hy@escapestring{\Form@action})%
3349     >>%
3350     \Fld@submitflags
3351     >>%
3352     \Fld@additionalactions
3353 }
3354 \ExplSyntaxOn
3355     \def\PDFForm@Reset{%
3356         /Subtype/Widget%
3357         \Fld@annotflags
3358         \Fld@pageobjref
3359         \Fld@annotnames
3360         /FT/Btn%
3361         \Fld@flags
3362         /H/P%
3363         /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3364         \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3365             \ifx\Fld@bordercolor\relax\else 1\fi
3366             \space
3367         \else
3368             /MK<<%
3369                 \ifnum\Fld@rotation=\z@
3370                 \else
3371                     /R~\Fld@rotation
3372                 \fi
3373                 \ifx\Fld@bordercolor\relax
3374                 \else
3375                     /BC[\Fld@bordercolor]%
3376                 \fi
3377             >>%
3378         \fi
3379         /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3380         /A<</S/ResetForm>>%
3381         \Fld@additionalactions
3382     }%
3383
3384
3385 %these patterns are used in hyperref checks.
3386 %it is unclear if they are really useful and if a backend support is
3387 %needed.

```

```

3388 \str_case:VnF \c_sys_backend_str
3389 {
3390   { pdfmode }
3391   {
3392     \def\HyPat@ObjRef
3393     {
3394       [0-9]*[1-9][0-9]*~0~R
3395     }
3396   }
3397   { dvipdfmx }
3398   {
3399     \def\HyPat@ObjRef
3400     {
3401       @[\~]+
3402     }
3403   }
3404   { xdvipdfmx }
3405   {
3406     \def\HyPat@ObjRef
3407     {
3408       @[\~]+
3409     }
3410   }
3411 }
3412 { %also set in hyperref sty, so probably not needed.
3413   \def\HyPat@ObjRef/{.+}
3414 }
3415
3416
3417 \ExplSyntaxOff
3418 % UF: removed Hy@writebookmark
3419 %   \Hy@currentbookmarklevel{0}
3420 %   \Hy@numberline
3421 %   \__hypwritetorep
3422 %   counter{bookmark@seq@number}
3423 % removed \HyPsd@SanitizeForOutFile, not needed
3424 % removed \currentpdfbookmark, defined by bookmark,
3425 % should use \newcommand there
3426 % removed \subpdfbookmark, defined by bookmark,
3427 % should use \newcommand there
3428 % removed \belowpdfbookmark, defined by bookmark,
3429 % should use \newcommand there
3430 % removed \pdfbookmark, defined by bookmark,
3431 % \BOOKMARK
3432 % \@BOOKMARK
3433 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3434 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3435 %% removed \ReadBookmarks / unneeded with bookmark.
3436 %% removed \Hy@OutlineName
3437 %% removed \check@bm@number
3438 %% removed \calc@bm@number
3439
3440 \ifHy@implicit
3441 \else

```

```

3442 \expandafter\endinput
3443 \fi
3444 \newlength\Hy@SectionHShift
3445 \def\Hy@SectionAnchorHref#1{%
3446 \ifx\protect\@typeset@protect
3447 \Hy__hypSectionAnchor{#1}%
3448 \fi
3449 }
3450 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%
3451 \leavevmode
3452 \hbox to Opt{%
3453 \kern-\Hy@SectionHShift
3454 \Hy@raisedlink{%
3455 \hyper@anchorstart{#1}\hyper@anchorend
3456 }%
3457 \hss
3458 }%
3459 }
3460 \let\H@old@ssect\@ssect
3461 \def\@ssect#1#2#3#4#5{%
3462 \Hy@MakeCurrentHrefAuto{section*}%
3463 \setlength{\Hy@SectionHShift}{#1}%
3464 \begingroup
3465 \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3466 \toks\tw\expandafter{%
3467 \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3468 #5%
3469 }%
3470 \edef\x{\endgroup
3471 \the\toks@\the\toks\tw}%
3472 }\x
3473 }
3474 \let\H@old@schapter\@schapter
3475 \def\@schapter#1{%
3476 \begingroup
3477 \let\@mkboth\@gobbletwo
3478 \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3479 \Hy@raisedlink{%
3480 \hyper@anchorstart{\@currentHref}\hyper@anchorend
3481 }%
3482 \endgroup
3483 \H@old@schapter{#1}%
3484 }
3485 \ltx@ifundefined{chapter}{-}{%
3486 \let\Hy@org@chapter\@chapter
3487 \def\@chapter{%
3488 \def\Hy@next{%
3489 \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3490 \Hy@raisedlink{%
3491 \hyper@anchorstart{\@currentHref}\hyper@anchorend
3492 }%
3493 }%
3494 \ifnum\c@secnumdepth>\m@ne
3495 \ltx@ifundefined{if@mainmatter}%

```

```

3496     \iftrue{\csname if@mainmatter\endcsname}%
3497     \let\Hy@next\relax
3498     \fi
3499     \fi
3500     \Hy@next
3501     \Hy@org@chapter
3502 }%
3503 }
3504 \let\H@old@part\@part
3505 \begingroup\expandafter\expandafter\expandafter\endgroup
3506 \expandafter\ifx\csname chapter\endcsname\relax
3507     \let\Hy@secnum@part\z@
3508 \else
3509     \let\Hy@secnum@part\m@ne
3510 \fi
3511 \def\@part{%
3512     \ifnum\Hy@secnum@part>\c@secnumdepth
3513         \phantomsection
3514     \fi
3515     \H@old@part
3516 }
3517 \let\H@old@spart\@spart
3518 \def\@spart#1{%
3519     \Hy@MakeCurrentHrefAuto{part*}%
3520     \Hy@raisedlink{%
3521         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3522     }%
3523     \H@old@spart{#1}%
3524 }
3525 \let\H@old@sect\@sect
3526 \def\@sect#1#2#3#4#5#6[#7]#8{%
3527     \ifnum #2>\c@secnumdepth
3528         \expandafter\@firstoftwo
3529     \else
3530         \expandafter\@secondoftwo
3531     \fi
3532     {%
3533         \Hy@MakeCurrentHrefAuto{section*}%
3534         \setlength{\Hy@SectionHShift}{#3}%
3535         \begingroup
3536         \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3537         \toks\tw@\expandafter{%
3538             \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3539             #8%
3540         }%
3541         \edef\x{\endgroup
3542             \the\toks@{\the\toks\tw@}}%
3543         }\x
3544     }{%
3545         \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3546     }%
3547 }
3548 \expandafter\def\csname Parent-4\endcsname{}
3549 \expandafter\def\csname Parent-3\endcsname{}

```

```

3550 \expandafter\def\csname Parent-2\endcsname{}
3551 \expandafter\def\csname Parent-1\endcsname{}
3552 \expandafter\def\csname Parent0\endcsname{}
3553 \expandafter\def\csname Parent1\endcsname{}
3554 \expandafter\def\csname Parent2\endcsname{}
3555 \expandafter\def\csname Parent3\endcsname{}
3556 \expandafter\def\csname Parent4\endcsname{}
3557 %%
3558 %% End of file 'hgeneric-testphase.def'.
3559 \package

3560 (*colorscheme)
3561 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3562 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3563 % linkcolor=
3564 %,filecolor=
3565 %,urlcolor=
3566 %,menucolor=
3567 %,runcolor=
3568 %,linkbordercolor=
3569 %,filebordercolor=
3570 %,urlbordercolor=
3571 %,menubordercolor=
3572 %,runbordercolor=
3573
3574 \prop_const_from_keyval:Nn \c__hyp_colorscheme_original_prop
3575 {
3576   linkcolor      = [rgb]{1,0,0}, %red
3577   filecolor      = [rgb]{0,1,1}, %cyan
3578   urlcolor       = [rgb]{1,0,1}, %magenta
3579   menucolor      = [rgb]{1, 0, 0}, %red
3580   runcolor       = [rgb]{0,1,1}, %cyan
3581   %-----
3582   linkbordercolor = [rgb]{1, 0 ,0 },
3583   filebordercolor = [rgb]{0, .5, .5},
3584   urlbordercolor  = [rgb]{0, 1, 1},
3585   menubordercolor = [rgb]{1, 0, 0},
3586   runbordercolor  = [rgb]{0, .7, .7}
3587 }
3588
3589 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3590 {
3591   linkcolor      = [rgb]{0,0.2,0.6},
3592   filecolor      = [rgb]{0.8,0,0.8},
3593   urlcolor       = [rgb]{0.8,0,0.8},
3594   menucolor      = [rgb]{0,0.2,0.6},
3595   runcolor       = [rgb]{0.8,0,0.8},
3596   %----- %-----
3597   linkbordercolor = [rgb]{0,0.2,0.6},
3598   filebordercolor = [rgb]{0.8,0,0.8},
3599   urlbordercolor  = [rgb]{0.8,0,0.8},
3600   menubordercolor = [rgb]{0,0.2,0.6},
3601   runbordercolor  = [rgb]{0.8,0,0.8}
3602 }
3603

```

```

3604 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3605 { %two colors: intern/extern
3606   linkcolor      = [rgb]{0.79216, 0, 0.12549},
3607   filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3608   urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3609   menucolor      = [rgb]{0.79216, 0, 0.12549 },
3610   runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3611   %----- %-----
3612   linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3613   filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3614   urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3615   menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3616   runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3617 }
3618
3619 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3620 { %all darkgray
3621   linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3622   filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3623   urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3624   menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3625   runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3626   %----- %-----
3627   linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3628   filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3629   urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3630   menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3631   runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3632 }
3633
3634 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3635 { %dvipsnam.def
3636   linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3637   filecolor      = [rgb]{1, 0, 0}, %Red
3638   urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3639   menucolor      = [rgb]{1, 0, 0}, %Red
3640   runcolor       = [rgb]{1, 0, 0}, %Red
3641   %----- %-----
3642   linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3643   filebordercolor = [rgb]{1, 0, 0}, %Red
3644   urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3645   menubordercolor = [rgb]{1, 0, 0}, %Red
3646   runbordercolor  = [rgb]{1, 0, 0} %Red
3647 }
3648
3649 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3650 { %dvipsnam.def
3651   linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3652   filecolor      = [rgb]{0, 1, 0}, %Green
3653   urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3654   menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3655   runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3656   %----- %-----
3657   linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed

```

```

3658     filebordercolor = [rgb]{0, 1, 0},      %Green
3659     urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3660     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3661     runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3662 }
3663
3664
3665 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3666 {
3667     linkcolor      = [rgb]{0.50196, 0, 0.02353},
3668     filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3669     urlcolor       = [rgb]{0.54118, 0, 0.52941},
3670     menucolor      = [rgb]{0.44706, 0.45882, 0},
3671     runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3672     %----- %-----
3673     linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3674     filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3675     urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3676     menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3677     runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3678 }
3679
3680 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3681 {
3682     linkcolor      = [rgb]{0,0,0},
3683     filecolor      = [rgb]{0,0,0},
3684     urlcolor       = [rgb]{0,0,0},
3685     menucolor      = [rgb]{0,0,0},
3686     runcolor       = [rgb]{0,0,0},
3687     %----- %-----
3688     linkbordercolor = [rgb]{0,0,0},
3689     filebordercolor = [rgb]{0,0,0},
3690     urlbordercolor  = [rgb]{0,0,0},
3691     menubordercolor = [rgb]{0,0,0},
3692     runbordercolor  = [rgb]{0,0,0}
3693 }
3694 </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	
\#	250, 671
\\$	249
\%	672
\.	489, 493, 495
@curropt commands:	
\@curropt:	2656, 2724
\\	17, 18, 19, 26, 30, 40, 60, 67, 74, 81, 88, 103, 108, 109, 117, 118, 127, 128, 129, 130, 131, 138, 251, 252, 801, 2658, 2726, 3013
_	487, 489, 493, 495
A	
\A	487
\Acrobatmenu	17, 151
\addcontentsline	12
\AddToDocumentProperties	363
\AddToHook	388
\advance	2589, 2663, 2707, 2708
allcolor (hypersetup key)	930
\author	2
B	
\b	491
\begingroup	179, 248, 350, 2431, 2623, 2652, 2664, 2698, 3035, 3464, 3476, 3505, 3535
\belowpdfbookmark	3428
\bgroup	245, 350
\BOOKMARK	3431
bookmarkstype (hypersetup key)	12
bool commands:	
\bool_if:N	238, 268, 330, 620, 635, 646, 659, 667, 697, 755, 812, 900, 910, 1150, 1161
\bool_if:nTF	767, 1857, 1882, 3172, 3280
\bool_lazy_and:nnTF	383
\bool_lazy_or:nnTF	1180, 1229
\bool_new:N	173, 174, 473, 477, 481
\bool_set_true:N	482
bordercolormodel (hypersetup key)	12, 951
box commands:	
\box_dp:N	563, 2795, 2834, 2869
\box_ht:N	562, 2794, 2833, 2868
\box_new:N	422, 484
\box_set_dp:Nn	2054
\box_set_ht:Nn	2061
\box_use:N	1167, 2797, 2843, 2872
\box_use_drop:N	1172
\box_wd:N	561, 2793, 2832, 2867
\l_tmpa_box	2791, 2793, 2794, 2795, 2797, 2830, 2832, 2833, 2834, 2843, 2865, 2867, 2868, 2869, 2872
C	
\catcode	249, 250
\char	2528, 2540, 2552, 2564
\chardef	136
\cite	30
clist commands:	
\clist_map_function:nN	110, 119
\clist_map_inline:nn	2218
\clist_map_variable:nNn	2655, 2723
color commands:	
\color_export:nnN	850
\color_export:nnnN	856
\color_select:n	867, 903, 1171
\color_select:nn	873
\color_set:nn	2, 368, 884
\color_set:nnn	2, 367, 890
color names:	
hyp/annot/file	470
hyp/annot/link	470
hyp/annot/menu	470
hyp/annot/run	470
hyp/annot/url	470
colorfile (hypersetup key)	930
colorlink (hypersetup key)	930
colorlinks (hypersetup key)	916
colormenu (hypersetup key)	930
colorrund (hypersetup key)	930
colorscheme (hypersetup key)	1, 1338
colorurl (hypersetup key)	930
cs commands:	
\cs_generate_variant:Nn	134, 135, 365, 401, 416, 421, 525, 859, 876, 893
\cs_gset:Npn	1142
\cs_if_exist_p:N	384
\cs_new:Npn	103, 417
\cs_new_protected:Npn	179, 277, 295, 313, 340, 360, 395, 404, 501, 505, 509, 513, 526, 528, 532, 574, 580, 587, 591, 613, 618, 633, 644, 657, 695, 753, 810, 843, 854, 860, 871, 877, 888, 1086, 2089, 2145

<code>\cs_set_eq:NN</code>	139, 144, 145, 185, 227, 671, 672, 1415
<code>\cs_set_protected:Npn</code>	137, 158, 542, 801
<code>\csname</code>	2670, 3005, 3016, 3496, 3506, 3548, 3549, 3550, 3551, 3552, 3553, 3554, 3555, 3556
<code>\currentpdfbookmark</code>	3424
D	
<code>\d</code>	489, 493, 495
<code>debug</code> (hypersetup key)	1362
<code>\DeclareDocumentMetadata</code>	3, 10, 13, 14, 17, 19, 27, 33
<code>\DeclareRobustCommand</code>	232, 261, 285, 303, 323, 353, 3450
<code>\def</code>	143, 350, 351, 353, 537, 2394, 2398, 2405, 2423, 2452, 2469, 2587, 2596, 2611, 2612, 2615, 2616, 2629, 2630, 2646, 2647, 2650, 2651, 2689, 2719, 2757, 2771, 2772, 2784, 2812, 2814, 2815, 2825, 2848, 2850, 2851, 2862, 2881, 2883, 2884, 2886, 2887, 2896, 2916, 2983, 2993, 2995, 3000, 3012, 3023, 3034, 3045, 3088, 3097, 3109, 3162, 3196, 3231, 3270, 3320, 3355, 3392, 3399, 3406, 3413, 3445, 3461, 3475, 3487, 3488, 3511, 3518, 3526, 3548, 3549, 3550, 3551, 3552, 3553, 3554, 3555, 3556
<code>\DefaultHeightofCheckBox</code>	2887
<code>\DefaultHeightofChoiceMenu</code>	2651
<code>\DefaultHeightofReset</code>	2851
<code>\DefaultHeightofSubmit</code>	2815
<code>\DefaultHeightofText</code>	2620
<code>\DefaultHeightofTextMultiline</code> ...	2618
<code>\DefaultOptionsofCheckBox</code>	2890
<code>\DefaultOptionsofPushButton</code>	2776
<code>\DefaultOptionsofReset</code>	2855
<code>\DefaultOptionsofSubmit</code>	2819
<code>\DefaultOptionsofText</code>	2625
<code>\DefaultWidthofCheckBox</code>	2886
<code>\DefaultWidthofChoiceMenu</code>	2650
<code>\DefaultWidthofReset</code>	2850
<code>\DefaultWidthofSubmit</code>	2814
<code>\DefaultWidthofText</code>	2615
<code>destlabel</code> (hypersetup key)	12
dim commands:	
<code>\dim_to_decimal_in_bp:n</code>	11, 139
<code>\do</code>	2656, 2724
<code>\documentclass</code>	20
<code>draft</code> (hypersetup key)	1362
E	
<code>\edef</code> 2433, 2666, 2702, 2709, 2998, 3470, 3541	
<code>\EdefEscapeName</code>	2722, 2727
<code>\egroup</code>	353
<code>\else</code>	2430, 2619, 2673, 2677, 2680, 2697, 2705, 2986, 2991, 3008, 3014, 3026, 3030, 3047, 3051, 3055, 3059, 3063, 3067, 3071, 3075, 3079, 3083, 3100, 3104, 3205, 3206, 3207, 3209, 3212, 3216, 3220, 3226, 3243, 3247, 3251, 3257, 3261, 3329, 3330, 3332, 3335, 3339, 3364, 3365, 3367, 3370, 3374, 3441, 3508, 3529
<code>\endcsname</code>	2684, 2982, 3005, 3016, 3496, 3506, 3548, 3549, 3550, 3551, 3552, 3553, 3554, 3555, 3556
<code>\endgroup</code>	254, 257, 351, 353, 2434, 2643, 2666, 2712, 2715, 3042, 3470, 3482, 3505, 3541
<code>\endinput</code>	3442
exp commands:	
<code>\exp_args:Ne</code>	3090
<code>\exp_args:Nne</code>	2183
<code>\exp_args:Nnno</code>	297, 315
<code>\exp_args:Nno</code>	279, 342, 2131
<code>\exp_args:Nnox</code>	705
<code>\exp_args:NNV</code>	1842, 1873
<code>\exp_args:NnV</code>	576, 583
<code>\exp_args:Nnx</code>	2271, 2376
<code>\exp_args:No</code>	605, 609, 2774, 2817, 2853, 2888
<code>\exp_args:NV</code>	559, 566
<code>\exp_args:Nx</code> ...	701, 2124, 2172, 2425
<code>\exp_not:N</code>	801
<code>\exp_not:n</code>	1277, 1311, 1422, 1482, 1503, 1540, 1560, 1583, 1627, 1660, 1685, 1736, 1761, 1788, 1837, 1914, 1942, 1969, 1989, 2204, 2321, 2349, 2359, 2370
<code>\expandafter</code>	251, 2624, 2658, 2667, 2669, 2670, 2726, 2999, 3015, 3442, 3466, 3467, 3505, 3506, 3528, 3530, 3537, 3538, 3548, 3549, 3550, 3551, 3552, 3553, 3554, 3555, 3556
<code>\ExplSyntaxOff</code>	9, 2462, 2579, 2606, 2912, 2980, 3096, 3160, 3195, 3267, 3318, 3417
<code>\ExplSyntaxOn</code>	7, 10, 2463, 2584, 2607, 2915, 3087, 3108, 3161, 3230, 3268, 3354
<code>extension</code> (hypersetup key)	12, 1384
F	
<code>\fbox</code>	2570, 2575, 2748
<code>\fi</code>	2419, 2446, 2449, 2458, 2621, 2629, 2630,

2661, 2679, 2682, 2683, 2689, 2692, 2701, 2703, 2710, 2713, 2988, 3010, 3020, 3022, 3028, 3032, 3039, 3040, 3049, 3053, 3057, 3061, 3065, 3069, 3073, 3077, 3081, 3085, 3102, 3106, 3205, 3206, 3207, 3214, 3218, 3222, 3224, 3226, 3245, 3249, 3253, 3257, 3264, 3329, 3330, 3337, 3341, 3343, 3364, 3365, 3372, 3376, 3378, 3443, 3448, 3498, 3499, 3510, 3514, 3531	\hook_new:n 366, 500 \hook_new_pair:nn 230, 259, 284, 302, 321 \hook_use:n 234, 255, 264, 281, 288, 299, 306, 317, 326, 346, 517, 624, 639 \href 3–5, 17, 232 \hreflaunch 5 \hrefpdf 4, 5, 17, 285, 358 \hrefrun 4, 303, 359 \hrefurl 4, 5, 17, 261, 357 \hss 3457
file (hypersetup key) 9, 1430	Hy internal commands:
file commands:	\Hy__hypescapeform . . 2999, 3012, 3018 \Hy__hypSectionAnchor . . . 3447, 3450
\file_input:n 8	HyField internal commands:
fileborderstyle (hypersetup key) 13, 1040	\HyField__hypAddToFields . 2425, 2452
filecolor (hypersetup key) 930	hyp commands:
final (hypersetup key) 1362	\l_hyp_annot_colorfile_bool . . . 471 \l_hyp_annot_colorlink_bool . . . 471 \l_hyp_annot_colormenu_bool . . . 471 \l_hyp_annot_colorrun_bool . . . 471 \l_hyp_annot_colorurl_bool . . . 471 \l_hyp_annot_ocgcolorfile_bool . 475 \l_hyp_annot_ocgcolorlink_bool . 475 \l_hyp_annot_ocgcolormenu_bool . 475 \l_hyp_annot_ocgcolorrun_bool . . 475 \l_hyp_annot_ocgcolorurl_bool . . 475
\fontencoding 2524, 2536, 2548, 2560	hyp internal commands:
\fontfamily 2523, 2535, 2547, 2559	\g__hyp_AcroForm_CoFields_prop 2386, 2407, 2481, 2483, 2499 \g__hyp_AcroForm_Fields_prop 2387, 2400, 2476 \l__hyp_annot_GoTo_bool 620, 635, 646 \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 479 \l__hyp_annot_GoToR_bool 697 \l__hyp_annot_Launch_bool 755 \l__hyp_annot_Named_bool 812 \c__hyp_annot_types_seq 434, 471, 475, 930, 1227, 1325 \l__hyp_annot_URI_bool 659 \g__hyp_bordercolormodel_str 470, 955, 971 \l__hyp_CheckmarkOff_tl 2466, 2467, 3124 \l__hyp_CheckmarkYes_tl 2464, 2465, 3123 __hyp_clist_display:n . . 103, 110, 119 __hyp_color_export:nnN 36, 843, 843, 859, 971
\fontseries 2525, 2537, 2549, 2561	
\fontshape 2526, 2538, 2550, 2562	
fp commands:	
\fp_eval:n 2012, 2040, 2056	
G	
\gdef 251, 252	
\GetDocumentProperties 2	
\global 2589, 2978	
group commands:	
\group_begin: 235, 265, 289, 307, 327, 515, 623, 638, 661, 699, 757, 814, 902, 1153, 1170, 2300, 2522, 2534, 2546, 2558, 2773, 2816, 2852, 2885 \group_end: 280, 298, 316, 345, 522, 627, 649, 685, 745, 795, 838, 912, 1173, 1176, 2307, 2529, 2541, 2553, 2565, 2809, 2845, 2878, 2910	
H	
\hbox 350, 3452	
hbox commands:	
\hbox_overlap_right:n 1167 \hbox_set:Nn 2791, 2830, 2865 \hbox_set:Nw 1154 \hbox_set_end: 1163 \hbox_set_to_wd:Nnn 2038	
hidefile (hypersetup key) 1314	
hidelink (hypersetup key) 1314	
hidelinks (hypersetup key) 1314	
hidemenu (hypersetup key) 1314	
hiderun (hypersetup key) 1314	
hideurl (hypersetup key) 1314	
hook commands:	
\hook_gput_code:nnn 375, 896, 906, 1146, 1157	

__hyp_color_export_aux:wnnN . . .	\l__hyp_href_url_encode_bool . . .
. 843, 847, 854 173, 184, 222, 238, 268, 330
__hyp_color_select:n 37, 860, 860, 876	__hyp_href_url_format
__hyp_color_select:nN 37 179, 185, 227, 343
__hyp_color_select_aux:wn	\l__hyp_href_url_ismap_bool
. 860, 864, 871 174, 198, 667
__hyp_color_set:nN 37	\l__hyp_href_url_protocol_tl
__hyp_color_set:nn 175, 186, 225, 279, 344
. 37, 371, 373, 877, 877, 893, 935	__hyp_link_goto_begin:nw
__hyp_color_set_aux:nwn 877, 881, 888 591, 625, 640
\c__hyp_colorscheme_daleif_prop 3589	__hyp_link_goto_end: . 613, 626, 648
\c__hyp_colorscheme_henryford_-	\c__hyp_map_annot_hyp_prop 434
prop 3680	\c__hyp_map_hyp_annot_prop
\c__hyp_colorscheme_julian_prop 3604 434, 894,
\c__hyp_colorscheme_original_-	958, 994, 1021, 1029, 1040, 1067,
prop 3574	1075, 1144, 1252, 1288, 1298, 1430
\c__hyp_colorscheme_phelype_prop	__hyp_ocg_init: 1086, 1086, 1142, 1152
. 3665	l__hyp_page/Trans 498
\c__hyp_colorscheme_szabolcsA_-	__hyp_PageLabels_gpush: 532, 532, 545
prop 3634	\l__hyp_para_tmpa_tl 430, 766, 769, 778
\c__hyp_colorscheme_szabolcsB_-	\l__hyp_para_tmpa_tl\l__hyp_-
prop 3649	text_tmpa_str\g__hyp_text_-
\c__hyp_colorscheme_tivv_prop . 3619	tmpa_str 427
\l__hyp_dest_box 29,	\l__hyp_RadioYes_tl 2717, 2718
484, 561, 562, 563, 2038, 2054, 2061	__hyp_ref_check:nn 394, 404, 416, 2594
\l__hyp_dest_name_tmpa_tl . . 427,	__hyp_ref_if_exist:nn . 394, 402, 403
594, 595, 601, 605, 607, 610, 718, 731	__hyp_ref_if_exist:nnTF . . 406, 2598
\l__hyp_dest_pdfremotestartview_-	__hyp_ref_label:nn 394, 395, 401, 2593
tl 453, 726, 1845, 1849	__hyp_ref_value:nn 394, 417, 421, 2602
\g__hyp_dest_pdfstartpage_tl . . .	__hyp_secondoftwoewithopt:wnn . . .
. . . 453, 1856, 1858, 1865, 1883, 1890 356, 357, 358, 359
\g__hyp_dest_pdfstartview_tl . . .	__hyp_setup_info_date_key:nn . . .
453, 1858, 1865, 1876, 1880, 1883, 1890 2145, 2177, 2178
\l__hyp_dest_pdfview_tl	__hyp_setup_info_key:nn . . 2089,
. 469, 577, 584,	2138, 2139, 2140, 2141, 2143, 2144
2007, 2010, 2017, 2020, 2021, 2022,	__hyp_store_metadata:nn
2023, 2024, 2025, 2030, 2034, 2070 360, 365, 2086,
\c__hyp_dest_startview_regex . . .	2106, 2124, 2158, 2172, 2193, 2265
. 485, 1843, 1874	__hyp_text_cleanup:N . 505, 505, 519
\c__hyp_dest_undefined_tl	\l__hyp_text_enc_dest_print_tl . . .
. 433, 600, 601 456, 717
__hyp_destination:nn	\l__hyp_text_enc_dest_tl
. 29, 548, 548, 577, 584 29, 456, 555, 606
\l__hyp_filename_tmpa_tl . . . 427,	\l__hyp_text_enc_file_print_tl . . .
700, 702, 708, 709, 714, 761, 762, 778 456, 760
__hyp_href_pdf_aux:nn 292, 295	\l__hyp_text_enc_info_print_tl . . .
\l__hyp_href_pdf_destination_tl 456, 530
. 176, 187, 226, 297	\l__hyp_text_enc_para_print_tl . . .
\l__hyp_href_pdf_page_tl 177, 195, 725 456, 765
__hyp_href_run_aux:nn 310, 313	\l__hyp_text_enc_uri_print_tl . . .
\l__hyp_href_run_parameter_tl 240,
. 178, 199, 315	243, 270, 273, 332, 335, 456, 664, 1441
__hyp_href_url_aux:n 337, 340	
__hyp_href_url_aux:nn 275, 277	

_hyp_text_pdfstring:nnN	hidelinks	1314
..... 513, 513, 525, 527,	hidemenu	1314
530, 553, 604, 662, 715, 758, 763, 1441	hiderun	1314
_hyp_text_pdfstring_info:nn	hideurl	1314
..... 528, 528, 2100, 2118, 2128	hypertextnames	1384
_hyp_text_purify:nn . 501, 501, 518	link	9, 1430
_hyp_text_string_from_unicode:nn	linkborder	13
..... 509, 509, 520	linkborderstyle	13, 1040
\g_hyp_text_tmpa_str . 432, 521, 523	linkcolor	930
\l_hyp_text_tmpa_str	linkfileprefix	1384
..... 431, 518, 519, 520, 521	linktoc	1384
\l_hyp_tmpa_box 422, 1154, 1167, 1172	linktocpage	1384
\l_hyp_tmpa_int	localanchorname	1384
..... 422, 1843,	menu	9, 1430
1845, 1874, 1876, 1997, 1998, 2002,	menuborder	13
2004, 2028, 2035, 2036, 2037, 2046,	menuborderstyle	13, 1040
2048, 2058, 2063, 2485, 2487, 2494	menucolor	930
\l_hyp_tmpa_str	naturalnames	1384
..... 422, 2100, 2101, 2103,	nesting	13
2118, 2119, 2121, 2128, 2129, 2133	ocgcolorfile	1180
\l_hyp_tmpa_tl 422, 556, 560, 568,	ocgcolorlink	1180
971, 975, 1441, 1442, 1447, 1841,	ocgcolorlinks	1180
1843, 1872, 1874, 2004, 2005, 2012	ocgcolormenu	1180
\l_hyp_uri_tmpa_tl ... 427, 665, 666	ocgcolorrun	1180
hyp/annot/file (color name)	ocgcolorurl	1180
..... 470	pageanchor	1384
hyp/annot/link (color name)	pdfauthor	2089
..... 470	pdfborder	13
hyp/annot/menu (color name)	pdfborderstyle	13, 1040
..... 470	pdfcreationdate	13, 2145
hyp/annot/run (color name)	pdfcreator	2089
..... 470	pdfencoding	1352
hyp/text/pdfstring	pdfinfo	2207
..... 500	pdfkeywords	2089
\hypercalcbp	pdflang	13, 2075
..... 11, 17, 139	pdflinkmargin	13
\HyperDestNameFilter	pdfmoddate	13, 2145
..... 554, 605	pdfproducer	2089
\hypersetup 1, 2, 5, 8, 9, 12, 17, 64, 97, 158	pdfremotestartview	11
\hypersetup keys:	pdfstartview	11
allcolor	pdfsubject	2089
..... 930	pdftitle	2089
bookmarkstyle	pdftrapped	2179
..... 12	pdfversion	1352
bordercolormodel	pdfview	11, 1993
..... 12, 951	plainpages	1384
colorfile	run	9, 1430
..... 930	runborder	13
colorlink	runborderstyle	13, 1040
..... 930	runcolor	930
colorlinks	unicode	1352
..... 916	url	9, 1430
colormenu	urlborder	13
..... 930	urlborderstyle	13, 1040
colormenu		
..... 930		
colorrun		
..... 930		
colorscheme		
..... 1, 1338		
colorurl		
..... 930		
debug		
..... 1362		
destlabel		
..... 12		
draft		
..... 1362		
extension		
..... 12, 1384		
file		
..... 9, 1430		
fileborderstyle		
..... 13, 1040		
filecolor		
..... 930		
final		
..... 1362		
hidefile		
..... 1314		
hidelink		
..... 1314		

urlcolor	930	\LayoutCheckField	2898
verbose	1362	\LayoutChoiceField	2693
hypertextnames (hypersetup key)	1384	\LayoutPushButtonField	2786, 2803
hypListbox internal commands:		\LayoutTextField	2631
_hypListbox	2711, 2757	\leavevmode	2632, 2730, 3451
hypRadio internal commands:		legacy commands:	
_hypRadio	2696, 2719	_legacy_if:nTF	146, 386, 390, 540, 2511, 2782, 2823, 2861, 2894, 3151
hypwritetorep internal commands:		\let	237, 267, 291, 309, 329, 357, 358, 359, 630, 680, 687, 741, 747, 791, 797, 840, 2389, 2390, 2392, 2393, 2420, 2471, 2580, 2581, 2582, 2613, 2614, 2648, 2649, 2691, 2978, 2985, 2987, 2992, 3009, 3043, 3460, 3474, 3477, 3486, 3497, 3504, 3507, 3509, 3517, 3525
_hypwritetorep	3421	link (hypersetup key)	9, 1430
I			
\ifcase	3205, 3329, 3364	linkborder (hypersetup key)	13
\ifcsname	2982	linkborderstyle (hypersetup key)	13, 1040
\ifdim	2661, 2700, 2701	linkcolor (hypersetup key)	930
\ifnum	3036, 3205, 3211, 3242, 3329, 3334, 3364, 3369, 3494, 3512, 3527	linkfileprefix (hypersetup key)	1384
\iftrue	3496	linktoc (hypersetup key)	1384
\ifx	2429, 2630, 2690, 3013, 3025, 3029, 3046, 3050, 3054, 3058, 3062, 3066, 3070, 3074, 3078, 3082, 3099, 3103, 3206, 3207, 3215, 3219, 3226, 3246, 3250, 3257, 3258, 3330, 3338, 3365, 3373, 3446, 3506	linktocpage (hypersetup key)	1384
\immediate	2413, 2416	localanchorname (hypersetup key)	1384
int commands:		M	
_int_compare:nNnTF	2002, 2028, 2489, 2749, 3128, 3181, 3291, 3313	\MakeButtonField	2791, 2806, 2830, 2865, 2876
_int_compare_p:nNn	3174, 3282	\MakeChoiceField	2766
_int_eval:n	724	\MakeFieldObject	2380, 2520, 2532, 2544, 2556, 2568, 2573
_int_max:nn	725	\MakeRadioField	2748
_int_new:N	425	\MakeTextField	2640
iow commands:		\mbox	1164
_iow_newline:	2931, 2937, 2950	menu (hypersetup key)	9, 1430
K			
\kern	3453	menuborder (hypersetup key)	13
keys commands:		menuborderstyle (hypersetup key)	13, 1040
_keys_define:nn	164, 182, 220, 369, 916, 932, 939, 951, 960, 981, 996, 1015, 1042, 1061, 1184, 1198, 1210, 1233, 1239, 1254, 1283, 1314, 1327, 1338, 1352, 1362, 1371, 1384, 1411, 1432, 1437, 1993, 2075, 2091, 2109, 2147, 2161, 2179, 2207, 2263, 2278, 2291, 2311	menucolor (hypersetup key)	930
_l_keys_key_str	169, 2133, 2376	mode commands:	
_keys_set:nn	161, 190, 209, 236, 266, 290, 308, 328, 377, 1344, 1351, 2184, 2211, 2214, 2215, 2216, 2217, 2272, 2301, 2376	_mode_if_horizontal:TF	550, 572
_keys_set_known:nn	2379	_mode_leave_vertical:	233, 263, 287, 305, 325, 593, 674, 733, 780, 817, 2760, 2788, 2805, 2827, 2857, 2900
\kvsetkeys	160, 2472	msg commands:	
L			
\label	8, 12	_msg_error:nn	385, 2802, 2875
		_msg_line_context:	68
		_g_msg_module_name_prop	11
		_msg_new:nnn	36, 43, 48, 52, 56, 63, 70, 77, 84, 91, 97, 104, 113, 123
		_msg_new:nnnn	12, 22
		_msg_warning:nn	148, 1358
		_msg_warning:nnn	157, 597, 835
		_msg_warning:nnnn	1202, 1243, 1480, 1524,

1558, 1581, 1643, 1672, 1706, 1721, 1734, 1747, 1772, 1805, 1821, 1848, 1879, 1912, 1926, 1953, 1987, 2069	\msg_warning:nnnnn 168, 1274, 1308, 1419, 1500, 1537, 1624, 1657, 1682, 1758, 1785, 1834, 1939, 1966, 2201, 2318, 2337, 2356, 2367
N	
naturalnames (hypersetup key)	1384
nesting (hypersetup key)	13
\newcommand	142 , 3425 , 3427 , 3429
\newcount	2585 , 2756
\NewDocumentCommand	2380
\NewExpandableDocumentCommand	356
\newlength	3444
\noexpand	2667 , 2668 , 2669 , 2670 , 2996
\nolinkurl	3 , 4
O	
ocgcolorfile (hypersetup key)	1180
ocgcolorlink (hypersetup key)	1180
ocgcolorlinks (hypersetup key)	1180
ocgcolormenu (hypersetup key)	1180
ocgcolorrun (hypersetup key)	1180
ocgcolorurl (hypersetup key)	1180
P	
pageanchor (hypersetup key)	1384
\paperwidth	3
\PassOptionsToPackage	392 , 1376 , 1381
pdf commands:	
\pdf_bdcobject:nn	1166 , 1169
\pdf_destination:nn	25 , 134 , 567
\pdf_destination:nnnn	560
\pdf_emc:	1168 , 1174
\pdf_link_user:nnn	2763
\pdf_name_from_unicode_e:n	215 , 704 , 819
\pdf_object_if_exist:nTF	702
\pdf_object_new:nn	1088 , 1089 , 1090 , 1091 , 2918 , 2919 , 2920
\pdf_object_ref:n	135 , 714 , 1094 , 1096 , 1118 , 1119 , 1122 , 1126 , 1131 , 1136 , 1141 , 2504 , 2506 , 2976
\pdf_object_ref_last:	735 , 2306
\pdf_object_unnamed_write:nn	734 , 2302
\pdf_object_write:nn	1092 , 1098 , 1108 , 1120 , 2921 , 2963 , 2970
\pdf_pageobject_ref:n	1865 , 1890 , 2600
\pdf_string_from_unicode:nnN	511
\pdf_version:	1528 , 1647 , 1676 , 1710 , 1725 , 1751 , 1776 , 1809 , 1825 , 1930 , 1957
\pdf_version_compare:NnTF	1518 , 1609 , 1631 , 1667 , 1700 , 1715 , 1741 , 1766 , 1792 , 1815 , 1920 , 1947
\pdf_version_compare_p:Nn	771 , 1181 , 1230
\pdf_version_major:	145 , 1182 , 1205 , 1231 , 1247
\pdf_version_minor:	144 , 1205 , 1247
pdfannot commands:	
\pdfannot_box:nnnn	2635 , 2733 , 2792 , 2831 , 2866 , 2903
\pdfannot_box_ref_last:	2427 , 2443
\pdfannot_dict_put:nnn	675 , 735 , 781 , 820 , 972 , 1007 , 1031 , 1053 , 1077 , 1259 , 1290
\pdfannot_dict_remove:nn	966 , 1002 , 1023 , 1048 , 1069 , 1267 , 1300
\pdfannot_link:nnn	676 , 736 , 782 , 821
\pdfannot_link_goto_begin:nw	610
\pdfannot_link_goto_end:	615
\pdfannot_link_margin:n	8 , 1562
\c_pdfannot_link_types_seq	479
pdfauthor (hypersetup key)	2089
\pdfbookmark	3430
pdfborder (hypersetup key)	13
pdfborderstyle (hypersetup key)	13 , 1040
pdfcreationdate (hypersetup key)	13 , 2145
pdfcreator (hypersetup key)	2089
\pdfdest	11
pdfdict commands:	
\pdfdict_new:n	498 , 653 , 690 , 749 , 806
\pdfdict_put:nnn	207 , 214 , 499 , 654 , 655 , 666 , 669 , 691 , 692 , 704 , 711 , 721 , 731 , 750 , 751 , 762 , 777 , 807 , 808 , 818 , 1588 , 1589 , 1593 , 1594 , 2315 , 2325 , 2329 , 2332 , 2334 , 2353 , 2362 , 2364
\pdfdict_remove:nn	204 , 774 , 1598 , 1599
\pdfdict_use:n	675 , 734 , 781 , 787 , 820 , 825 , 2304
pdfencoding (hypersetup key)	1352
\pdfescapestring	2985
pdffile commands:	
\pdffile_embed_file:nnn	706
PDFForm internal commands:	
\PDFForm_hypName	3024 , 3027 , 3031 , 3034
pdfinfo (hypersetup key)	2207
pdfkeywords (hypersetup key)	2089
pdflang (hypersetup key)	13 , 2075
pdflinkmargin (hypersetup key)	13
pdfmanagement commands:	
\pdfmanagement_add:nn	1571

<code>\pdfmanagement_add:nnn</code>	<code>\prop_item:Nn</code>
..... 534, 1118, 1119,	2499
1141, 1447, 1472, 1488, 1492, 1512,	<code>\prop_map_inline:Nn</code>
1520, 1549, 1604, 1614, 1638, 1652, 894, 958, 994, 1021,
1664, 1669, 1694, 1702, 1717, 1743,	1029, 1040, 1067, 1075, 1144, 1252,
1768, 1800, 1817, 1863, 1888, 1902,	1288, 1298, 1342, 1430, 2476, 2483
1922, 1949, 1978, 2084, 2103, 2121,	<code>\prop_new:N</code>
2132, 2156, 2170, 2188, 2287, 2306,	2386, 2387
2478, 2496, 2503, 2505, 2507, 2513	<code>\protect</code> 237, 267, 291, 309, 329, 408, 630,
<code>\pdfmanagement_if_active_p:</code> ... 384	680, 687, 741, 747, 791, 797, 840, 3446
<code>\pdfmanagement_remove:nn</code>	<code>\providecommand</code>
... 1444, 1468, 1476, 1496, 1508, 140, 141, 2414, 2417, 2645, 2981
1533, 1545, 1553, 1567, 1576, 1620,	<code>\ProvidesFile</code>
1635, 1654, 1691, 1730, 1755, 1781,	3
1796, 1830, 1860, 1885, 1898, 1907,	
1935, 1962, 1974, 1983, 2081, 2097,	
2115, 2153, 2167, 2284, 2297, 2517	
<code>\pdfmanagement_show:n</code>	
2479	
pdfmeta commands:	
<code>\pdfmeta_standard_item:n</code>	
..... 2269, 2274, 2275	
<code>\pdfmeta_standard_verify:nnTF</code> ...	
..... 815, 2779, 2858	
<code>\pdfmeta_standard_verify:nTF</code> ...	
..... 2509, 3092	
pdfmoddate (hypersetup key)	
13, 2145	
pdfproducer (hypersetup key)	
2089	
pdfremotestartview (hypersetup key) ..	
11	
pdfstartview (hypersetup key)	
11	
<code>\pdfstringdef</code>	
8, 9, 21, 26, 3041	
<code>\pdfstringdefDisableCommands</code>	
..... 357, 358, 359	
pdfsubject (hypersetup key)	
2089	
pdftitle (hypersetup key)	
2089	
pdftrapped (hypersetup key)	
2179	
pdfversion (hypersetup key)	
1352	
pdfview (hypersetup key)	
11, 1993	
pdfxform commands:	
<code>\pdfxform_new:nnn</code>	
2382	
<code>\pdfxform_ref:n</code>	
.. 2743, 2744, 2838, 2839, 3123, 3124	
<code>\phantom</code>	
2540, 2564	
<code>\phantomsection</code>	
3513	
plainpages (hypersetup key)	
1384	
prg commands:	
<code>\prg_generate_conditional_</code>	
variant:Nnn	
403	
<code>\prg_new_eq_conditional:NNn</code> ...	
402	
prop commands:	
<code>\prop_const_from_keyval:Nn</code>	
..... 436, 444, 3574, 3589,	
3604, 3619, 3634, 3649, 3665, 3680	
<code>\prop_gput:Nnn</code>	
11, 2400, 2407	
<code>\prop_if_empty:Nn</code>	
2481	
	<code>\prop_map_inline:Nn</code>
 894, 958, 994, 1021,
	1029, 1040, 1067, 1075, 1144, 1252,
	1288, 1298, 1342, 1430, 2476, 2483
	<code>\prop_new:N</code>
	2386, 2387
	<code>\protect</code> 237, 267, 291, 309, 329, 408, 630,
	680, 687, 741, 747, 791, 797, 840, 3446
	<code>\providecommand</code>
 140, 141, 2414, 2417, 2645, 2981
	<code>\ProvidesFile</code>
	3
	R
	<code>\ReadBookmarks</code>
	3435
	ref commands:
	<code>\ref_if_exist:nn</code>
	402
	<code>\ref_label:nn</code>
	398
	<code>\ref_value:nn</code>
	23, 419
	regex commands:
	<code>\regex_const:Nn</code>
	485
	<code>\regex_extract_once:NnNTF</code> 1843, 1874
	<code>\relax</code>
	630, 680, 687,
	741, 747, 791, 797, 840, 2648, 2649,
	2690, 2978, 3025, 3029, 3099, 3103,
	3206, 3207, 3215, 3219, 3246, 3250,
	3330, 3338, 3365, 3373, 3497, 3506
	<code>\RemoveFromHook</code>
	1452
	<code>\RenewDocumentCommand</code>
	151
	<code>\RequirePackage</code>
	6, 18, 388, 3433
	run (hypersetup key)
	9, 1430
	runborder (hypersetup key)
	13
	runborderstyle (hypersetup key) ..
	13, 1040
	runcolor (hypersetup key)
	930
	S
	<code>\selectfont</code>
	2527, 2539, 2551, 2563
	seq commands:
	<code>\seq_const_from_clist:Nn</code>
	434
	<code>\seq_count:N</code>
	2002, 2028
	<code>\seq_get_right:NN</code>
	2004
	<code>\seq_item:Nn</code>
	1845,
	1876, 1998, 2046, 2048, 2058, 2063
	<code>\seq_map_inline:Nn</code>

	471, 475, 479, 930, 1227, 1325, 2494
	<code>\seq_new:N</code>
	424
	<code>\seq_put_right:Nn</code>

	2035, 2036, 2037, 2485
	<code>\seq_set_split:Nnn</code>
	1997
	<code>\seq_sort:Nn</code>
	2487
	<code>\setbox</code>
	350
	<code>\setlength</code>
	3463, 3534
	<code>\setpdfinkmargin</code>
	8
	<code>\settowidth</code>
	2660
	<code>\show</code>
	2721

sort commands:		
\sort_return_same:	2492	
\sort_return_swapped:	2491	
\space	412, 2729, 3208, 3225,	
	3226, 3256, 3257, 3331, 3363, 3366	
\spacefactor	550, 572	
\stockwidth	3	
str commands:		
\c_colon_str	801	
\c_hash_str	671	
\c_percent_str	672	
\str_case:nnTF	3388	
\str_case_e:nnTF	1998	
\str_compare:nNnTF	2490	
\str_gset:Nn	955	
\str_gset_eq:NN	521	
\str_head:n	2190, 2195	
\str_if_eq:nnTF		
...	557, 2101, 2119, 2129, 2269, 3090	
\str_if_eq_p:nn	769, 1182, 1231	
\str_lowercase:n		
...	1998, 2124, 2172, 2191, 2196	
\str_new:N	426, 431, 432, 470	
\str_set:Nn	503	
\str_set_convert:Nnnn	8	
\str_set_eq:NN	523	
\str_tail:n	2191, 2196	
\str_uppercase:n	2184, 2190, 2195	
\string	2414,	
	2417, 2438, 2456, 3001, 3005, 3016	
\subpdfbookmark	3426	
sys commands:		
\c_sys_backend_str	3388	
T		
T _E X and L ^A T _E X 2 _ε commands:		
\@BOOKMARK	3432	
\@CheckBox	2881	
\@ChoiceMenu	2646	
\@Form	2469	
\@PushButton	2771	
\@Reset	2848	
\@Submit	2812	
\@TextField	2611	
\@bookmarksopenlevel	1463	
\@bsphack	397	
\@chapter	3486, 3487	
\@currDisplay	2660, 2729	
\@currValue	2727, 2743	
\@currentHref		
...	3467, 3480, 3491, 3521, 3538	
\@curropt	2645, 2655, 2658, 2723, 2726	
\@empty	2614, 2630, 3046,	
	3050, 3054, 3058, 3062, 3066, 3070,	
	3074, 3078, 3082, 3226, 3257, 3258	
\@endForm	2580	
\@esphack	399	
\@firstofone	2420, 2987, 2992, 3009	
\@firstoftwo	3528	
\@for	2656, 2724	
\@gobble	2471	
\@gobbletwo	3477	
\@hyper@launch	35, 801	
\@ifnextchar	245	
\@ifundefined	2471, 3001	
\@latex@warning	409	
\@mainaux	2413, 2416, 2436, 2455	
\@mkboth	3477	
\@nil	2999, 3012, 3018	
\@part	3504, 3511	
\@pdfauthor	21	
\@pdfborder	140	
\@pdfborderstyle	141	
\@savsf	550, 572	
\@schapter	3474, 3475	
\@secondoftwo	3530	
\@sect	3525, 3526	
\@spart	3517, 3518	
\@ssect	3460, 3461	
\@tempdima	2654, 2661, 2663, 2700,	
	2701, 2702, 2706, 2707, 2708, 2709	
\@tempdimb	2660, 2661	
\@typeset@protect	3446	
\@unexpandable@protect		
	237, 267, 291, 309, 329	
\@secnumdepth	3494, 3512, 3527	
\calc@bm@number	3438	
\check@bm@number	3437	
\Fld@additionalactions	3088, 3158,	
	3192, 3228, 3265, 3312, 3352, 3381	
\Fld@align	3117, 3203, 3278	
\Fld@altname	3025, 3027, 3099, 3101	
\Fld@annotflags	3112,	
	3165, 3198, 3234, 3273, 3322, 3357	
\Fld@annotnames	3097, 3114,	
	3167, 3200, 3236, 3275, 3324, 3359	
\Fld@bcolor	3136, 3138, 3219,	
	3221, 3250, 3252, 3286, 3299, 3301	
\fld@bcolor	3207	
\Fld@bordercolor	3132,	
	3134, 3176, 3185, 3187, 3206, 3215,	
	3217, 3246, 3248, 3284, 3295, 3297,	
	3330, 3338, 3340, 3365, 3373, 3375	
\Fld@borderstyle	3118,	
	3171, 3204, 3240, 3279, 3328, 3379	

\Fld@borderwidth .	2707, 2708, 3118, 3171, 3204, 3240, 3279, 3328, 3379
\Fld@calculate@code .	2429, 3058, 3060
\Fld@calculate@sortkey	2440
\Fld@cbsymbol	3140
\Fld@charsize 2706, 3144, 3225, 3256, 3307, 3363
\Fld@ckeckequals	2658, 2726
\Fld@choices	3227
\Fld@color .	3145, 3147, 3226, 3257, 3308
\Fld@default 2613, 2630, 2648, 2691, 2721, 2722, 2884, 3258, 3262, 3263, 3310
\Fld@flags	3116, 3169, 3202, 3238, 3277, 3326, 3361
\Fld@format@code	3050, 3052
\Fld@height	2616, 2637, 2640, 2651, 2709, 2735, 2748, 2766, 2815, 2851, 2887, 2905
\Fld@keystroke@code	3046, 3048
\Fld@listcount .	2720, 2728, 2749, 2756
\Fld@mappingname .	3029, 3031, 3103, 3105
\Fld@maxlen	3313, 3315
\Fld@menulength	2653, 2659, 2706
\Fld@name 2612, 2647, 2772, 2883, 3024, 3098
\Fld@onblur@code	3066, 3068
\Fld@onclick@code	3191
\Fld@onenter@code	3078, 3080
\Fld@onexit@code	3082, 3084
\Fld@onfocus@code	3062, 3064
\Fld@onmousedown@code	3070, 3072
\Fld@onmouseup@code	3074, 3076
\Fld@pageobjref 2582, 2596, 2981, 3113, 3166, 3199, 3235, 3274, 3323, 3358
\Fld@radiosymbol	3254
\Fld@rotation	3128, 3130, 3174, 3181, 3183, 3205, 3211, 3213, 3242, 3244, 3282, 3291, 3293, 3329, 3334, 3336, 3364, 3369, 3371
\Fld@submitflags	3350
\Fld@validate@code	3054, 3056
\Fld@value 2614, 2630, 2649, 2690, 2691, 3311
\Fld@width	2615, 2629, 2636, 2640, 2650, 2689, 2700, 2702, 2734, 2748, 2766, 2784, 2814, 2825, 2850, 2862, 2886, 2896, 2904
\Fld@X@additionalactions 3045, 3090, 3093
\Form@action	3348
\G@refundefinedtrue	408
\H@old@part	3504, 3515
\H@old@schapter	3474, 3483
\H@old@sect	3525, 3536, 3545
\H@old@spart	3517, 3523
\H@old@ssect	3460, 3465
\href@	245, 251
\href@split	251, 252
\Hy@abspage	544
\Hy@activeanchorfalse	589
\Hy@activeanchortrue	582
\Hy@AtBeginDocument	2394, 2410
\Hy@bookmarksnumberedfalse	1456
\Hy@bookmarksnumberedtrue	1457
\Hy@bookmarksopenfalse	1460
\Hy@bookmarksopentru	1461
\Hy@bookmarkstype	1464
\Hy@chapapp	3478, 3489
\Hy@colorlink	31
\Hy@currentbookmarklevel	3419
\Hy@DisableOption	150
\Hy@drafttrue	1375
\Hy@escapeform 2634, 2732, 2762, 2790, 2829, 2864, 2902, 2983, 2990, 2993
\Hy@escapestring	2985, 2987, 2992, 2995, 2996, 2999, 3000, 3009, 3012, 3048, 3052, 3056, 3060, 3064, 3068, 3072, 3076, 3080, 3084, 3140, 3191, 3254, 3310, 3311, 3348
\Hy@finaltrue	1380
\Hy@FormObjects	2475, 2916, 2978
\Hy@gtemp	3041, 3043
\Hy@href	245
\Hy@href@nextactionraw	208
\Hy@href@page	196
\Hy@hypertextnamesfalse	1390
\Hy@hypertextnametrue	1389
\Hy@linkfileprefix	1392
\Hy@linktoc	1415
\Hy@localanchornamefalse	1395
\Hy@localanchornametrue	1394
\Hy@MakeCurrentHrefAuto 3462, 3478, 3489, 3519, 3533
\Hy@naturalnamesfalse	1399
\Hy@naturalnametrue	1398
\Hy@next	3488, 3497, 3500
\Hy@numberline	143, 3420
\Hy@org@chapter	3486, 3501
\Hy@OutlineName	3436
\Hy@OutlineRerunCheck	3434
\Hy@pageanchorfalse	1403
\Hy@pageanchortrue	1402
\Hy@pdfmajorversion	145
\Hy@pdfminorversion	144
\Hy@pdfstringtrue	27, 516

\Hy@pdfversion	3036	\hyper@anchorstart	574, 3455, 3480, 3491, 3521
\Hy@plainpagesfalse	1407	\hyper@link	31, 618
\Hy@plainpagetrue	1406	\hyper@linkend	31, 644
\Hy@pstringdef	526, 3016	\hyper@linkfile	297, 695
\Hy@PutCatalog	532	\hyper@linklaunch	34, 315, 753, 804
\Hy@raisedlink	3454, 3479, 3490, 3520	\hyper@linknamed	35, 153, 810
\Hy@RestoreLastskip	571	\hyper@linkstart	31, 633
\Hy@safe@activetrue	673, 2432	\hyper@linkurl	279, 343, 657
\Hy@SaveLastskip	551	\hyper@normalise	245, 275, 292, 310, 337, 350
\Hy@secnum@part	3507, 3509, 3512	\HyPL@Labels	534, 544
\Hy@SectionAnchorHref	3445, 3467, 3538	\HyPL@storePageLabel	532
\Hy@SectionHShift	3444, 3453, 3463, 3534	\HyPsd@SanitizeForOutFile	3423
\Hy@StepCount	2659, 2728	\if@filesw	2412, 2435, 2454
\Hy@temp	2433, 2448, 2998, 2999	\ifFld@combo	2674, 2704
\Hy@unicodedefalse	3038	\ifFld@hidden	2629, 2689
\Hy@VerboseAnchor	552	\ifFld@multiline	2617
\Hy@verbosefalse	1366	\ifFld@popdown	2675
\Hy@VerboseLinkStart	622, 637	\ifFld@radio	2671, 2694
\Hy@VerboseLinkStop	628, 650, 683, 743, 793, 831	\ifHy@implicit	3440
\Hy@verbostrue	1365	\ifHy@pdfescapeform	2984, 2994
\Hy@VersionChecked	136	\ifHy@unicode	3037
\Hy@WrapperDef	548	\kv@set@family@handler	156
\Hy@xspace@end	625, 682, 742, 792, 830	\ltx@empty	2389, 2390, 2429, 2580, 2581, 2582, 2613
\HyAnn@AbsPageLabel	2581, 2587, 2633, 2731, 2761, 2789, 2828, 2863, 2901	\ltx@gobble	2392
\HyAnn@Count	2585, 2586, 2589, 2590, 2591, 2592, 2593, 2594, 2598, 2602	\ltx@gobbletwo	2393
\HyField@AddToFields	2423, 2641, 2750, 2767, 2798, 2842, 2871, 2908	\ltx@ifUndefined	3485, 3495
\HyField@afields	2389	\ltx@one	2589
\HyField@AfterAuxOpen	2394, 2420, 2453	\ltx@ReturnAfterFi	3017
\HyField@AuxAddToCoFields	2393, 2405, 2417, 2438	\ltx@zero	2586
\HyField@AuxAddToFields	2392, 2398, 2414, 2456	\m@ne	3494, 3509
\HyField@cofields	2390	\OBJ@OCG@view	142
\HyField@FlagsCheckBox	2893	\p@	2663
\HyField@FlagsChoice	2699	\pdf@ifdraftmode	2473
\HyField@FlagsPushButton	2781, 2821, 2860	\PDF@SetupDoc	137
\HyField@FlagsRadioButton	2695	\pdf@strcmp	2489
\HyField@FlagsSubmit	2822	\PDFForm@Check	2902, 2907, 3109
\HyField@FlagsText	2628	\PDFForm@List	2762, 2765, 3196
\HyField@PDFChoices	2759	\PDFForm@Name	2627, 2688, 2778, 2892, 3023
\HyField@SetKeys	2624, 2665, 2668, 2687, 2774, 2817, 2853, 2888	\PDFForm@Push	2790, 2796, 3162
\HyPat@ObjRef	3392, 3399, 3406, 3413	\PDFForm@Radio	2732, 2738, 3231
\hyper@@link	253	\PDFForm@Reset	2864, 2870, 3355
\hyper@anchor	574	\PDFForm@Submit	2829, 2836, 3320
\hyper@anchorend	574, 3455, 3480, 3491, 3521	\PDFForm@Text	2634, 2639, 3270
		\protected@edef	594
		\strip@pt	3144, 3225, 3256, 3307, 3363
		\toks@	3465, 3471, 3536, 3542
		\tw@	3466, 3471, 3537, 3542
		\url@	350, 353
		\Url@def	351
		\Url@HyperHook	351

<code>\XR@ext</code>	1386	<code>\tl_new:N</code>	175,
<code>\z@</code>	350, 2654, 3205, 3211,		176, 177, 178, 423, 427, 428, 429,
	3242, 3329, 3334, 3364, 3369, 3507		430, 453, 454, 455, 456, 457, 458,
<code>\zref@labelbylist</code>	2591		459, 460, 461, 469, 2464, 2466, 2717
<code>\zref@labelbyprops</code>	2590	<code>\tl_set:Nn</code>	195, 196, 208, 240,
<code>\zref@refused</code>	2592		243, 270, 273, 332, 335, 362, 463,
<code>\texorpdfstring</code>	27		464, 465, 466, 467, 468, 700, 1841,
text commands:			1845, 1849, 1872, 2007, 2010, 2017,
<code>\text_expand:n</code>	700		2020, 2021, 2022, 2023, 2024, 2025,
<code>\text_purify:n</code>	8, 503		2030, 2034, 2070, 2465, 2467, 2718
<code>\textcolor</code>	2471, 2570, 2575	<code>\tl_set_eq:NN</code>	601
<code>\textsf</code>	2570, 2575	<code>\tl_to_str:N</code>	702, 709, 714
<code>\the</code>	544,	<code>\tl_to_str:n</code>	
	2590, 2591, 2592, 2593, 2594, 2598,	 18, 19, 20, 31, 411, 2156, 2170
	2602, 2702, 2706, 2709, 3471, 3542	token commands:	
<code>\thepage</code>	412	<code>\token_to_str:N</code>	27
<code>\ThisShouldNotHappen</code>	3003	<code>\toks</code>	3466, 3471, 3537, 3542
<code>\title</code>	2		
tl commands:			
<code>\c_space_tl</code>	143, 1095,		
	1126, 1131, 1136, 2743, 2744, 2751,		
	2927, 2929, 2939, 2941, 2943, 2946,		
	2948, 3144, 3147, 3279, 3307, 3308		
<code>\tl_const:Nn</code>	433		
<code>\tl_gput_right:Nn</code>	544		
<code>\tl_gset:Nn</code>	1856, 1876, 1880		
<code>\tl_if_blank:nTF</code>			
	719, 2095, 2113, 2151, 2165, 2282, 2295		
<code>\tl_if_empty:NTF</code>			
	... 595, 1442, 3132, 3136, 3145, 3308		
<code>\tl_if_empty:nTF</code>			
 202, 964, 1000, 1019,		
	1046, 1065, 1633, 1689, 1794, 2079		
<code>\tl_if_empty_p:N</code>	1858, 1883		
<code>\tl_if_eq:NnTF</code>	2005		
<code>\tl_if_exist:NTF</code>	3185, 3295, 3299		
<code>\tl_if_exist_p:N</code>	3176, 3284, 3286		
<code>\tl_if_head_eq_charcode:nNTF</code>			
 845, 862, 879		

U

unicode (hypersetup key)	1352
<code>\Url</code>	3, 5, 179
url (hypersetup key)	9, 1430
<code>\url</code>	3–6, 21, 323, 350
urlborder (hypersetup key)	13
urlborderstyle (hypersetup key) .	13, 1040
urlcolor (hypersetup key)	930
<code>\urldef</code>	5, 21, 350, 352
<code>\Urlfont</code>	6
<code>\UrlSpecial</code>	4
use commands:	
<code>\use:n</code>	799

V

verbose (hypersetup key)	1362
--------------------------------	------

W

<code>\write</code>	2413, 2416, 2436, 2455
---------------------------	------------------------

X

<code>\x</code>	2666, 2686, 3470, 3472, 3541, 3543
-----------------------	------------------------------------