

The hyperref-generic module

A generic driver for hyperref

The L^AT_EX Project*

Version 0.96g, released 2024-03-26

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 (this requires L^AT_EX-2022-06-01 or newer):

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

2 Bookmarks / outlines

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

3 “Metadata”

“Metadata”, 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 are added by code from `l3pdfmeta`. (without the `pdfmanagement` the XMP-metadata can be added with packages like `pdfx` and `hyperxmp`).

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

The current handling of the metadata is problematic:

- External package like `hyperxmp` wants to access them too and for this had to patch a number of internal `hyperref` commands—which is a problem if the internal commands change (as happens with this new driver)
- `hyperref` (and also `hyperxmp`) tries to deduce some 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}}
```

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

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

- 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 `\DocumentMetadata` should be used:

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

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

4 Dates

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

5 PDF page size (mediabox)

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

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

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

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

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

7.1 Background information

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

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

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

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

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

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

7.2 New Keys

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

- Beside `pdfborder` there are also `linkborder`, `urlborder` etc

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

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

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

The default is `phelype`.

7.3 Public interfaces

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 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	<code>urlbordercolor = [rgb]{1,1,0}</code>
color expression	<code>urlbordercolor = red!50!blue</code>
command	<code>urlbordercolor = \mycolor</code>

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

frenchlinks The option `frenchlinks` does nothing at all.

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

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

8 PDF strings

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

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

The new method is under heavy development!

Important differences here are

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

9 Package options from hyperref

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

Options that still must be set as package options are

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

Options that can be without problems set as package options are

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

Ignored options:

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

10 Disabling links

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

`link` (*setup key*) The `draft` option is a global option that can't be undone (at least not easily). So the 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.

`nested-links` (*setup key*)

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

11 Draftmode

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

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

12 Dropped options

A number of options are ignored by this driver

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

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

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

unicode This is always true.

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

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

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

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

13 Destinations

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

From these instructions two (`Fit` and `FitB`) don't take an argument. All others take one (`FitH`, `FitV`, `FitBH`, `FitBV`) or more (`XYZ`, `FitR`) arguments. These arguments are normally coordinates, `XYZ` takes also a zoom factor. The coordinates are absolute coordinates in `bp` relative to the lower left corner of the PDF.

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

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

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

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

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

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

- `Fit`, `FitB`, `FitH`, `FitV`, `FitBH`, `FitBV` which can be followed by a positive integer (separated by a space) or the keyword `null`. The number can be gives as a $\langle dimension\ expression \rangle$ 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 is it is quite unpredictable how viewers which support `FitR` handles zero sizes.
- `pdfstartview` and `pdfremotestartview` will pass the values as absolute coordinates.

13.1 Names of destinations

`hyperref` creates two types of destination names: For numbered structures (so when the anchor is set by `\refstepcounter`) it builds the name from the counter name and the `\theH...` representation: `<counter name>.\theH<counter name>`.

For unnumbered structures, e.g. starred chapters or anchors created with `\phantomsection` it uses names like `section*.<number>` and `chapter*.<number>`.

Typically the name of destination can be retrieved by setting a label, this works also with unnumbered sections. The anchor and also the page can be retrieve in an expandable way with the help of commands from the `refcount` package which is loaded by `hyperref`. For example with the following commands it is possible to use the label to create a bookmark:

```
\bookmark[dest=\getrefbykeydefault{label}{anchor}{Doc-Start}]{my bookmark}
\bookmark[dest=page.\getrefbykeydefault{label}{page}{Doc-Start}]{my bookmark}
```

If a `\HyperDestNameFilter` is defined, this must be added around the definition, so actually the full code has to look like this

```
\bookmark[dest=
\HyperDestNameFilter{\getrefbykeydefault{label}{anchor}{Doc-Start}}]{mysection}
```

To simplify this `hyperref` provides `\hyperget{anchor}{label}` and `\hyperget{pageanchor}{label}`

14 Assorted key descriptions

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

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

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

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

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

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

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

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

will create

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

<code>nesting</code> (<i>setup key</i>)	Typically PDF viewer can handle only GoToR annotations pointing to a PDF. So normally the default value <code>pdf</code> 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.		
<code>pdfborder</code> (<i>setup key</i>)	This key set accept as value three numbers or three numbers and an array describing		
<code>linkborder</code> (<i>setup key</i>)	a dash pattern, examples are <code>0 0 1</code> or <code>0 0 1 [3 2]</code> . The first two numbers should		
<code>urlborder</code> (<i>setup key</i>)	according to the reference set round corners, but PDF viewer seem to ignore it. The		
<code>runborder</code> (<i>setup key</i>)	third number is the line width of the border. Settings done with <code>pdfborderstyle</code> should		
<code>menuborder</code> (<i>setup key</i>)	take precedence.		
<code>pdfborderstyle</code> (<i>setup key</i>)	The value of this key is the content of the BS dictionary. As an example		
<code>linkborderstyle</code> (<i>setup key</i>)	<code>/Type/Border /W 1 /S/U /D[3 2]</code>		
<code>urlborderstyle</code> (<i>setup key</i>)	Key	Values	comment / example
<code>fileborderstyle</code> (<i>setup key</i>)	<code>/Type</code>	<code>/Border</code>	optional
<code>runborderstyle</code> (<i>setup key</i>)	<code>/W</code>	<code><number></code>	Width of border line
<code>menuborderstyle</code> (<i>setup key</i>)	<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])
<code>pdfcreationdate</code> (<i>setup key</i>)	Setting these keys is normally not needed. If they are used the values of the first		
<code>pdfmoddate</code> (<i>setup key</i>)	two keys are stored directly in the Info dictionary for <code>/Creationdate</code> and <code>/ModDate</code> .		
<code>pdfmetadate</code> (<i>setup key</i>)	All three keys are used in XMP-metadata. The values are converted to strings but not processed further, so they should have the correct PDF format without the enclosing parentheses, e.g. <code>D:20200202111111+01'00'</code> .		
<code>pdflinkmargin</code> (<i>setup key</i>)	As described in the <code>hyperref</code> manual the behaviour differs between the backends: with dvips it is possible to change links locally, pdf _l atex and lua _l atex work by page, with dvipdfmx the setting is global (and has to be done in the preamble).		
<code>pdflang</code> (<i>setup key</i>)	The key will work, but it is recommended to the set the language in <code>\DocumentMetadata</code> instead.		

File I

hyperref-generic driver implementation

```

1 \*package>
2 \@@=hyp>
3 \ProvidesFile{hgeneric-testphase.def}[2024-03-26 v0.96g %
4   generic Hyperref driver for the LaTeX PDF management testphase bundle]
5
6 \RequirePackage{etoolbox} %why?
```

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

```

7 \long\def\Hy@ReturnAfterFi#1\fi{\fi#1}
8 \ExplSyntaxOn
9 \file_input:n {hyperref-colorschemes.def}
10 \ExplSyntaxOff
```

1 messages

Redirect the message name:

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

At first a message for the testing of the resource management

```
13 \cs_if_exist:NTF \DocumentMetadata
14 {
15   \msg_new:nnnn
16   { hyp }
17   { missing-resource-management }
18   { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
19   {
20     Activate~it~with ~\
21     \tl_to_str:n{\DocumentMetadata{<options>}}\\
22     before~\tl_to_str:n{\documentclass}
23   }
24 }
25 {
26   \msg_new:nnnn
27   { hyp }
28   { missing-resource-management }
29   { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
30   {
31     Activate~it~with ~\
32     \tl_to_str:n{\RequirePackage{pdfmanagement-testphase}}\\
33     \tl_to_str:n{\DocumentMetadata{<options>}}\\
34     before~\tl_to_str:n{\documentclass}
35   }
36 }
```

The pdfversion should be set in \DocumentMetadata

```
37 \msg_new:nnnn
38 { hyp }
39 { pdfversion-disabled }
40 {
41   This~hyperref~driver~ignores~the~pdfversion~key!\\
42   Set~the~pdfversion~in~\token_to_str:N \DocumentMetadata
43 }
44 {
45   For example:\\
46   \tl_to_str:n
47   {
48     \DocumentMetadata { pdfversion=1.7 }
49   }
50 }
```

A generic message for ignored keys.

```
51 \msg_new:nnn
52 { hyp }
53 { key-dropped }
54 {
55   This~hyperref~driver~ignores~the~key~#1!\\
56   Please~check~the~documentation.
57 }
```

```
57 }
```

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

```
58 \msg_new:nnn
59 { hyp }
60 { pdfa-no-push-button }
61 { PDF/A:~Push~button~with~JavaScript~is~prohibited }
62
63 \msg_new:nnn
64 { hyp }
65 { pdfa-no-reset-button }
66 { PDF/A:~Reset~action~is~prohibited }
```

pdf/A message for not allowed Named actions

```
67 \msg_new:nnn
68 { hyp }
69 { pdfa-no-named-action }
70 { PDF/A:~Named~action~#1~is~prohibited }
```

A message if the destination name is empty.

```
71 \msg_new:nnn
72 { hyp }
73 { empty-destination-name }
74 {
75   Empty~destination~name,\\
76   using~'~#1'
77 }
```

A message if the destination check fails

```
78 \msg_new:nnn
79 { hyp }
80 { invalid-destination-value }
81 {
82   Invalid~value~'~#1'~of~'~#2'  \\
83   is~replaced~by~'Fit'~\msg_line_context:.
84 }
```

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

```
85 \msg_new:nnn
86 { hyp }
87 { ignore-deprecated-or-unknown-option-in-pdf-version }
88 {
89   Option~'~#1'~is~unknown~or~deprecated~in\\
90   pdf~version~#2.~Ignored.
91 }
92 \msg_new:nnn
93 { hyp }
94 { ignore-deprecated-or-unknown-value-in-pdf-version }
95 {
96   Value~'~#1'~is~unknown~or~deprecated~in\\
97   pdf~version~#2.~Ignored.
98 }
99 \msg_new:nnn
100 { hyp }
101 { replace-deprecated-or-unknown-value-in-pdf-version }
102 {
```



```

103     Value~'#1'~is~unknown~or~deprecated~in\\
104     pdf~version~#2. Value~'#3'~is used instead.
105 }

```

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:

```

106 \msg_new:nnn
107 { hyp }
108 { unknown-key }
109 {
110     unknown~key~#2~of~module~'#1'~set~to~'#3'.
111 }
112 \msg_new:nnn
113 { hyp }
114 { unknown-key-to-Hyp }
115 {
116     ignored~in~family~Hyp~unknown~key~#1.
117 }

```

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

```

118 \cs_new:Npn \__hyp_clist_display:n #1 {*~#1\\}
119 \msg_new:nnn
120 { hyp }
121 { unknown-choice }
122 {
123     Value~'#3'~is~invalid~for~key~'#1'.\\
124     The~key~accepts~only~the~choices\\
125     \clist_map_function:nN { #2 }\__hyp_clist_display:n
126 }
127
128 \msg_new:nnn
129 { hyp }
130 { unknown-choice+empty }
131 {
132     Value~'#3'~is~invalid~for~key~'#1'.\\
133     The~key~accepts~only~the~choices\\
134     \clist_map_function:nN { #2 }\__hyp_clist_display:n
135     An~empty~value~removes~the~setting.
136 }
137
138 \msg_new:nnn
139 { hyp }
140 { no-bool }
141 {
142     Value~'#2'~is~invalid~for~key~'#1'.\\
143     The~key~accepts~only~the~choices\\
144     *~true\\
145     *~false \\
146     *~and~an~empty~value~which~removes~the~setting.\\
147     No~value~is~equivalent~to~using~'true'.
148 }

```

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

```

149 \msg_new:nnn

```

```

150 { hyp }
151 { empty-info-value }
152 {
153   Empty-value-for-key-#1.\
154   This~isn't~honored~by~all~backends.
155 }

```

2 Variants

```

156 \cs_generate_variant:Nn\pdf_destination:nn {nf}
157 \cs_generate_variant:Nn\pdf_object_ref:n {e}
158 \cs_generate_variant:Nn\pdf_pageobject_ref:n {e}

```

3 Overwriting/providing commands from hyperref

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

```

159 \chardef\Hy@VersionChecked=1 %don't check the version!
160 %\cs_set_protected:Npn \PDF@SetupDoc{}
161 %\PDF@FinishDoc{}% dummy needed for hyperref ...

```

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

\hypercalcbp

```

162 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n

```

(End of definition for \hypercalcbp. This function is documented on page 18.)

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

```

163 \providecommand\@pdfborder{}
164 \providecommand\@pdfborderstyle{}
165 \newcommand\OBJ@OCG@view {} % needed in hyperref
166 \def\Hy@numberline#1{#1\c_space_tl} %needed by bookmark

```

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

```

167 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
168 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:
169 \legacy_if:nT { Hy@setpdfversion }
170 {
171   \msg_warning:nn { hyp }{ pdfversion-disabled }
172 }
173 \Hy@DisableOption{pdfversion}

```

\Acrobatmenu should use the new internal link command

```

174 \RenewDocumentCommand \Acrobatmenu { m m }
175 {
176   \hyper@linknamed {#1} {#2}
177 }

```

`\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!

```

178 % TODO should go at some time ...
179 % \kv@set@family@handler{Hyp}
180 % { \msg_warning:nne {hyp}{unknown-key-to-Hyp}{#1} }
181 \cs_set_protected:Npn \hypersetup #1
182 {
183     %\kvsetkeys{Hyp} {#1}
184     \keys_set:nn { hyp }{ #1 }
185 }
186 % TODO for now unknown keys should only give warnings.
187 \keys_define:nn { hyp }
188 {
189     unknown .code:n =
190     {
191         \msg_warning:nneee { hyp } { unknown-key }
192         { hyp }{ \l_keys_key_str } { #1 }
193     }
194 }

```

Hyperref creates a number of destinations automatically. E.g. in unnumbered chapters and sections and with `\phantomsection`. The following key allows to force a specific name for the destination so that it can be used by bookmarks.

```

195 \keys_define:nn { hyp }
196 {
197     next-anchor .code:n =
198     {
199         \AddToHookNext{__hyp/dest/make}
200         {\Hy@MakeCurrentHref{#1}}
201     }
202 }

```

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`)

```

203
204 \bool_new:N \l__hyp_href_url_encode_bool
205 \bool_new:N \l__hyp_href_url_ismap_bool
206 \tl_new:N \l__hyp_href_url_protocol_tl
207 \tl_new:N \l__hyp_href_pdf_destination_tl
208 \tl_new:N \l__hyp_href_pdf_page_tl
209 \tl_new:N \l__hyp_href_run_parameter_tl
210 \cs_new_protected:Npn \__hyp_href_url_format: {\begingroup\Url}
211
212
213 \keys_define:nn { hyp / href }
214 {
215     ,urlencode .bool_set:N = \l__hyp_href_url_encode_bool
216     ,format .code:n = { \cs_set:Nn \__hyp_href_url_format: {#1} },
217     ,protocol .tl_set:N = \l__hyp_href_url_protocol_tl
218     ,destination .tl_set:N = \l__hyp_href_pdf_destination_tl

```

```

219 ,pdfremotestartview .code:n =
220 {
221   \keys_set:nn { hyp }
222   { pdfremotestartview = #1 }
223 }
224 ,page .code:n =
225 {
226   \tl_set:Nn \l__hyp_href_pdf_page_tl {#1}
227   \tl_set:Nn \Hy@href@page {#1}
228 }
229 ,ismap .bool_set:N = \l__hyp_href_url_ismap_bool
230 ,run-parameter .tl_set:N = \l__hyp_href_run_parameter_tl
231 ,nextactionraw .code:n =
232 { %perhaps some safety match later, see hyperref code
233   \tl_if_empty:nTF {#1}
234   {
235     \pdfdict_remove:nn{l_hyp/annot/A}{Next}
236   }
237   {
238     \pdfdict_put:nnn{l_hyp/annot/A}{Next}{#1}
239     \tl_set:Nn \Hy@href@nextactionraw {/Next~#1}
240     \keys_set:nn {hyp }{ pdfnewwindow = true}
241   }
242 }
243 ,afrelationship .code:n =
244 {
245   \pdfdict_put:nne
246   { l_pdffile/Filespec}{AFRelationship}{ \pdf_name_from_unicode_e:n {#1}}
247 }
248
249 }
250
251 \keys_define:nn { hyp }
252 {
253   ,href / urlencode .bool_set:N = \l__hyp_href_url_encode_bool
254   ,href / urlencode .default:n = {true}
255   ,href / urlencode .initial:n = {false}
256   ,href / protocol .tl_set:N = \l__hyp_href_url_protocol_tl
257   ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
258   ,href / format .code:n = { \cs_set:Nn \__hyp_href_url_format:{#1} }
259 }
260
261 \hook_new_pair:nn{cmd/href/before}{cmd/href/after}
262
263 \DeclareRobustCommand*{\href}[1][\relax]{%
264   \mode_leave_vertical:
265   \hook_use:n{cmd/href/before}
266   \group_begin:
267   \keys_set:nn { hyp / href } {#1}
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   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
276 }
277
278 \begingroup
279   \catcode'\$=6 %
280   \catcode'\#=12 %
281   \gdef\href@#1{\expandafter\href@split$1##\}%
282   \gdef\href@split$1#$2#$3\\$4{%
283     \hyper@@link{$1}{$2}{$4}%<---__hyp-docstrip doubling!
284   \endgroup
285   \hook_use:n{cmd/href/after}
286 }%
287 \endgroup
288
289 \hook_new_pair:nn{cmd/hrefurl/before}{cmd/hrefurl/after}
290
291 \DeclareRobustCommand*{\hrefurl}[1] []
292 {
293   \mode_leave_vertical:
294   \hook_use:n{cmd/href/before}
295   \group_begin:
296   \keys_set:nn { hyp / href } {#1}
297   \bool_if:NTF \l__hyp_href_url_encode_bool
298   {
299     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
300   }
301   {
302     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
303   }
304   \hyper@normalise\__hyp_href_url_aux:nn}
305
306 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
307 {
308   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
309   \group_end:
310   \hook_use:n{cmd/href/after}
311 }
312
313 \hook_new_pair:nn{cmd/hrefpdf/before}{cmd/hrefpdf/after}
314 \DeclareRobustCommand*{\hrefpdf}[1] []
315 {
316   \mode_leave_vertical:
317   \hook_use:n{cmd/hrefpdf/before}
318   \group_begin:
319   \keys_set:nn { hyp / href } {#1}
320   \hyper@normalise\__hyp_href_pdf_aux:nn
321 }
322
323 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
324 {
325   \exp_args:Nno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
326   \group_end:

```

```

327     \hook_use:n{cmd/hrefpdf/after}
328   }
329
330 \hook_new_pair:nn{cmd/hrefrun/before}{cmd/hrefrun/after}
331 \DeclareRobustCommand*{\hrefrun}[1] []
332 {
333   \mode_leave_vertical:
334   \hook_use:n{cmd/hrefrun/before}
335   \group_begin:
336   \keys_set:nn { hyp / href } {#1}
337   \hyper@normalise\__hyp_href_run_aux:nn
338 }
339
340 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
341 {
342   \exp_args:Nno\hyper@linklaunch{#1}{#2}{\l__hyp_href_run_parameter_tl}
343   \group_end:
344   \hook_use:n{cmd/hrefrun/after}
345 }
346
347
348 \hook_new_pair:nn{cmd/url/before}{cmd/url/after}
349
350 \DeclareRobustCommand*{\url}[1] []
351 {
352   \mode_leave_vertical:
353   \hook_use:n{cmd/url/before}
354   \group_begin:
355   \keys_set:nn { hyp / href } {#1}
356   \bool_if:NTF \l__hyp_href_url_encode_bool
357   {
358     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
359   }
360   {
361     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
362   }
363   \hyper@normalise\__hyp_href_url_aux:n
364 }
365
366 \cs_new_protected:Npn \__hyp_href_url_aux:n #1
367 {
368   \exp_args:Nno
369   \hyper@linkurl{\__hyp_href_url_format: {#1}}
370   {\l__hyp_href_url_protocol_tl#1}
371   \group_end:
372   \hook_use:n{cmd/url/after}
373 }
374

```

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

```

375
376 \def\urldef#1#2{\begingroup\def\url{\hyper@normalise\url@}\setbox\z@\hbox\bgroup
377   \def\Url@HyperHook##1\endgroup{\Url@def{#1}{#2}}}%

```

```

378 % Because hyperref breaks \urldef and does not define its own (Grrrr!)...
379 \def\url@##1{\egroup\endgroup\DeclareRobustCommand#1{#2{##1}}}%
380 #2}
381

```

make the new commands compatible with `\pdfstringdef`:

```

382 \NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
383 \pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
384 \pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}
385 \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.

```

386 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
387 {
388   %\tl_set:cn {@#1}{#2}
389   \AddToDocumentProperties[hyperref]{#1}{#2}
390 }
391 \cs_generate_variant:Nn \__hyp_store_metadata:nn {en,ne,ee}

```

4.2 citecolor

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

```

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

```

```

416         \keys_set:nn { hyp }
417         {
418             linkbordercolor      = hyp/color/citeborder
419         }
420     }
421     \cs_gset_eq:NN \__hyp_citebordercolor_hook_init: \prg_do_nothing:
422 }
423

```

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

```

424 \bool_lazy_and:nnF
425 { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
426 { \msg_error:nn { hyp}{ missing-resource-management } }

```

Outlines/bookmarks require the bookmark package. TODO check pdfpagemode if bookmarks are suppressed. TODO We overwrite the color key here for now, but this should be moved to bookmark

```

427 \AddToHook { package/bookmark/after}
428 {
429     \define@key{BKM}{color}
430     {
431         \__hyp_color_set:ne {__hyp/tmpa}{#1}
432         \color_export:nVN
433         {__hyp/tmpa}
434         \g__hyp_bordercolormodel_str
435         \BKM@color
436     }
437 }
438 \legacy_if:nT { Hy@bookmarks }
439 {
440     \AddToHook{begindocument/before}[hyperref/bookmark]
441     {
442         \RequirePackage{bookmark}
443     }
444 }
445 \legacy_if:nT { Hy@draft}
446 {
447     \PassOptionsToPackage{draft}{bookmark}
448 }

```

6 Reference and label commands

This uses the in-built property module.

```

\__hyp_property_record:nn

```

```

449 %

```


A label command which adds the space commands from LaTeX:

```

450 \cs_new_protected:Npn \__hyp_property_record:nn #1 #2 %label/attributes
451 {
452   \@bsphack
453   \property_record:nn{#1}{#2}
454   \@esphack
455 }
```

we generate a few variants. We use ee-variants as they already exist in the module and once this is there it can go here.

```

456 \cs_generate_variant:Nn \__hyp_property_record:nn {ee}
```

(End of definition for __hyp_property_record:nn.)

7 Variables

7.1 Private temporary variables

At first a few generic tmp variables

```

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

(End of definition for \l__hyp_tmpa_tl and others.)

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

```

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

TODO: document and check use!

(End of definition for \l__hyp_dest_name_tmpa_tl and others.)

7.2 Constants

This variable is used if a destination name is empty.

```

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

(End of definition for \c__hyp_dest_undefined_tl.)

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

```

\c__hyp_annot_types_seq
\c__hyp_map_annot_hyp_prop
\c__hyp_map_hyp_annot_prop
469 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
470 {url,link,file,menu,run}
471 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
472 {
473   URI    = url,
```

```

474     GoTo  = link,
475     GoToR = file,
476     Named = menu,
477     Launch= run
478 }
479 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
480 {
481     url    = URI,
482     link   = GoTo,
483     file   = GoToR,
484     menu   = Named,
485     run    = Launch
486 }
487

```

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

7.3 Variables

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

```

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

```

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

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

```

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

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

```

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

It is also unclear how the `/Contents` entry would look at best. So we use sockets. The first argument is the target (url or destination), For `GoTo` we also pass the text as argument. The sockets should define `\l__hyp_link_Contents_tl`

```

504 \tl_new:N\l__hyp_link_Contents_tl
505 \socket_new:nn {hyp/link/GoTo/Contents}{2}

```

```

506 \socket_new:nn {hyp/link/URI/Contents}{1}
507 \socket_new_plug:nnn {hyp/link/GoTo/Contents}{default}
508 {
509   \__hyp_text_pdfstring:eoN
510   { Go~to~destination~#1 }
511   { \l__hyp_text_enc_info_print_tl }
512   \l__hyp_link_Contents_tl
513 }
514 \socket_new_plug:nnn {hyp/link/URI/Contents}{default}
515 {
516   \__hyp_text_pdfstring:eoN
517   { #1 }
518   { \l__hyp_text_enc_info_print_tl }
519   \l__hyp_link_Contents_tl
520 }
521 \socket_assign_plug:nn{hyp/link/GoTo/Contents}{default}
522 \socket_assign_plug:nn{hyp/link/URI/Contents}{default}

```

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

```

523 \tl_new:N \l__hyp_dest_pdfview_tl

```

(End of definition for `\l__hyp_dest_pdfview_tl`.)

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

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

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

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

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

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

```

524 \str_new:N \g__hyp_bordercolormodel_str

```

(End of definition for `\g__hyp_bordercolormodel_str`.)

7.4 Booleans

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

`\l_hyp_annot_colorurl_bool`

`\l_hyp_annot_colorfile_bool`

`\l_hyp_annot_colorrurl_bool`

`\l_hyp_annot_colormenu_bool`

```

525 \seq_map_inline:Nn \c__hyp_annot_types_seq

```

```

526 {

```

```

527   \bool_new:c {l_hyp_annot_color#1_bool}

```

```

528 }

```

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

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

`\l_hyp_annot_ocgcolorurl_bool`

`\l_hyp_annot_ocgcolorfile_bool`

`\l_hyp_annot_ocgcolorrurl_bool`

`\l_hyp_annot_ocgcolormenu_bool`

```

529 \seq_map_inline:Nn \c__hyp_annot_types_seq

```

```

530 {

```

```

531   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}

```

```

532 }

```

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

not_Named_bool\l__hyp_annot_Launch_bool

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

```
533 \seq_map_inline:Nn \c_pdfannot_link_types_seq
534 {
535   \bool_new:c {l__hyp_annot_#1_bool}
536   \bool_set_true:c {l__hyp_annot_#1_bool}
537 }
```

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

7.5 Boxes

\l__hyp_dest_box

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

```
538 \box_new:N \l__hyp_dest_box
```

(End of definition for \l__hyp_dest_box.)

7.6 Regex

\c__hyp_dest_startview_regex

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

```
539 \regex_const:Nn \c__hyp_dest_startview_regex
540 {
541   \A\ *
542   (?:
543     (?:XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
544     |
545     (?:Fit\b|FitB\b)
546     |
547     (?:\b(?:FitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1})
548     |
549     (?:FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
550   )
551 }
```

(End of definition for \c__hyp_dest_startview_regex.)

7.7 PDF dictionaries

l__hyp_page/Trans

This dictionary is used for page transitions.

```
552 \pdfdict_new:n {l__hyp_page/Trans}
553 \pdfdict_put:nnn {l__hyp_page/Trans}{Type}{/Trans}
```

(End of definition for l__hyp_page/Trans.)

8 PDF string conversion

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

hyp/text/pdfstring

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

(End of 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

```
555 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
556 {
557   \str_set:Ne #2 {\text_purify:n { #1 } }
558 }
```

(End of definition for __hyp_text_purify:nN.)

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

__hyp_text_cleanup:N

```
559 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
560 {
561
562 }
```

(End of definition for __hyp_text_cleanup:N.)

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

#2 is str variable, #1 should be one of	
utf8/string	(lit) (utf8/string)
utf8/string-raw	lit (utf8/string)
utf8/URI	(percent encoded url)
utf8/URI-raw	percent encoded url
utf16/hex	<HEX> (utf16/hex)
utf16/hex-raw	HEX (utf16/hex)
utf16/string	(lit) (utf16/string)
utf16/string-raw	lit (utf16/string)

__hyp_text_string_from_unicode:nN

```
563 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
564 {
565   \pdf_string_from_unicode:nVN { #1 } #2 #2
566 }
```

(End of definition for __hyp_text_string_from_unicode:nN.)

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

__hyp_text_pdfstring:nnN

```
567 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
568 {
569   \group_begin:
570   \Hy@pdfstringtrue
```

```

571 \hook_use:n {hyp/text/pdfstring}
572 \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
573 \__hyp_text_cleanup:N \l__hyp_text_tmpa_str
574 \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
575 \str_gset_eq:NN \g__hyp_text_tmpa_str\l__hyp_text_tmpa_str
576 \group_end:
577 \str_set_eq:NN #3 \g__hyp_text_tmpa_str
578 }
579 \cs_generate_variant:Nn \__hyp_text_pdfstring:nnN {enN,onN,eoN,ooN,noN}

```

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

!!! temporary until all instances are gone

```

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

```

This is a special version for info keys:

`__hyp_text_pdfstring_info:nN`

```

582 \cs_new_protected:Npn \__hyp_text_pdfstring_info:nN #1 #2
583 {
584   \__hyp_text_pdfstring:noN { #1 } { \l__hyp_text_enc_info_print_tl } #2
585 }

```

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

9 Pagelabels

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

The current implementation in `hyperref/hyperref` drivers:

xetex: `hxdetex.def`, line 80-110

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

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

dvipdfm: identical to `xetex`

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

luatex identical to `pdftex`

The code in `hyperref` inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-30 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
586 \cs_new_protected:Npn\__hyp_PageLabels_gpush:
587   {
588     \pdfmanagement_add:nne {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
589   }
590
591 \def\Hy@PutCatalog #1 {}
592
593
594 \legacy_if:nT { Hy@pdfpagelabels }
595   {
596     \cs_set_protected:Npn \HyPL@StorePageLabel #1
597       {
598         \tl_gput_right:Ne \HyPL@Labels { \the\Hy@abspage<<#1>> }
599         \__hyp_PageLabels_gpush:
600       }
601   }

```

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

10 Core Hyperref Commands

Every hyperref has to define eight core command:

```

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

```

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

10.1 Link level

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

```

g__hyp_linknestlevel_int
602 \int_new:N \g__hyp_linknestlevel_int

```

(End of definition for `g__hyp_linknestlevel_int`.)

```

603 \prg_new_conditional:Npnn \__hyp_if_outer_link: {TF}
604 {
605   \int_compare:nNnTF { \g__hyp_linknestlevel_int } > {1}
606   { \prg_return_false: }
607   { \prg_return_true: }
608 }
609 \cs_new:Npn \__hyp_check_link_nesting:TF #1 #2
610 {
611   \use_i:nn {#1}{#2}
612 }
613 \keys_define:nn { hyp }
614 {
615   nested-links .choice:,
616   nested-links / true .code:n =
617   { \cs_set_eq:NN \__hyp_check_link_nesting:TF \use_i:nn },
618   nested-links / false .code:n =
619   { \cs_set_eq:NN \__hyp_check_link_nesting:TF \__hyp_if_outer_link:TF },
620   nested-links .default:n = {true}
621 }

```

10.2 Anchors / destinations

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

```

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

```

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

```

\__hyp_destination:nn
622 \Hy@WrapperDef \__hyp_destination:nn #1 #2
623 {
624   \mode_if_horizontal:T { \@savs\spacefactor }
625   \Hy@SaveLastskip      %defined in hyperref
626   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
627   \__hyp_text_pdfstring:eoN
628   { \HyperDestNameFilter{#1} }
629   { \l__hyp_text_enc_dest_tl }
630   \l__hyp_tmpa_tl
631   \str_if_eq:nnTF {#2} {fitrbx}
632   {
633     \exp_args:NV
634     \pdf_destination:nnnn \l__hyp_tmpa_tl
635     { \box_wd:N \l__hyp_dest_box }
636     { \box_ht:N \l__hyp_dest_box }
637     { \box_dp:N \l__hyp_dest_box }
638   }
639   {

```



```

640     \exp_args:NV
641     \pdf_destination:nf
642     { \l__hyp_tmpa_tl }
643     { #2 }
644   }
645   \Hy@RestoreLastskip   %defined in hyperref
646   \mode_if_horizontal:T { \spacefactor\@savsf }
647 }

```

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

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

```

\hyper@anchor
\hyper@anchorstart
\hyper@anchorend
hyp/anchor
\l_hyp_current_dest_name_tl
648 \tl_new:N\l_hyp_current_dest_name_tl
649 \hook_new:n{hyp/anchor}
650 \cs_new_protected:Npn \hyper@anchor #1
651 {
652   \exp_args:NnV
653   \__hyp_destination:nn {#1} \l_hyp_dest_pdfview_tl
654   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
655   \hook_use:n{hyp/anchor}
656 }
657
658 \cs_new_protected:Npn \hyper@anchorstart #1
659 {
660   \Hy@activeanchortrue
661   \exp_args:NnV
662   \__hyp_destination:nn {#1} \l_hyp_dest_pdfview_tl
663   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
664   \hook_use:n{hyp/anchor}
665 }
666
667 \cs_new_protected:Npn \hyper@anchorend
668 {
669   \Hy@activeanchorfalse
670 }

```

(End of definition for `\hyper@anchor` and others. These functions are documented on page ??.)

10.3 GoTo Links

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

At first the internal link commands:

```

671 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
672 {
673   \mode_leave_vertical:

```

```

674 \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
675 \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
676 {
677   \msg_warning:nne
678   { hyp }
679   { empty-destination-name }
680   { \c__hyp_dest_undefined_tl }
681   \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
682 }
683 {
684   \__hyp_text_pdfstring:eoN
685   { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
686   { \l__hyp_text_enc_dest_tl }
687   \l__hyp_dest_name_tmpa_tl
688 }
689 \exp_args:No
690 \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
691 }
692
693 \cs_new_protected:Npn \__hyp_link_goto_end:
694 {
695   \pdfannot_link_goto_end:
696 }

```

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. We store again the destination name in the public `tl \l_hyp_current_dest_name_tl` so that the hook code can make use of it

```

697
698 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
699 {
700   \bool_if:NTF \l__hyp_annot_GoTo_bool
701   {

```

```

702         \int_gincr:N\g__hyp_linknestlevel_int
703         \__hyp_check_link_nesting:TF
704         {
705             \Hy@VerboseLinkStart{#1}{#2}
706             \group_begin:
707             \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
this socket defines \l__hyp_link_Contents_tl
708             \socket_use:nnn{hyp/link/GoTo/Contents}{#2}{#3}
709             \pdfannot_dict_put:nne {link/GoTo}{Contents}
710             {\l__hyp_link_Contents_tl}
711             \hook_use:n {hyp/link/#1}
712             \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
713             \__hyp_link_goto_end:
714             \group_end:
715             \Hy@VerboseLinkStop
716         }
717         {
718             \group_begin: #3\group_end:
719         }
720         \int_gdecr:N\g__hyp_linknestlevel_int
721     }
722     {{\let\protect\relax#3}}
723 }
724 \cs_new_protected:Npn \hyper@linkstart #1 #2 %#1 context, #2=destination name
725 {
726     \bool_if:NT \l__hyp_annot_GoTo_bool
727     {
728         \int_gincr:N\g__hyp_linknestlevel_int
729         \__hyp_check_link_nesting:TF
730         {
731             \Hy@VerboseLinkStart{#1}{#2}% only for debug
732             \group_begin:
733             \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
734             \hook_use:n {hyp/link/#1}
735             \__hyp_link_goto_begin:nw {#2}
736         }
737         {
738             \group_begin:
739         }
740     }
741 }
742
743 \cs_new_protected:Npn \hyper@linkend
744 {
745     \bool_if:NT \l__hyp_annot_GoTo_bool
746     {
747         \__hyp_check_link_nesting:TF
748         {
749             \__hyp_link_goto_end:
750             \group_end:
751             \Hy@VerboseLinkStop
752         }
753         {
754             \group_end:

```

```

755     }
756     \int_gdecr:N\g__hyp_linknestlevel_int
757   }
758 }

```

10.4 URI links

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

```

759 \pdfdict_new:n {l_hyp/annot/A/URI}
760 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
761 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
762
763 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
764 {
765   \bool_if:NTF \l__hyp_annot_URI_bool
766   {
767     \int_gincr:N\g__hyp_linknestlevel_int
768     \__hyp_check_link_nesting:TF
769     {
770       \group_begin:
771       \__hyp_text_pdfstring:eoN
772       { #2}
773       { \l__hyp_text_enc_uri_print_tl }
774       \l__hyp_uri_tmpa_tl
775       \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
776       \bool_if:NT \l__hyp_href_url_ismap_bool
777       {
778         \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
779       }

```

This socket defines \l__hyp_link_Contents_tl

```

780   \socket_use:nn{hyp/link/URI/Contents}{#2}
781   \pdfannot_dict_put:nne {link/URI}{Contents}{\l__hyp_link_Contents_tl}
782   \cs_set_eq:NN \# \c_hash_str
783   \cs_set_eq:NN \% \c_percent_str
784   \Hy@safe@activetrue
785   \mode_leave_vertical:
786   \pdfannot_dict_put:nne {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
787   \pdfannot_link:nen { URI }
788   {
789   }
790   {
791     \let\protect\relax
792     #1
793     \Hy@xspace@end
794     \Hy@VerboseLinkStop %where is the start??
795   }
796   \group_end:
797 }
798 {
799   \group_begin: #1 \group_end:
800 }
801 \int_gdecr:N\g__hyp_linknestlevel_int
802 }

```

```

803     {\let\protect\relax#1}}
804   }
805

```

10.5 GoToR Links files

```

806 \pdfdict_new:n {l_hyp/annot/A/GoToR}
807 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
808 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
809
810 \cs_generate_variant:Nn \pdffile_embed_file:nnn {noe}
811 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
812 {
813   \bool_if:NTF \l__hyp_annot_GoToR_bool
814   {
815     \int_gincr:N\g__hyp_linknestlevel_int
816     \__hyp_check_link_nesting:TF
817     {
818       \group_begin:
819       \tl_set:Nx \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
820       \exp_args:Nx
821       \pdf_object_if_exist:nF { __hyp_file\__tl_to_str:N \l__hyp_filename_tmpa_tl }
822       {
823         \pdfdict_put:nne { l_pdffile/Filespec } { Subtype } { \pdf_name_from_unicode_e:n
824         \pdffile_embed_file:noe
825         {}
826         { \l__hyp_filename_tmpa_tl }
827         { __hyp_file\__tl_to_str:N \l__hyp_filename_tmpa_tl }
828       }
829       \pdfdict_put:nne
830       { l_hyp/annot/A/GoToR }
831       { F }
832       { \pdf_object_ref:e { __hyp_file\__tl_to_str:N \l__hyp_filename_tmpa_tl } }
833       \__hyp_text_pdfstring:nnN
834       { #3 }
835       { \l__hyp_text_enc_dest_print_tl }
836       \l__hyp_dest_name_tmpa_tl
837       \tl_if_blank:eTF { #3 }
838       {
839         \pdfdict_put:nne { l_hyp/annot/A/GoToR } { D }
840         {
841           [
842             \int_eval:n
843             { \int_max:nn { 0 } { 0 \l__hyp_href_pdf_page_tl - 1 } }
844             / \l__hyp_dest_pdfremotestartview_tl
845           ]
846         }
847       }
848       {
849         \pdfdict_put:nno { l_hyp/annot/A/GoToR } { D } { \l__hyp_dest_name_tmpa_tl }
850       }
851       \mode_leave_vertical:

```

We use an extra object here, as ghostscript doesn't like the object reference in the dict

<https://chat.stackexchange.com/transcript/message/57361080#57361080>

```
852         \pdf_object_unnamed_write:ne{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
853         \pdfannot_dict_put:nne {link/GoToR}{A}{\pdf_object_ref_last:}
854         \pdfannot_link:nnn %expansion??
855         { GoToR }
856         {
857         }
858         {
859         \let\protect\relax
860         #1\Hy@xspace@end
861         \Hy@VerboseLinkStop %where is the start??
862         }
863     \group_end:
864 }
865 {
866     \group_begin: #1 \group_end:
867 }
868 \int_gdecr:N\g__hyp_linknestlevel_int
869 }
870 {\let\protect\relax#1}}
871 }
```

10.6 Launch links

We define `\hyper@linklaunch` for naming consistency

```
872 \pdfdict_new:n {l_hyp/annot/A/Launch}
873 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
874 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
875
876 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
877 {
878     \bool_if:NTF \l__hyp_annot_Launch_bool
879     {
880         \int_gincr:N\g__hyp_linknestlevel_int
881         \__hyp_check_link_nesting:TF
882         {
883             \group_begin:
884             \__hyp_text_pdfstring:nnN
885             { #1 }
886             { \l__hyp_text_enc_file_print_tl }
887             \l__hyp_filename_tmpa_tl
888             \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
889             \__hyp_text_pdfstring:noN
890             { #3 }
891             { \l__hyp_text_enc_para_print_tl }
892             \l__hyp_para_tmpa_tl
893             \bool_if:NTF
894             {
895                 \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}{}
896                 ||
897                 \pdf_version_compare_p:Nn > {1.9}
898             }
899             {
900                 \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
901             }
902         }
903     }
904 }
```

```

902         {
903             \pdfdict_put:nne
904             {l_hyp/annot/A/Launch}
905             {Win}
906             {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
907         }
908         \mode_leave_vertical:
909         \pdfannot_dict_put:nne {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}
910         \pdfannot_link:nen
911         { Launch }
912         {
913             % /A
914             % <<
915             % \pdfdict_use:n {l_hyp/annot/A/Launch}
916             % >>
917         }
918         {
919             \let\protect\relax
920             #2\Hy@xspace@end
921             \Hy@VerboseLinkStop %where is the start??
922         }
923         \group_end:
924     }
925     { \group_begin: #2 \group_end: }
926     \int_gdecr:N\g__hyp_linknestlevel_int
927 }
928 {{\let\protect\relax#2}}
929 }

```

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

```

930 \use:e
931 { % filename, anchor text, linkname
932   \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str #1 \exp_not:N \ \ #2 #3
933 }
934 {
935   \hyper@linklaunch {#1}{#2}{#3}
936 }

```

10.7 Named links (menu)

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

```

937 \pdfdict_new:n {l_hyp/annot/A/Named}
938 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
939 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
940
941 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
942 {
943   \bool_if:NTF \l__hyp_annot_Named_bool
944   {
945     \int_gincr:N\g__hyp_linknestlevel_int
946     \__hyp_check_link_nesting:TF
947     {
948       \group_begin:

```

```

949         \pdfmeta_standard_verify:nnTF {named_actions}{#1}
950         {
951             \mode_leave_vertical:
952             \pdfdict_put:nne {l_hyp/annot/A/Named}{N}
953             {\pdf_name_from_unicode_e:n{#1}}
954             \pdfannot_dict_put:nne {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named
955             \pdfannot_link:nnn { Named }
956             {
957                 % /A
958                 % <<
959                 % \pdfdict_use:n { l_hyp/annot/A/Named }
960                 % >>
961             }
962             {
963                 #2
964                 \Hy@xspace@end
965                 \Hy@VerboseLinkStop
966             }
967         }
968         {
969             \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
970             #2
971         }
972         \group_end:
973     }
974     { \group_begin: #2 \group_end: }
975     \int_gdecr:N\g__hyp_linknestlevel_int
976 }
977 {\let\protect\relax#2}}
978 }
979

```

11 Link decorations

11.1 Functions to export and select colors

We support two input syntax: color expressions and model with values. Exporting can be done by first setting the color with `__hyp_color_set:nn` (if needed to a temporary color name) and then using `\color_export:nnN`. But we need a variant as the export format `space-sep-cmyk` or `space-sep-rgb` is stored in a tl.

```

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

```

```

\__hyp_color_select:n \__hyp_color_select:n {\color}

```

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

```

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

```

```

981 \cs_new_protected:Npn \__hyp_color_select:n #1
982 {
983     \tl_if_head_eq_charcode:nNTF {#1}[ %]

```



```

984     {
985       \__hyp_color_select_aux:wn #1
986     }
987     {
988       \color_select:n {#1}
989     }
990   }
991
992   \cs_new_protected:Npn \__hyp_color_select_aux:wn [#1] #2
993   {
994     \color_select:nn {#1}{#2}
995   }
996
997   \cs_generate_variant:Nn \__hyp_color_select:n {e}

```

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

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

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

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

```

998   \cs_new_protected:Npn \__hyp_color_set:nn #1 #2
999   {
1000     \tl_if_head_eq_charcode:nNTF {#2}[ %]
1001     {
1002       \__hyp_color_set_aux:nwn { #1 } #2
1003     }
1004     {
1005       \color_set:nn {#1} {#2}
1006     }
1007   }
1008
1009   \cs_new_protected:Npn \__hyp_color_set_aux:nwn #1 [#2] #3
1010   {
1011     \color_set:nnn {#1}{#2}{#3}
1012   }
1013
1014   \cs_generate_variant:Nn \__hyp_color_set:nn {ne}

```

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

11.2 Textcolor of links

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

```

1015   \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1016   {
1017     \hook_gput_code:nnn
1018     {pdfannot/link/#2/begin}
1019     {hyp/color}

```

```

1020     {
1021         \bool_if:cT { l_hyp_annot_color#1_bool }
1022         {
1023             \group_begin:
1024             \color_select:n { hyp/color/#1}
1025         }
1026     }
1027     \hook_gput_code:nnn
1028     {pdfannot/link/#2/end}
1029     {hyp/color}
1030     {
1031         \bool_if:cT { l_hyp_annot_color#1_bool }
1032         {
1033             \group_end:
1034         }
1035     }
1036 }

```

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

```

1037 \keys_define:nn { hyp }
1038 {
1039     ,colorlinks .meta:n =
1040     {
1041         ,pdfborder={0~0~0}
1042         ,pdfborderstyle=
1043         ,colorurl   =#1
1044         ,colorlink  =#1
1045         ,colorrun   =#1
1046         ,colormenu  =#1
1047         ,colorfile  =#1
1048     }
1049     ,colorlinks .default:n = {true}
1050 }

```

```

colorurl (setup key)
colorlink (setup key) 1051 \seq_map_inline:Nn \c__hyp_annot_types_seq
colorrun (setup key) 1052 {
colormenu (setup key) 1053     \keys_define:nn { hyp }
colorfile (setup key) 1054     {
urlcolor (setup key) 1055         ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
linkcolor (setup key) 1056         ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
runcolor (setup key) 1057     }
menucolor (setup key) 1058 }
filecolor (setup key) 1059
allcolors (setup key) 1060 \keys_define:nn { hyp }
1061 {
1062     ,allcolors .meta:n =
1063     {
1064         ,urlcolor=#1
1065         ,linkcolor=#1
1066         ,runcolor=#1
1067         ,filecolor=#1
1068         ,menucolor=#1
1069     }

```

```

1070     ,allcolors .value_required:n = true
1071 }

```

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*)

```

1072 \keys_define:nn { hyp }
1073 {
1074     ,bordercolormodel .choices:nn =
1075     {rgb,cmyk}
1076     { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
1077     ,bordercolormodel .initial:n ={rgb}
1078 }

1079 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1080 {
1081     \keys_define:nn { hyp }
1082     {
1083         #1bordercolor .code:n =
1084         {
1085             \tl_if_empty:nTF { ##1 }
1086             {
1087                 \pdfannot_dict_remove:nn
1088                 {link/#2}
1089                 { C }
1090             }
1091             {
1092                 \__hyp_color_set:ne {hyp/color/#1border}{##1}
1093                 \color_export:nVN
1094                 {hyp/color/#1border}
1095                 \g__hyp_bordercolormodel_str
1096                 \l__hyp_tmpa_tl
1097                 \pdfannot_dict_put:nne
1098                 {link/#2}
1099                 { C }
1100                 { [\l__hyp_tmpa_tl] }
1101             }
1102         }
1103     }
1104 }

1105
1106 \keys_define:nn { hyp }
1107 {
1108     ,allbordercolors .meta:n =
1109     {
1110         ,linkbordercolor=#1
1111         ,urlbordercolor =#1
1112         ,filebordercolor=#1
1113         ,menubordercolor=#1
1114         ,runbordercolor =#1

```

```

1115     }
1116     ,allbordercolors .value_required:n = true
1117 }
1118

```

11.3.2 Borderwidth and -arc

```

1119 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1120 {
1121   \keys_define:nn { hyp }
1122   {
1123     #1border .code:n =
1124     {
1125       \tl_if_empty:nTF { ##1 }
1126       {
1127         \pdfannot_dict_remove:nn
1128         {link/#2}
1129         { Border }
1130       }
1131       {
1132         \pdfannot_dict_put:nnn
1133         {link/#2}
1134         { Border }
1135         { [##1] }
1136       }
1137     }
1138   }
1139 }
1140 \keys_define:nn { hyp }
1141 {
1142   ,pdfborder .code:n =
1143   {
1144     \tl_if_empty:nTF { #1 }
1145     {
1146       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1147       {
1148         \pdfannot_dict_remove:nn
1149         {link/##2}
1150         { Border }
1151       }
1152     }
1153     {
1154       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1155       {
1156         \pdfannot_dict_put:nnn
1157         {link/##2}
1158         { Border }
1159         { [#1] }
1160       }
1161     }
1162   }
1163   ,pdfborder .initial:n = {0~0~1},
1164 }

```

11.3.3 Borderstyle

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

```
pdfborderstyle (setup key)
urlborderstyle (setup key) 1165 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1166 {
runborderstyle (setup key) 1167   \keys_define:nn { hyp }
fileborderstyle (setup key) 1168   {
menuborderstyle (setup key) 1169     #1borderstyle .code:n =
1170     {
1171       \tl_if_empty:nTF { ##1 }
1172       {
1173         \pdfannot_dict_remove:nn
1174         {link/##2}
1175         { BS }
1176       }
1177       {
1178         \pdfannot_dict_put:nnn
1179         {link/##2}
1180         { BS }
1181         { <<##1>> }
1182       }
1183     }
1184   }
1185 }
1186 \keys_define:nn { hyp }
1187 {
1188   ,pdfborderstyle .code:n =
1189   {
1190     \tl_if_empty:nTF { #1 }
1191     {
1192       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1193       {
1194         \pdfannot_dict_remove:nn
1195         {link/##2}
1196         { BS }
1197       }
1198     }
1199     {
1200       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1201       {
1202         \pdfannot_dict_put:nnn
1203         {link/##2}
1204         { BS }
1205         { <<##1>> }
1206       }
1207     }
1208   }
1209   ,pdfborderstyle .initial:n = {},
1210 }
```

11.4 ocgcolorlinks

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

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

```
1211 \cs_new_protected:Npn \__hyp_ocg_init:
1212 {
1213   \pdf_object_new:n { __hyp/OCG/View }
1214   \pdf_object_new:n { __hyp/OCG/Print }
1215   \pdf_object_new:n { __hyp/OCG/config }
1216   \pdf_object_new:n { __hyp/OCG/refarray }
1217   \pdf_object_write:nne { __hyp/OCG/refarray } { array }
1218   {
1219     \pdf_object_ref:n { __hyp/OCG/View }
1220     \c_space_tl
1221     \pdf_object_ref:n { __hyp/OCG/Print }
1222   }
1223   \pdf_object_write:nnn { __hyp/OCG/View } { dict }
1224   {
1225     /Type/OCG
1226     /Name(View)
1227     /Usage
1228     <<
1229       /Print <</PrintState/OFF>>~
1230       /View <</ViewState/ON >>~
1231     >>
1232   }
1233   \pdf_object_write:nnn { __hyp/OCG/Print } { dict }
1234   {
1235     /Type/OCG
1236     /Name(Print)
1237     /Usage
1238     <<
1239       /Print <</PrintState/ON>>~
1240       /View <</ViewState/OFF>>~
1241     >>
1242   }
1243   \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/View } }
1244   \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/Print } }
1245   \pdf_object_write:nne { __hyp/OCG/config } { dict }
1246   {
1247     /OFF[\pdf_object_ref:n { __hyp/OCG/Print } ]
1248     /AS[
1249       <<
1250         /Event/View
1251         /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1252         /Category[/View]
1253       >>
1254       <<
1255         /Event/Print
1256         /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1257         /Category[/Print]
```

```

1258         >>
1259         <<
1260         /Event/Export
1261         /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1262         /Category[/Print]
1263         >>
1264     ]
1265 }
1266 \pdfmanagement_add:nne { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG
1267 \cs_gset:Npn \__hyp_ocg_init: {}
1268 }

```

(End of definition for __hyp_ocg_init:.)

We use like with colors a hook, this allows ocgx to replace it. The implementation is rather simple and uses a box.

```

1269 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1270 {
1271     \hook_gput_code:nnn
1272     {pdfannot/link/#2/begin}
1273     {hyp/ocg}
1274     {
1275         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1276         {
1277             \__hyp_ocg_init:
1278             \group_begin:
1279             \hbox_set:Nw \l__hyp_tmpa_box
1280         }
1281     }
1282     \hook_gput_code:nnn
1283     {pdfannot/link/#2/end}
1284     {hyp/ocg}
1285     {
1286         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1287         {
1288             \hbox_set_end:
1289             \mbox
1290             {
1291                 \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1292                 \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1293                 \pdf_emc:
1294                 \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1295                 \group_begin:
1296                 \color_select:n { hyp/color/#1 }
1297                 \box_use_drop:N \l__hyp_tmpa_box
1298                 \group_end:
1299                 \pdf_emc:
1300             }
1301             \group_end:
1302         }
1303     }
1304 }

```

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

ocgcolorlink (setup key) 1305 \bool_lazy_or:nnTF

ocgcolorurl (setup key)

ocgcolorfile (setup key)

ocgcolormenu (setup key)

ocgcolorrun (setup key)

```

1306 { \pdf_version_compare_p:Nn > {1.4} }
1307 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1308 {
1309   \keys_define:nn { hyp }
1310   {
1311     ,_ocgcolorlinks .meta:n =
1312     {
1313       ocgcolorlink=#1,
1314       ocgcolorurl=#1,
1315       ocgcolorfile=#1,
1316       ocgcolorrun=#1,
1317       ocgcolormenu=#1
1318     }
1319     ,_ocgcolorlinks .default:n = true
1320   }
1321 }
1322 {
1323   \keys_define:nn { hyp }
1324   {
1325     ,_ocgcolorlinks .code:n =
1326     {
1327       \msg_warning:nnee
1328       { hyp }
1329       { ignore-deprecated-or-unknown-option-in-pdf-version }
1330       { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1331     }
1332   }
1333 }
1334
1335 \keys_define:nn { hyp }
1336 {
1337   ,ocgcolorlinks .choice:
1338   ,ocgcolorlinks / true .meta:n =
1339   {
1340     pdfborder      = {0~0~0},
1341     pdfborderstyle = {},
1342     colorlinks     = false,
1343     _ocgcolorlinks = true
1344   }
1345   ,ocgcolorlinks / false .meta:n =
1346   {
1347     _ocgcolorlinks = false
1348   }
1349   ,ocgcolorlinks .default:n = {true}
1350 }
1351
1352 \seq_map_inline:Nn \c__hyp_annot_types_seq
1353 {
1354   \bool_lazy_or:nnTF
1355   { \pdf_version_compare_p:Nn > {1.4} }
1356   { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1357   {
1358     \keys_define:nn { hyp }
1359     {

```



```

1360         ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1361     }
1362 }
1363 {
1364     \keys_define:nn { hyp }
1365     {
1366         ,ocgcolor#1 .code:n=
1367         {
1368             \msg_warning:nnee
1369             { hyp }
1370             { ignore-deprecated-or-unknown-option-in-pdf-version }
1371             { ocgcolor#1 }
1372             { \pdf_version_major:.\pdf_version_minor: }
1373         }
1374     }
1375 }
1376 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1377 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1378 {
1379     \keys_define:nn { hyp }
1380     {
1381         ,#1highlight .choices:nn =
1382         { /I, /N, /O, /P }
1383         {
1384             \pdfannot_dict_put:nnn
1385             { link/#2 }
1386             { H }
1387             { ##1 }
1388         }
1389     }
1390     ,#1highlight / .code:n =
1391     {
1392         \pdfannot_dict_remove:nn
1393         { link/#2 }
1394         { H }
1395     }
1396 }
1397 ,#1highlight / unknown .code:n =
1398 {
1399     \msg_warning:nneee { hyp } { unknown-choice+empty }
1400     { #1highlight }
1401     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1402     { \exp_not:n {##1} }
1403 }
1404 }
1405 }
1406
1407 \keys_define:nn { hyp }
1408 {
1409

```

```

1410 ,pdfhighlight .choices:nn =
1411 { /I, /N, /O, /P}
1412 {
1413   \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1414   {
1415     \pdfannot_dict_put:nnn
1416     {link/####2}
1417     { H }
1418     { #1 }
1419   }
1420 }
1421 ,pdfhighlight / .code:n =
1422 {
1423   \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1424   {
1425     \pdfannot_dict_remove:nn
1426     {link/##2}
1427     { H }
1428   }
1429 }
1430 ,pdfhighlight .initial:n = {/I},
1431 ,pdfhighlight / unknown .code:n =
1432 {
1433   \msg_warning:nneee { hyp } { unknown-choice+empty }
1434   { pdfhighlight }
1435   { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1436   { \exp_not:n {#1} }
1437 }
1438 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1439 \keys_define:nn { hyp }
hideurl (setup key) 1440 {
hidefile (setup key) 1441   hidelinks .meta:n =
hiderun (setup key) 1442   {
hidemenu (setup key) 1443     ,colorlinks      = false
1444     ,ocgcolorlinks = false
1445     ,pdfborder      = { 0~0~0 }
1446     ,pdfborderstyle=
1447   }
1448 }
1449
1450 \seq_map_inline:Nn \c__hyp_annot_types_seq
1451 {
1452   \keys_define:nn { hyp }
1453   {
1454     hide#1 .meta:n =
1455     {
1456       ,color#1      = false
1457       ,ocgcolor#1   = false

```

```

1458             ,#1border      = { 0~0~0 }
1459             ,#1borderstyle =
1460         }
1461     }
1462 }

```

11.7 color schemes and settings

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

`colorscheme (setup key)`

```

1463 \keys_define:nn { hyp }
1464 {
1465     colorscheme .code:n =
1466     {
1467         \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1468         {
1469             \keys_set:nn { hyp }
1470             {
1471                 ##1 = ##2
1472             }
1473         }
1474     }
1475 }
1476 \keys_set:nn { hyp } {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)`

```

1477 \keys_define:nn { hyp }
1478 {
1479     ,unicode      .code:n = {}
1480     ,pdfencoding  .code:n = {}
1481     ,pdfversion   .code:n =
1482     {
1483         \msg_warning:nn { hyp } { pdfversion-disabled }
1484     }
1485 }
1486 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

`verbose (setup key)`

`debug (setup key)`

`draft (setup key)`

`final (setup key)`

```

1487 \keys_define:nn { hyp }
1488 {
1489     ,verbose      .legacy_if_set:n = {Hy@verbose}
1490     ,debug        .legacy_if_set:n = {Hy@verbose}

```

```

1491 }
1492 \keys_define:nn { hyp }
1493 {
1494   ,draft .code:n =
1495   {
1496     \Hy@drafttrue
1497     \PassOptionsToPackage{draft}{bookmark}
1498   }
1499   ,final .code:n =
1500   {
1501     \Hy@finaltrue
1502     \PassOptionsToPackage{final}{bookmark}
1503   }
1504 }

extension (setup key)
hypertextnames (setup key) 1505 \keys_define:nn { hyp }
naturalnames (setup key) 1506 {
pageanchor (setup key) 1507   ,extension .tl_set:N = \XR@ext
linktoc (setup key) 1508   ,extension .initial:n= pdf
linktocpage (setup key) 1509   ,hypertextnames .legacy_if_set:n = {Hy@hypertextnames}
plainpages (setup key) 1510   ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
localanchorname (setup key) 1511   ,localanchorname .legacy_if_set:n = {Hy@localanchorname}
linkfileprefix (setup key) 1512   ,naturalnames .legacy_if_set:n = {Hy@naturalnames}
1513   ,pageanchor .legacy_if_set:n = {Hy@pageanchor}
1514   ,plainpages .legacy_if_set:n = {Hy@plainpages}
1515 }
1516
1517 \keys_define:nn { hyp }
1518 {
1519   ,linktoc .choices:nn = { none, section, all, page }
1520   {
1521     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1522   }
1523   ,linktoc / unknown .code:n =
1524   {
1525     \msg_warning:nneee { hyp } { unknown-choice }
1526     { linktoc }
1527     { none, section, all, page }
1528     { \exp_not:n {#1} }
1529   }
1530   ,linktocpage .choice:
1531   ,linktocpage / true .meta:n = {linktoc=page}
1532   ,linktocpage / false .meta:n = {linktoc=section}
1533   ,linktocpage .default:n = true
1534 }
1535

link (setup key) This booleans allow to disable the link types.
url (setup key) 1536 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1537 {
menu (setup key) 1538   \keys_define:nn { hyp }
run (setup key) 1539   {
1540     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}

```

```

1541     }
1542 }

1543 \keys_define:nn { hyp }
1544 {
1545   ,baseurl .code:n =
1546   {
1547     \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1548     \tl_if_empty:NTF \l__hyp_tmpa_tl
1549     {
1550       \pdfmanagement_remove:nn {Catalog} { URI }
1551     }
1552     {
1553       \pdfmanagement_add:nne {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1554     }
1555     \__hyp_store_metadata:nn {baseurl}{#1}
1556   }
1557   %only false does something ...
1558   ,bookmarks .choice:
1559   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1560   ,bookmarks / true .code:n = {}
1561   ,bookmarks .default:n = {true}
1562   ,bookmarksnumbered .legacy_if_set:n = {Hy@bookmarksnumbered}
1563   ,bookmarksopen .legacy_if_set:n = {Hy@bookmarksopen}
1564   ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1565   ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1566   ,pdfcenterwindow .choice:
1567   ,pdfcenterwindow / false .code:n =
1568   {
1569     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1570   }
1571   ,pdfcenterwindow / true .code:n =
1572   {
1573     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1574   }
1575   ,pdfcenterwindow / .code:n =
1576   {
1577     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1578   }
1579   ,pdfcenterwindow / unknown .code:n =
1580   {
1581     \msg_warning:nnee { hyp } { no-bool }
1582     { pdfcenterwindow }
1583     { \exp_not:n {#1} }
1584   }
1585   ,pdfcenterwindow .default:n = true
1586   ,pdffdirection .choice:
1587   ,pdffdirection / L2R .code:n =
1588   {
1589     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1590   }
1591   ,pdffdirection / R2L .code:n =
1592   {
1593     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }

```

```

1594     }
1595 ,pdfdirection / .code:n =
1596 {
1597     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1598 }
1599 ,pdfdirection / unknown .code:n =
1600 {
1601     \msg_warning:nneee { hyp } { unknown-choice+empty }
1602     { pdfdirection }
1603     { L2R , R2L }
1604     { \exp_not:n {#1} }
1605 }
1606 ,pdffitwindow .choice:
1607 ,pdffitwindow / false .code:n =
1608 {
1609     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1610 }
1611 ,pdffitwindow / true .code:n =
1612 {
1613     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1614 }
1615 ,pdffitwindow .default:n = true
1616 ,pdfduplex .choices:nn =
1617 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1618 {
1619     \pdf_version_compare:NnTF > {1.6}
1620     {
1621         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1622         { PrintDuplex } { /#1 }
1623     }
1624     {
1625         \msg_warning:nnee
1626         {hyp}
1627         {ignore-deprecated-or-unknown-option-in-pdf-version}
1628         {pdfduplex}
1629         {\pdf_version:}
1630     }
1631 }%
1632 ,pdfduplex / .code:n =
1633 {
1634     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1635 }
1636 ,pdfduplex / unknown .code:n =
1637 {
1638     \msg_warning:nneee { hyp } { unknown-choice+empty }
1639     { pdfduplex }
1640     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1641     { \exp_not:n {#1} }
1642 }
1643 ,pdffitwindow .choice:
1644 ,pdffitwindow / false .code:n =
1645 {
1646     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1647 }

```

```

1648 ,pdffitwindow / true .code:n =
1649 {
1650   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1651 }
1652 ,pdffitwindow / .code:n =
1653 {
1654   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1655 }
1656 ,pdffitwindow .default:n = true
1657 ,pdffitwindow / unknown .code:n =
1658 {
1659   \msg_warning:nnee { hyp } { no-bool }
1660   { pdffitwindow }
1661   { \exp_not:n {#1} }
1662 }
1663 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1664 ,pdflinkmargin .initial:n = {1pt}
1665 ,pdfmenubar .choice:
1666 ,pdfmenubar / true .code:n =
1667 {
1668   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1669 }
1670 ,pdfmenubar / false .code:n =
1671 {
1672   \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1673   { HideMenubar } { true }
1674 }
1675 ,pdfmenubar / .code:n =
1676 {
1677   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1678 }
1679 ,pdfmenubar .default:n = true
1680 ,pdfmenubar / unknown .code:n =
1681 {
1682   \msg_warning:nnee { hyp } { no-bool }
1683   { pdfmenubar }
1684   { \exp_not:n {#1} }
1685 }
1686 ,pdfnewwindow .choice:
1687 ,pdfnewwindow / true .code:n =
1688 {
1689   \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1690   \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1691 }
1692 ,pdfnewwindow / false .code:n =
1693 {
1694   \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1695   \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1696 }
1697 ,pdfnewwindow / .code:n =
1698 {
1699   \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1700   \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1701 }

```

```

1702 ,pdfnonfullscreenpagemode .choices:nn =
1703 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1704 {
1705   \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1706   { NonFullScreenPageMode} {/#1}
1707 }
1708 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1709 {
1710   \pdf_version_compare:NnTF < {1.6}
1711   {
1712     %message
1713   }
1714   {
1715     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1716     {NonFullScreenPageMode}{/UseAttachments}
1717   }
1718 }
1719 ,pdfnonfullscreenpagemode / .code:n =
1720 {
1721   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1722 }
1723 ,pdfnonfullscreenpagemode / unknown .code:n =
1724 {
1725   \msg_warning:nneee { hyp } { unknown-choice+empty }
1726   { pdfnonfullscreenpagemode }
1727   { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1728   { \exp_not:n {#1} }
1729 }
1730 ,pdfnumcopies .code:n =
1731 {
1732   \pdf_version_compare:NnTF > {1.6}
1733   {
1734     \tl_if_empty:nTF {#1}
1735     {
1736       \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1737     }
1738     {
1739       \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1740       {NumCopies}{#1}
1741     }
1742   }
1743   {
1744     \msg_warning:nnee
1745     {hyp}
1746     {ignore-deprecated-or-unknown-option-in-pdf-version}
1747     {pdfnumcopies}
1748     {\pdf_version:}
1749   }
1750 }
1751 ,pdfpagelayout .choices:nn =
1752 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1753 { \pdfmanagement_add:nne {Catalog} { PageLayout }{ /#1 } }
1754 ,pdfpagelayout / .code:n =
1755 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }

```



```

1756 ,pdfpagelayout / unknown .code:n =
1757 {
1758     \msg_warning:nneee { hyp } { unknown-choice+empty }
1759     { pdfpagelayout }
1760     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1761     { \exp_not:n {#1} }
1762 }
1763 ,pdfpagemode .choices:nn =
1764 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1765 { \pdfmanagement_add:nne {Catalog} { PageMode }{ /#1 } }
1766 ,pdfpagemode / UseAttachments .code:n =
1767 {
1768     \pdf_version_compare:NnTF > {1.5}
1769     {
1770         \pdfmanagement_add:nne {Catalog} { PageMode }{ /UseAttachments }
1771     }
1772     {
1773         \msg_warning:nnee
1774         {hyp}
1775         {ignore-deprecated-or-unknown-value-in-pdf-version}
1776         {UseAttachments}
1777         {\pdf_version:}
1778     }
1779 }
1780 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1781 ,pdfpagemode / unknown .code:n =
1782 {
1783     \msg_warning:nneee { hyp } { unknown-choice+empty }
1784     { pdfpagemode }
1785     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1786     { \exp_not:n {#1} }
1787 }
1788 ,pdfpagescrop .code:n =
1789 {
1790     \tl_if_empty:nTF {#1} %or blank?
1791     {
1792         \pdfmanagement_remove:nn {Pages} { CropBox }
1793     }
1794     {
1795         \pdfmanagement_add:nne {Pages} { CropBox } { [#1] }
1796     }
1797 }
1798 ,pdfpicktraybypdfsize .choice:
1799 ,pdfpicktraybypdfsize / true .code:n =
1800 {
1801     \pdf_version_compare:NnTF > {1.6}
1802     {
1803         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1804         { PickTrayByPDFSize } { true }
1805     }
1806     {
1807         \msg_warning:nnee
1808         {hyp}
1809         {ignore-deprecated-or-unknown-option-in-pdf-version}

```

```

1810         {pdfpicktraybypdfsize}
1811         {\pdf_version:}
1812     }
1813 }
1814 ,pdfpicktraybypdfsize / false .code:n =
1815 {
1816     \pdf_version_compare:NnTF > {1.6}
1817     {
1818         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1819         { PickTrayByPDFSize } { false }
1820     }
1821     {
1822         \msg_warning:nnee
1823         {hyp}
1824         {ignore-deprecated-or-unknown-option-in-pdf-version}
1825         {pdfpicktraybypdfsize}
1826         {\pdf_version:}
1827     }
1828 }
1829 ,pdfpicktraybypdfsize / .code:n =
1830 {
1831     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1832 }
1833 ,pdfpicktraybypdfsize / unknown .code:n =
1834 {
1835     \msg_warning:nnee { hyp } { no-bool }
1836     { picktraybypdfsize }
1837     { \exp_not:n {#1} }
1838 }
1839 ,pdfprintarea .choices:nn =
1840 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1841 {
1842     \pdf_version_compare:NnTF < {2.0}
1843     {
1844         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1845         { PrintArea } { /#1 }
1846     }
1847     {
1848         \msg_warning:nnee
1849         {hyp}
1850         {ignore-deprecated-or-unknown-option-in-pdf-version}
1851         {pdfprintarea}
1852         {\pdf_version:}
1853     }
1854 }%
1855 ,pdfprintarea / .code:n =
1856 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1857 ,pdfprintarea / unknown .code:n =
1858 {
1859     \msg_warning:nneee { hyp } { unknown-choice+empty }
1860     { pdfprintarea }
1861     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1862     { \exp_not:n {#1} }
1863 }

```

```

1864 ,pdfprintclip .choices:nn =
1865 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1866 {
1867   \pdf_version_compare:NnTF < {2.0}
1868   {
1869     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1870     { PrintClip } { /#1 }
1871   }
1872   {
1873     \msg_warning:nnee
1874     {hyp}
1875     {ignore-deprecated-or-unknown-option-in-pdf-version}
1876     {pdfprintclip}
1877     {\pdf_version:}
1878   }
1879 }%
1880 ,pdfprintclip / .code:n =
1881 {
1882   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1883 }
1884 ,pdfprintclip / unknown .code:n =
1885 {
1886   \msg_warning:nnee
1887   { hyp }
1888   { unknown-choice+empty }
1889   { pdfprintclip }
1890   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1891   { \exp_not:n {#1} }
1892 }
1893 ,pdfprintpagerange .code:n =
1894 {
1895   \pdf_version_compare:NnTF > {1.6}
1896   {
1897     \tl_if_empty:nTF { #1}
1898     {
1899       \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1900       { PrintPageRange }
1901     }
1902     {
1903       \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1904       {PrintPageRange}{[#1]}
1905     }
1906   }
1907   {
1908     \msg_warning:nnee
1909     {hyp}
1910     {ignore-deprecated-or-unknown-option-in-pdf-version}
1911     {pdfprintpagerange}
1912     {\pdf_version:}
1913   }
1914 }
1915 ,pdfprintscaling .choices:nn =
1916 { None, AppDefault }
1917 {

```

```

1918     \pdf_version_compare:NnTF > {1.5}
1919     {
1920         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1921         { PrintScaling } { /#1 }
1922     }
1923     {
1924         \msg_warning:nnee
1925         {hyp}
1926         {ignore-deprecated-or-unknown-option-in-pdf-version}
1927         {pdfprintscaling}
1928         {\pdf_version:}
1929     }
1930 }%
1931 ,pdfprintscaling / .code:n =
1932 {
1933     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1934 }
1935 ,pdfprintscaling / unknown .code:n =
1936 {
1937     \msg_warning:nneee { hyp } { unknown-choice+empty }
1938     { pdfprintarea }
1939     { None, AppDefault }
1940     { \exp_not:n {#1} }
1941 }
1942 ,pdfremotestartview .code:n =
1943 {
1944     \tl_set:Ne \l__hyp_tmpa_tl {#1-null-null-null~}
1945     \exp_args:NNV
1946     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_seq
1947     {
1948         \tl_set:Ne \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1949     }
1950     {
1951         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1952         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1953     }
1954 }
1955 ,pdfremotestartview .initial:n = {Fit}
1956 % pdfstartpage is special as it shares code with pdfstartview
1957 ,pdfstartpage .code:n =
1958 {
1959     \tl_gset:Ne \g__hyp_dest_pdfstartpage_tl { #1 }
1960     \bool_if:nTF
1961     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl }
1962     {
1963         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1964     }
1965     {
1966         \pdfmanagement_add:nne {Catalog} { OpenAction }
1967         {
1968             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartpage_tl]
1969         }
1970     }
1971 }

```

```

1972 ,pdfstartpage .initial:n =1
1973 ,pdfstartview .code:n =
1974 {
1975     \tl_set:Nc \l__hyp_tmpa_tl {#1~null~null~null~}
1976     \exp_args:NNV
1977     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1978     {
1979         \tl_gset:Nc \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1980     }
1981     {
1982         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1983         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1984     }
1985     \bool_if:nTF
1986     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1987     {
1988         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1989     }
1990     {
1991         \pdfmanagement_add:nne {Catalog} { OpenAction }
1992         {
1993             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1994         }
1995     }
1996 }
1997 ,pdfstartview .initial:n = Fit
1998 ,pdftoolbar .choice:
1999 ,pdftoolbar / true .code:n =
2000 {
2001     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
2002 }
2003 ,pdftoolbar / false .code:n =
2004 {
2005     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
2006     { HideToolbar } { true }
2007 }
2008 ,pdftoolbar / true .code:n =
2009 {
2010     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
2011 }
2012 ,pdftoolbar .default:n = true
2013 ,pdftoolbar / unknown .code:n =
2014 {
2015     \msg_warning:nnee { hyp } { no-bool }
2016     { pdftoolbar }
2017     { \exp_not:n {#1} }
2018 }
2019 % pdfview see below.
2020 ,pdfviewarea .choices:nn =
2021 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2022 {
2023     \pdf_version_compare:NnTF < {2.0}
2024     {
2025         \pdfmanagement_add:nnn {Catalog / ViewerPreferences}

```

```

2026         { ViewArea } { /#1 }
2027     }
2028     {
2029         \msg_warning:nnee
2030         {hyp}
2031         {ignore-deprecated-or-unknown-option-in-pdf-version}
2032         {pdfviewarea}
2033         {\pdf_version:}
2034     }
2035 }%
2036 ,pdfviewarea / .code:n =
2037 {
2038     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
2039 }
2040 ,pdfviewarea / unknown .code:n =
2041 {
2042     \msg_warning:nneee { hyp } { unknown-choice+empty }
2043     { pdfviewarea }
2044     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2045     { \exp_not:n {#1} }
2046 }
2047 ,pdfviewclip .choices:nn =
2048 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2049 {
2050     \pdf_version_compare:NnTF < {2.0}
2051     {
2052         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2053         { ViewClip } { /#1 }
2054     }
2055     {
2056         \msg_warning:nnee
2057         {hyp}
2058         {ignore-deprecated-or-unknown-option-in-pdf-version}
2059         {pdfviewclip}
2060         {\pdf_version:}
2061     }
2062 }%
2063 ,pdfviewclip / .code:n =
2064 {
2065     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
2066 }
2067 ,pdfviewclip / unknown .code:n =
2068 {
2069     \msg_warning:nneee { hyp } { unknown-choice+empty }
2070     { pdfviewclip }
2071     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2072     { \exp_not:n {#1} }
2073 }
2074 ,pdfwindowui .choice:
2075 ,pdfwindowui / true .code:n =
2076 {
2077     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2078 }
2079 ,pdfwindowui / false .code:n =

```

```

2080     {
2081       \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2082       { HideWindowUI } { true }
2083     }
2084     ,pdfwindowui / .code:n =
2085     {
2086       \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2087     }
2088     ,pdfwindowui / unknown .code:n =
2089     {
2090       \msg_warning:nnee { hyp } { no-bool }
2091       { pdfwindowui }
2092       { \exp_not:n {#1} }
2093     }
2094     ,pdfwindowui .default:n = true
2095   }

```

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

```

2096   \keys_define:nn { hyp }
2097   {
2098     ,pdfview .code:n =
2099     {
2100       \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
2101       \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } } {
2102         {
2103           { xyz }
2104           {
2105             \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
2106             {
2107               \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2108               \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2109               {
2110                 \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2111               }
2112               {
2113                 \tl_set:Ne \l__hyp_dest_pdfview_tl
2114                 {
2115                   \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2116                 }
2117               }
2118             }
2119             {
2120               \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2121             }
2122           }
2123           { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2124           { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2125           { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2126           { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2127           { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2128           { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2129           { fitr }
2130           {
2131             \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}

```

```

2132         {
2133         \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2134         }
2135         {
2136         %ensure 4 values ...
2137         \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2138         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2139         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2140         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2141         \hbox_set_to_wd:Nnn \l__hyp_dest_box
2142         {
2143         \fp_eval:n
2144         {
2145         round
2146         (
2147         abs
2148         (
2149         \seq_item:Nn\l__hyp_tmpa_seq{4}
2150         -
2151         (\seq_item:Nn\l__hyp_tmpa_seq{2}))
2152         ),
2153         3
2154         )
2155         }bp
2156         }{}
2157         \box_set_dp:Nn \l__hyp_dest_box
2158         {
2159         \fp_eval:n
2160         {
2161         round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2162         }bp
2163         }
2164         \box_set_ht:Nn \l__hyp_dest_box
2165         {
2166         \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2167         }
2168         }
2169     }
2170 }
2171 {
2172 \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2173 \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2174 }
2175 }
2176 ,pdfview .initial:n = {xyz}
2177 }

```

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.

```

2178 \keys_define:nn { hyp }

```



```

2179 {
2180   ,pdflang .code:n =
2181   {
2182     \tl_if_empty:nF { #1 }
2183     {
2184       \pdfmanagement_add:nne {Catalog} { Lang } { (#1) }
2185       \AddToDocumentProperties[document]{lang}{#1}
2186     }
2187   }
2188 }

```

12.3.1 “info keys”

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

pdftitle (*setup key*) Creator and Producer can’t be removed with the pdfmanagement, but we allow to set an

pdfcreator (*setup key*) empty value. If the value begin with an optional argument, we assume a multilanguage

pdfsubject (*setup key*) clist and use only the first value.

```

pdfproducer (setup key) 2189 \regex_new:N\l__hyp_optlang_regex
pdfkeywords (setup key) 2190 \regex_set:Nn\l__hyp_optlang_regex {\A\[([A-Za-z\-\_]+)\](.*)}
2191 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2192 {
2193   \keys_define:nn { hyp }
2194   {
2195     pdf#1 .code:n =
2196     {
2197       \tl_if_blank:nTF {##1}
2198       {
2199         \str_case:nnF { #1 }
2200         {
2201           {creator}
2202           {
2203             \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2204             \pdfmanagement_add:nne {Info}{Creator}{()}
2205           }
2206           {producer}
2207           {
2208             \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2209             \pdfmanagement_add:nne {Info}{Producer}{()}
2210           }
2211         }
2212         {
2213           \pdfmanagement_remove:nn {Info}{#2}
2214         }
2215       }
2216     }
2217     \tl_set:N\l__hyp_tmpa_tl {\clist_item:nn{##1}{1}}
2218     \exp_args:NNV
2219     \regex_extract_once:NnN \l__hyp_optlang_regex \l__hyp_tmpa_tl\l__hyp_tmpa_seq
2220     \seq_if_empty:NTF\l__hyp_tmpa_seq
2221     {
2222       \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2223     }
2224     {

```

```

2225         \exp_args:Ne
2226         \__hyp_text_pdfstring_info:nN {\seq_item:Nn \l__hyp_tmpa_seq{3}}\l__hyp_t
2227     }
2228     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2229     {
2230         \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2231     }
2232 }
2233 \__hyp_store_metadata:nn {pdf#1}{##1}
2234 }
2235 }
2236 \keys_define:nn { hyp / info }
2237 {
2238     #2 .code:n =
2239     {
2240         \tl_if_blank:nTF {##1}
2241         {
2242             \pdfmanagement_remove:nn {Info}{#2}
2243         }
2244         {
2245             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2246             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2247             {
2248                 \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2249             }
2250         }
2251         \exp_args:Ne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2252     }
2253 ,unknown .code:n =
2254 {
2255     \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2256     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2257     {
2258         \exp_args:Nno
2259         \pdfmanagement_add:nne {Info}
2260             { \l_keys_key_str } { \l__hyp_tmpa_str }
2261     }
2262 }
2263 }
2264 }
2265 \__hyp_setup_info_key:nn {author} {Author}
2266 \__hyp_setup_info_key:nn {title} {Title}
2267 \__hyp_setup_info_key:nn {producer} {Producer}
2268 \__hyp_setup_info_key:nn {creator} {Creator}
2269 % ignored key: addtopdfcreator
2270 \__hyp_setup_info_key:nn {subject} {Subject}
2271 \__hyp_setup_info_key:nn {keywords} {Keywords}

```

pdfcreationdate (setup key) These keys are not really needed. We store them too in the container. CreationDate and
pdfmoddate (setup key) ModDate should not use the hex encoding.
pdfmetadate (setup key)

```

2272 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2273 {
2274     \keys_define:nn { hyp }
2275     {

```

```

2276 pdf#1 .code:n =
2277 {
2278   \tl_if_blank:nTF {##1}
2279   {
2280     \pdfmanagement_remove:nn {Info}{#2}
2281   }
2282   {
2283     \pdfmanagement_add:nne {Info}{#2}{(##1)}
2284   }
2285   \__hyp_store_metadata:nn {pdf#1}{##1}
2286   \AddToDocumentProperties[document]{#1}{##1}
2287 }
2288 }
2289 \keys_define:nn { hyp / info }
2290 {
2291   #2 .code:n =
2292   {
2293     \tl_if_blank:nTF {##1}
2294     {
2295       \pdfmanagement_remove:nn {Info}{#2}
2296     }
2297     {
2298       \pdfmanagement_add:nne {Info}{#2}{(##1)}
2299     }
2300     \exp_args:Ne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2301   }
2302 }
2303 }
2304
2305 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2306 \__hyp_setup_info_date_key:nn {moddate} {ModDate}
2307 \keys_define:nn { hyp }
2308 {
2309   pdfmetadate .code:n = { \__hyp_store_metadata:nn {pdfmetadate}{#1} }
2310 }

```

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

```

2311 \keys_define:nn { hyp }
2312 {
2313   ,pdftrapped .code:n =
2314   {
2315     \exp_args:Nne
2316     \keys_set:nn { hyp } { _pdftrapped = \str_uppercase:n { #1 } }
2317   }
2318   ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2319   {
2320     \pdfmanagement_add:nne {Info}{Trapped}
2321     {/
2322       \str_uppercase:f { \str_head:n { #1 } }
2323       \str_lowercase:f { \str_tail:n { #1 } }
2324     }
2325     \__hyp_store_metadata:ne {pdftrapped}
2326     {
2327       \str_uppercase:f { \str_head:n { #1 } }

```

```

2328         \str_lowercase:f { \str_tail:n { #1 } }
2329     }
2330 }
2331 ,_pdftrapped / unknown .code:n =
2332 {
2333     \msg_warning:nneee { hyp } { unknown-choice }
2334     { pdftrapped }
2335     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2336     { \exp_not:n {#1} }
2337 }
2338 }

```

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

```

2339 \keys_define:nn { hyp }
2340 {
2341     pdfinfo .code:n =
2342     {
2343         \keys_set:nn { hyp / info } { #1 }
2344     }
2345 }

```

Now we set some default values

```

2346 \keys_set:nn { hyp } {pdfcreator = LaTeX-with-hyperref}
2347 \keys_set:nn { hyp } {pdfauthor  = }
2348 \keys_set:nn { hyp } {pdftitle   = }
2349 \keys_set:nn { hyp } {pdfsubject = }

```

12.4 hyperxmp keys

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

```

2350 \clist_map_inline:nn
2351 {
2352     ,pdfcopyright
2353     ,pdftype
2354     ,pdflicenseurl
2355     ,pdfauthortitle
2356     ,pdfcaptionwriter
2357     ,pdfmetalang
2358     ,pdfsource
2359     ,pdfdocumentid
2360     ,pdfinstanceid
2361     ,pdfversionid
2362     ,pdfrendition
2363     ,pdfpublication
2364     ,pdfpubtype
2365     ,pdfbytes
2366     ,pdfnumpages
2367     ,pdfissn
2368     ,pdfeissn
2369     ,pdfisbn
2370     ,pdfbookedition
2371     ,pdfpublisher

```

```

2372 ,pdfvolumenum
2373 ,pdfissuenum
2374 ,pdfpagerange
2375 ,pdfdoi
2376 ,pdfurl
2377 ,pdfidentifier
2378 ,pdfsubtitle
2379 ,pdfpubstatus
2380 ,pdfcontactaddress
2381 ,pdfcontactcity
2382 ,pdfcontactregion
2383 ,pdfcontactpostcode
2384 ,pdfcontactcountry
2385 ,pdfcontactphone
2386 ,pdfcontactemail
2387 ,pdfcontacturl
2388 ,pdfdate
2389 }
2390 {
2391   \keys_define:nn { hyp }
2392   {
2393     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2394   }
2395 }
2396

```

12.5 Transitions

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

```

2397 \keys_define:nn { hyp }
2398 {
2399   pdfpageduration .code:n =
2400   {
2401     \tl_if_blank:nTF { #1 }
2402     {
2403       \pdfmanagement_remove:nn {Page}{Dur}
2404     }
2405     {
2406       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2407     }
2408   }
2409 }

```

Transition settings are used by (some) pdf viewers when presenting a pdf in full screen mode. They are added to the page settings and describe the transition from the previous page to current page. Transition setting can be set in the preamble for all pages or in the document for the current and the following pages. Due to the asynchronous page breaking one has to be careful to set it on the right page, e.g. only after a `\newpage`. The generic driver uses a different syntax than the other `hyperref` drivers: various transition options can be set by a keyval syntax in the value of `pdfpagetransition`. A typical setting looks e.g. like this

```
\hypersetup{pdfpagetransition={style=Fly,duration=2,direction=90,opaque=false}}
```

The keys allowed in the argument of `pdfpagetransition` are

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

```

2410 \keys_define:nn { hyp }
2411 {
2412   pdfpagetransition .code:n =
2413   {
2414     \tl_if_blank:nTF {#1}
2415     {
2416       \pdfmanagement_remove:nn {Page}{Trans}
2417     }
2418     {
2419       \group_begin:
2420       \keys_set:nn { hyp / trans }{style=R,#1}
2421       \pdf_object_unnamed_write:ne { dict }
2422       {
2423         \pdfdict_use:n {l__hyp_page/Trans}
2424       }
2425       \pdfmanagement_add:nne {Page}{Trans}{\pdf_object_ref_last:}
2426       \group_end:
2427     }
2428   }
2429 }
2430 \keys_define:nn { hyp / trans }
2431 {
2432   ,style .choices:nn =
2433   {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2434   { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2435   ,style / unknown .code:n =
2436   {
2437     \msg_warning:nneee { hyp } { unknown-choice }
2438     { trans / style }
2439     { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2440     { \exp_not:n {#1} }
2441   }
2442   ,duration .code:n =
2443   {
2444     \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}
  
```

```

2445     }
2446 ,direction .choices:nn =
2447     {H,V}
2448     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2449 ,direction .choices:nn =
2450     {0,90,180,270,315}
2451     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2452 ,direction / None .code:n =
2453     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2454 ,direction / unknown .code:n =
2455     {
2456         \msg_warning:nneee { hyp } { unknown-choice }
2457         { trans / direction }
2458         {
2459             H~(horizontal,~only~Split,~Blinds),
2460             V~(vertical,~only~Split,~Blinds),
2461             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2462             90~(bottom~to~top,~only~Wipe),
2463             180~(right~to~left,~only~Wipe),
2464             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2465             315~(top~left~to~bottom,~only~Glitter),
2466             None~(only~Fly)
2467         }
2468         { \exp_not:n {#1} }
2469     }
2470 ,motion .choices:nn =
2471     {I,O}
2472     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2473 ,motion / unknown .code:n =
2474     {
2475         \msg_warning:nneee { hyp } { unknown-choice }
2476         { trans / motion }
2477         { I~(inwards) , O~(outwards) }
2478         { \exp_not:n {#1} }
2479     }
2480 ,scale .code:n =
2481     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2482 ,opaque .choices:nn = {true,false}
2483     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2484 ,opaque / unknown .code:n =
2485     {
2486         \msg_warning:nneee { hyp } { unknown-choice }
2487         { trans / B }
2488         { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2489         { \exp_not:n {#1} }
2490     }
2491 % try to set unknown keys as style
2492 ,unknown .code:n =
2493     {
2494         % warning ...
2495         \exp_args:Nne\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2496     }
2497 }

```

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

```

2498 \keys_set_known:nv{ hyp }{opt@hyperref.sty}
    Unfinished Form field code
2499 \NewDocumentCommand \MakeFieldObject { m m }
2500 {
2501   \pdfxform_new:nnn { #2 }{ } { #1 }
2502 }
2503
2504
2505 \prop_new:N    \g__hyp_AcroForm_CoFields_prop
2506 \prop_new:N    \g__hyp_AcroForm_Fields_prop
2507
2508 \let\HyField@afields\@empty
2509 \let\HyField@cofields\@empty
2510 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2511
2512 % the value doesn't matter, but with a prop we avoid duplicates and it is
2513 % clearly faster than removing them from a sequence
2514 \def\HyField@AuxAddToFields#1
2515 {
2516   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2517 }%
2518
2519 %fields with empty key get a value too -- lets hope that
2520 %this give the expected behaviour
2521 \def\HyField@AuxAddToCoFields #1 #2
2522 {
2523   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2524 }
2525
2526 \Hy@AtBeginDocument
2527 {
2528   \if@filesw
2529     \immediate\write\@mainaux{%
2530       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2531     }%
2532     \immediate\write\@mainaux{%
2533       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2534     }%
2535   \fi
2536   \let\HyField@AfterAuxOpen\@firstofone
2537 }%
2538
2539 \def\HyField@AddToFields
2540 {
2541   \exp_args:Ne\HyField__hypAddToFields
2542   {
2543     \pdfannot_box_ref_last:
2544   }
2545   \ifx\Fld@calculate@code\@empty
2546   \else
2547     \begingroup
2548     \Hy@safe@activetrue

```



```

2549     \edef\Hy@temp{%
2550     \endgroup
2551     \if@filesw
2552     \write\@mainaux
2553     {
2554     \string\HyField@AuxAddToCoFields
2555     {
2556     \Fld@calculate@sortkey
2557     }
2558     {
2559     \pdfannot_box_ref_last:
2560     }
2561     }
2562     \fi
2563     }%
2564     \Hy@temp
2565     \fi
2566     }%
2567
2568     \def\HyField__hypAddToFields#1{
2569     \HyField@AfterAuxOpen{%
2570     \if@filesw
2571     \write\@mainaux{%
2572     \string\HyField@AuxAddToFields{#1}%
2573     }%
2574     \fi
2575     }%
2576     }%
2577
2578     \ExplSyntaxOff
2579     \ExplSyntaxOn
2580
2581     \def\@Form[#1]
2582     {
2583     \kvsetkeys{Form}{#1}
2584     \pdf@ifdraftmode{}
2585     {
2586     \Hy@FormObjects
2587     \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2588     {
2589     \pdfmanagement_add:nne { Catalog / AcroForm } { Fields }{##1}
2590     %\pdfmanagement_show:n { Catalog / AcroForm }
2591     }
2592     \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2593     {
2594     \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2595     {
2596     \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2597     }
2598     \seq_sort:Nn \l__hyp_tmpa_seq
2599     {
2600     \str_compare:nNnTF {##1} > {##2}
2601     { \sort_return_swapped: }
2602     { \sort_return_same: }

```

```

2603     }
2604     \seq_map_inline:Nn \l__hyp_tmpa_seq
2605     {
2606         \pdfmanagement_add:nne { Catalog / AcroForm }
2607         { CO }
2608         {
2609             \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2610         }
2611     }
2612 }
2613 \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2614 {ZaDb} {\pdf_object_ref:n {\__hyp/Font/ZaDb} }
2615 \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2616 {Helv} {\pdf_object_ref:n {\__hyp/Font/Helv} }
2617 \pdfmanagement_add:nne {Catalog /AcroForm}
2618 {DA}{(/Helv~10~Tf~0~g)}
2619 \pdfmeta_standard_verify:nTF {form_no_NeedAppearances}
2620 {
2621     \legacy_if:nT { HyField@NeedAppearances }
2622     {
2623         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2624     }
2625 }
2626 {
2627     \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2628 }
2629 }
2630 }
2631 \ExplSyntaxOff
2632 \let\@endForm\@empty
2633 \let\HyAnn@AbsPageLabel\@empty
2634 \let\Fld@pageobjref\@empty
2635
2636 \ExplSyntaxOn
2637 \newcount\HyAnn@Count
2638 \HyAnn@Count=\z@
2639 \def\HyAnn@AbsPageLabel
2640 {
2641     \global\advance\HyAnn@Count by\@ne
2642     %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2643     %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2644     %\zref@refused{HyAnn@\the\HyAnn@Count}%
2645     \__hyp_property_record:ee {HyAnn@\the\HyAnn@Count}{abspage}
2646     \property_ref_undefined_warn:ee {HyAnn@\the\HyAnn@Count}{abspage}
2647 }%
2648 \prg_generate_conditional_variant:Nnn \property_if_recorded:nn {ee} {T}
2649 \def\Fld@pageobjref
2650 {
2651     \property_if_recorded:eeT {HyAnn@\the\HyAnn@Count}{abspage}
2652     {
2653         /P~\pdf_pageobject_ref:e
2654         {
2655             \property_ref:ee{HyAnn@\the\HyAnn@Count}{abspage}
2656         }

```

```

2657     }
2658 }
2659 \ExplSyntaxOff
2660 \ExplSyntaxOn
2661 %% check if the attr should be set through
2662 %% hooks.
2663 %% check if options are missing.
2664 \def\@TextField[#1]#2{% parameters, label
2665   \def\Fld@name{#2}%
2666   \let\Fld@default\@empty
2667   \let\Fld@value\@empty
2668   \def\Fld@width{\DefaultWidthofText}%
2669   \def\Fld@height{%
2670     \ifFld@multiline
2671       \DefaultHeightofTextMultiline
2672     \else
2673       \DefaultHeightofText
2674     \fi
2675   }%
2676   \begingroup
2677     \expandafter\HyField@SetKeys\expandafter{%
2678       \DefaultOptionsofText,#1%
2679     }%
2680     \PDFForm@Name
2681     \HyField@FlagsText
2682     \ifFld@hidden\def\Fld@width{1sp}\fi
2683     \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2684     \LayoutTextField{#2}{%
2685       \leavevmode
2686       \HyAnn@AbsPageLabel
2687       \Hy@escapeform\PDFForm@Text
2688       \pdfannot_box:nnnn
2689         {\Fld@width}
2690         {\Fld@height}
2691         {Opt} %is this correct?
2692         {\PDFForm@Text}
2693       \MakeTextField{\Fld@width}{\Fld@height}
2694       \HyField@AddToFields
2695     }%
2696   \endgroup
2697 }
2698 \providecommand\@curropt{}
2699 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2700   \def\Fld@name{#2}
2701   \let\Fld@default\relax
2702   \let\Fld@value\relax
2703   \def\Fld@width{\DefaultWidthofChoiceMenu}
2704   \def\Fld@height{\DefaultHeightofChoiceMenu}
2705   \begingroup
2706     \Fld@menulength=0 %
2707     \@tempdima\z@
2708     \clist_map_variable:nNn { #3 } \@curropt
2709     %\@for\@curropt:=#3\do
2710     {%

```

```

2711 \expandafter\Fld@checkequals\@curropt==\\%
2712 \Hy@StepCount\Fld@menulength
2713 \settowidth{\@tempdimb}{\@currDisplay}%
2714 \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2715 }%
2716 \advance\@tempdima by~15\p@
2717 \begingroup
2718 \HyField@SetKeys{#1}
2719 \edef\x{\endgroup
2720 \noexpand\expandafter
2721 \noexpand\HyField@SetKeys
2722 \noexpand\expandafter{%
2723 \expandafter\noexpand\csname DefaultOptionsof%
2724 \ifFld@radio
2725 Radio%
2726 \else
2727 \ifFld@combo
2728 \ifFld@popdown
2729 PopdownBox%
2730 \else
2731 ComboBox%
2732 \fi
2733 \else
2734 ListBox%
2735 \fi
2736 \fi
2737 \endcsname
2738 }%
2739 }\x
2740 \HyField@SetKeys{#1}%
2741 \PDFForm@Name
2742 \ifFld@hidden\def\Fld@width{1sp}\fi
2743 \ifx\Fld@value\relax
2744 \let\Fld@value\Fld@default
2745 \fi
2746 \LayoutChoiceField{#2}{%
2747 \ifFld@radio
2748 \HyField@FlagsRadioButton
2749 \__hypRadio{#3}%
2750 \else
2751 \begingroup
2752 \HyField@FlagsChoice
2753 \ifdim\Fld@width<\@tempdima
2754 \ifdim\@tempdima<1cm\@tempdima1cm\fi
2755 \edef\Fld@width{\the\@tempdima}%
2756 \fi
2757 \ifFld@combo
2758 \else
2759 \@tempdima=\the\Fld@menulength\Fld@charsize
2760 \advance\@tempdima by~\Fld@borderwidth bp %
2761 \advance\@tempdima by~\Fld@borderwidth bp %
2762 \edef\Fld@height{\the\@tempdima}%
2763 \fi
2764 \__hypListbox{#3}%

```

```

2765         \endgroup
2766     \fi
2767 }%
2768 \endgroup
2769 }
2770
2771 \def\__hypRadio#1{%
2772     \Fld@listcount=0~%
2773     %\show\Fld@default
2774     \EdefEscapeName\Fld@default{\Fld@default}%
2775     \clist_map_variable:nNn { #1 } \@curropt
2776     %\@for\@curropt:=#1\do
2777     {%
2778         \expandafter\Fld@checkequals\@curropt==\%
2779         \EdefEscapeName\@currValue{\@currValue}%
2780         \Hy@StepCount\Fld@listcount
2781         \@currDisplay\space
2782         \leavevmode
2783         \HyAnn@AbsPageLabel
2784         \Hy@escapeform\PDFForm@Radio
2785         \pdfxform_if_exist:nF { __hyp_xform_Ding }
2786         {
2787             \pdfxform_new:nnn { __hyp_xform_Ding } {}
2788             {
2789                 \group_begin:
2790                 \fontfamily{pzd}
2791                 \fontencoding{U}
2792                 \fontseries{m}
2793                 \fontshape{n}
2794                 \selectfont
2795                 \char123
2796                 \group_end:
2797             }
2798         }
2799         \pdfannot_box:nnne
2800         {\Fld@width}
2801         {\Fld@height}
2802         {0pt} %is this correct?
2803         {
2804             \PDFForm@Radio
2805             /AP
2806             <<
2807             /N
2808             <<
2809             /\@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2810             %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2811             >>
2812             >>
2813         }
2814         {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2815         \int_compare:nNnT { \Fld@listcount } = { 1 }
2816         { \HyField@AddToFields }
2817         \c_space_tl % deliberate space between radio buttons
2818         % to do: --> should be configurable

```

```

2819 }%
2820 }
2821
2822 \newcount\Fld@listcount
2823 \def\__hypListbox#1
2824 {
2825   \HyField@PDFChoices{#1}
2826   \mode_leave_vertical:
2827   \HyAnn@AbsPageLabel
2828   \Hy@escapeform\PDFForm@List
2829   \pdf_link_user:nnn
2830   {widget} %perhaps we need more types??
2831   {\PDFForm@List}
2832   {\MakeChoiceField{\Fld@width}{\Fld@height}}
2833   \HyField@AddToFields
2834 }
2835
2836
2837 \def\@PushButton[#1]#2{% parameters, label
2838   \def\Fld@name{#2}%
2839   \group_begin:
2840     \exp_args:No\HyField@SetKeys
2841     {
2842       \DefaultOptionsofPushButton,#1
2843     }
2844     \PDFForm@Name
2845     \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2846     {
2847       \HyField@FlagsPushButton
2848       \legacy_if:nT {Fld@hidden}
2849       {
2850         \def\Fld@width{1sp}
2851       }
2852       \LayoutPushButtonField
2853       {
2854         \mode_leave_vertical:
2855         \HyAnn@AbsPageLabel
2856         \Hy@escapeform\PDFForm@Push
2857         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2858         \pdfannot_box:nnnn
2859         {\box_wd:N\l_tmpa_box}
2860         {\box_ht:N\l_tmpa_box}
2861         {\box_dp:N\l_tmpa_box} %is this correct?
2862         {\PDFForm@Push}
2863         {\box_use:N\l_tmpa_box}
2864         \HyField@AddToFields
2865       }
2866     }
2867     {
2868       \msg_error:nn { hyp }{ pdfa-no-push-button }
2869       \LayoutPushButtonField
2870       {
2871         \mode_leave_vertical:
2872         \MakeButtonField{#2}

```

```

2873     }
2874 }
2875 \group_end:
2876 }
2877
2878 \def\@Submit[#1]#2
2879 {
2880   \def\Fld@width {\DefaultWidthofSubmit}
2881   \def\Fld@height{\DefaultHeightofSubmit}
2882   \group_begin:
2883     \exp_args:No\HyField@SetKeys
2884     {
2885       \DefaultOptionsofSubmit,#1
2886     }
2887     \HyField@FlagsPushButton
2888     \HyField@FlagsSubmit
2889     \legacy_if:nT { Fld@hidden }
2890     {
2891       \def\Fld@width{1sp}
2892     }
2893     \mode_leave_vertical:
2894     \HyAnn@AbsPageLabel
2895     \Hy@escapeform\PDFForm@Submit
2896     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2897     \pdfxform_if_exist:nF
2898     { __hyp_xform_Submit }
2899     {
2900       \pdfxform_new:nnn { __hyp_xform_Submit }{}
2901       {
2902         \fbox{\color_select:n{yellow}\textsf{Submit}}
2903       }
2904       \pdfxform_new:nnn { __hyp_xform_SubmitP }{}
2905       {
2906         \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2907       }
2908     }
2909     \pdfannot_box:nnnn
2910     {\box_wd:N\l_tmpa_box}
2911     {\box_ht:N\l_tmpa_box}
2912     {\box_dp:N\l_tmpa_box} %is this correct?
2913     {
2914       \PDFForm@Submit
2915       /AP<<
2916       /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2917       /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2918       >>
2919     }
2920     \HyField@AddToFields
2921     \box_use:N\l_tmpa_box
2922
2923   \group_end:
2924 }
2925
2926 \def\@Reset[#1]#2

```

```

2927 {
2928   \def\Fld@width {\DefaultWidthofReset}
2929   \def\Fld@height{\DefaultHeightofReset}
2930   \group_begin:
2931     \exp_args:No\HyField@SetKeys
2932     {
2933       \DefaultOptionsofReset,#1
2934     }
2935     \mode_leave_vertical:
2936     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2937     {
2938       \HyField@FlagsPushButton
2939       \legacy_if:nT { Fld@hidden }
2940       { \def\Fld@width{1sp} }
2941       \HyAnn@AbsPageLabel
2942       \Hy@escapeform\PDFForm@Reset
2943       \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2944       \pdfannot_box:nnnn
2945       {\box_wd:N\l_tmpa_box}
2946       {\box_ht:N\l_tmpa_box}
2947       {\box_dp:N\l_tmpa_box} %is this correct?
2948       { \PDFForm@Reset }
2949       \HyField@AddToFields
2950       \box_use:N \l_tmpa_box
2951     }
2952     {
2953       \msg_error:nn { hyp }{ pdfa-no-reset-button }
2954       \MakeButtonField{#2}
2955     }
2956   \group_end:
2957 }
2958
2959 \def\@CheckBox[#1]#2
2960 {% parameters, label
2961   \def\Fld@name{#2}
2962   \def\Fld@default{0}
2963   \group_begin:
2964     \def\Fld@width {\DefaultWidthofCheckBox}
2965     \def\Fld@height{\DefaultHeightofCheckBox}
2966     \exp_args:No\HyField@SetKeys
2967     {
2968       \DefaultOptionsofCheckBox,#1
2969     }
2970     \PDFForm@Name
2971     \HyField@FlagsCheckBox
2972     \legacy_if:nT { Fld@hidden }
2973     {
2974       \def\Fld@width{1sp}
2975     }
2976     \LayoutCheckField{#2}
2977     {
2978       \mode_leave_vertical:
2979       \HyAnn@AbsPageLabel
2980       \Hy@escapeform\PDFForm@Check

```



```

2981 \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
2982 {
2983   \pdfxform_new:nnn
2984   {__hyp_xform_CheckMarkYes}-{ }
2985   {
2986     \group_begin:
2987     \fontfamily{pzd}
2988     \fontencoding{U}
2989     \fontseries{m}
2990     \fontshape{n}
2991     \selectfont
2992     \char51
2993     \group_end:
2994   }
2995   \pdfxform_new:nnn
2996   {__hyp_xform_CheckMarkOff}-{ }
2997   {
2998     \group_begin:
2999     \fontfamily{pzd}
3000     \fontencoding{U}
3001     \fontseries{m}
3002     \fontshape{n}
3003     \selectfont
3004     \phantom{\char51} %perhaps xetex needs some small glyph ..
3005     \group_end:
3006   }
3007 }
3008 \pdfannot_box:nnnn
3009 {\Fld@width}
3010 {\Fld@height}
3011 {Opt} %is this correct?
3012 {\PDFForm@Check}
3013 \HyField@AddToFields %check if this works with xelatex ...
3014 }
3015 \group_end:
3016 }
3017 \ExplSyntaxOff
3018
3019 %hm. Should a luatex driver use type1 fonts in fields????
3020 \ExplSyntaxOn
3021 \def\Hy@FormObjects
3022 {
3023   \pdf_object_new:n {__hyp/Encoding/pdfdoc }
3024   \pdf_object_new:n {__hyp/Font/ZaDb }
3025   \pdf_object_new:n {__hyp/Font/Helv }
3026   \pdf_object_write:nne {__hyp/Encoding/pdfdoc } { dict }
3027   {
3028     /Type/Encoding
3029     /Differences[
3030       24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
3031       /ring/tilde
3032       \c_space_tl
3033       39/quotesingle
3034       \c_space_tl

```

```

3035      96/grave %
3036      \iow_newline:
3037      128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
3038      /fraction/guilsinglleft/guilsinglright/minus/perthousand
3039      /quotedblbase/quotedblleft/quotedblright/quoteleft
3040      /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
3041      /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
3042      \iow_newline:
3043      164/currency
3044      \c_space_tl
3045      166/brokenbar
3046      \c_space_tl
3047      168/dieresis/copyright/ordfeminine
3048      \c_space_tl
3049      172/logicalnot/.notdef/registered/macron/degree/plusminus
3050      /twosuperior/threesuperior/acute/mu
3051      \c_space_tl
3052      183/periodcentered/cedilla/onesuperior/ordmasculine
3053      \c_space_tl
3054      188/onequarter/onehalf/threequarters
3055      \iow_newline:
3056      192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
3057      /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
3058      /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
3059      /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
3060      /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
3061      /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
3062      /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
3063      /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
3064      /ocircumflex/otilde/odieresis/divide/oslash/ugrave
3065      /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
3066    ]
3067  }
3068  \pdf_object_write:nnn {__hyp/Font/ZaDb } { dict }
3069  {
3070    /Type/Font
3071    /Subtype/Type1
3072    /Name/ZaDb
3073    /BaseFont/ZapfDingbats
3074  }
3075  \pdf_object_write:nne {__hyp/Font/Helv } { dict }
3076  {
3077    /Type/Font
3078    /Subtype/Type1
3079    /Name/Helv
3080    /BaseFont/Helvetica
3081    /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
3082  }
3083  \global\let\Hy@FormObjects\relax
3084  }
3085  \ExplSyntaxOff
3086  \providecommand*{\Fld@pageobjref}{\}
3087  \ifcsname pdf@escapestring\endcsname
3088    \def\Hy@escapeform#1{%

```

```

3089     \ifHy@pdfescapeform
3090     \let\Hy@escapestring\pdfescapestring
3091     \else
3092     \let\Hy@escapestring\@firstofone
3093     \fi
3094 }%
3095 \Hy@escapeform{}%
3096 \else
3097 \let\Hy@escapestring\@firstofone
3098 \def\Hy@escapeform#1{%
3099     \ifHy@pdfescapeform
3100     \def\Hy@escapestring##1{%
3101         \noexpand\Hy@escapestring{\noexpand##1}%
3102     }%
3103     \edef\Hy@temp{#1}%
3104     \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{}\@nil
3105     \def\Hy@escapestring##1{%
3106         \@ifundefined{Hy@esc@\string##1}{%
3107             ##1%
3108             \ThisShouldNotHappen
3109         }{%
3110             \csname Hy@esc@\string##1\endcsname
3111         }%
3112     }%
3113     \else
3114     \let\Hy@escapestring\@firstofone
3115     \fi
3116 }%
3117 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3118     \ifx\#3\%
3119     \else
3120     \expandafter
3121     \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3122     \Hy@ReturnAfterFi{%
3123         \Hy__hypescapeform#3\@nil
3124     }%
3125     \fi
3126 }%
3127 \fi
3128 \def\PDFForm@Name{%
3129     \PDFForm__hypName\Fld@name
3130     \ifx\Fld@altname\relax
3131     \else
3132     \PDFForm__hypName\Fld@altname
3133     \fi
3134     \ifx\Fld@mappingname\relax
3135     \else
3136     \PDFForm__hypName\Fld@mappingname
3137     \fi
3138 }
3139 \def\PDFForm__hypName#1{%
3140     \begingroup
3141     \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3142     \ifHy@unicode

```

```

3143         \Hy@unicodedefalse
3144     \fi
3145 \fi
3146 \pdfstringdef\Hy@gtemp#1%
3147 \endgroup
3148 \let#1\Hy@gtemp
3149 }
3150 \def\Fld@X@additionalactions{%
3151 \ifx\Fld@keystroke@code\@empty
3152 \else
3153 /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3154 \fi
3155 \ifx\Fld@format@code\@empty
3156 \else
3157 /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3158 \fi
3159 \ifx\Fld@validate@code\@empty
3160 \else
3161 /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3162 \fi
3163 \ifx\Fld@calculate@code\@empty
3164 \else
3165 /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3166 \fi
3167 \ifx\Fld@onfocus@code\@empty
3168 \else
3169 /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3170 \fi
3171 \ifx\Fld@onblur@code\@empty
3172 \else
3173 /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3174 \fi
3175 \ifx\Fld@onmousedown@code\@empty
3176 \else
3177 /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3178 \fi
3179 \ifx\Fld@onmouseup@code\@empty
3180 \else
3181 /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3182 \fi
3183 \ifx\Fld@onenter@code\@empty
3184 \else
3185 /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3186 \fi
3187 \ifx\Fld@onexit@code\@empty
3188 \else
3189 /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3190 \fi
3191 }
3192 \ExplSyntaxOn
3193 \def\Fld@additionalactions
3194 {%
3195 \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{\}
3196 {

```

```

3197     \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3198     {/AA<<\Fld@X@additionalactions>>}
3199   }
3200 }
3201 \ExplSyntaxOff
3202 \def\Fld@annotnames{%
3203   /T(\Fld@name)%
3204   \ifx\Fld@altname\relax
3205   \else
3206     /TU(\Fld@altname)%
3207   \fi
3208   \ifx\Fld@mappingname\relax
3209   \else
3210     /TM(\Fld@mappingname)%
3211   \fi
3212 }
3213 \ExplSyntaxOn
3214 \def\PDFForm@Check
3215 {
3216   /Subtype/Widget
3217   ~\Fld@annotflags
3218   ~\Fld@pageobjref
3219   ~\Fld@annotnames
3220   /FT/Btn
3221   \Fld@flags
3222   /Q~\Fld@align
3223   /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3224   /AP
3225   <<
3226     /N
3227     <<
3228       /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3229       /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3230     >>
3231   >>
3232   /MK<<
3233     \int_compare:nNnF {\Fld@rotation}={0}
3234     {
3235       /R~\Fld@rotation
3236     }
3237     \tl_if_empty:NF\Fld@bordercolor
3238     {
3239       /BC[\Fld@bordercolor]
3240     }
3241     \tl_if_empty:NF\Fld@bcolor
3242     {
3243       /BG[\Fld@bcolor]
3244     }
3245     /CA(\Hy@escapestring{\Fld@cbsymbol})%
3246   >>
3247   /DA
3248   (
3249     /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3250     \tl_if_empty:NF \Fld@color

```

```

3251         {
3252             \c_space_tl \Fld@color
3253         }
3254     )
3255 /H/P
3256 \legacy_if:nTF {Fld@checked}
3257 {
3258     /V/Yes /AS/Yes
3259 }
3260 {
3261     /V/Off /AS/Off
3262 }
3263 \Fld@additionalactions
3264 }
3265 \ExplSyntaxOff
3266 \ExplSyntaxOn
3267 \def\PDFForm@Push
3268 {
3269     /Subtype/Widget
3270     ~\Fld@annotflags
3271     ~\Fld@pageobjref
3272     ~\Fld@annotnames
3273     /FT/Btn
3274     ~\Fld@flags
3275     /H/P
3276     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3277     \bool_if:nT
3278     {
3279         !\int_compare_p:nNn {\Fld@rotation} = {0}
3280         ||
3281         \tl_if_exist_p:N \Fld@bordercolor
3282     }
3283     {
3284         /MK
3285         <<
3286         \int_compare:nNnF {\Fld@rotation} = {0}
3287         {
3288             /R~\Fld@rotation
3289         }
3290         \tl_if_exist:NT \Fld@bordercolor
3291         {
3292             /BC[\Fld@bordercolor]
3293         }
3294         >>
3295     }
3296     /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3297     \Fld@additionalactions
3298 }
3299
3300 \ExplSyntaxOff
3301 \def\PDFForm@List{%
3302     /Subtype/Widget%
3303     \Fld@annotflags
3304     \Fld@pageobjref

```

```

3305 \Fld@annotnames
3306 /FT/Ch%
3307 \Fld@flags
3308 /Q \Fld@align
3309 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3310 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3311 \ifx\Fld@bordercolor\relax\else 1\fi
3312 \ifx\Fld@bcolor\relax \else 1\fi
3313 \space
3314 \else
3315 /MK<<%
3316 \ifnum\Fld@rotation=\z@
3317 \else
3318 /R \Fld@rotation
3319 \fi
3320 \ifx\Fld@bordercolor\relax
3321 \else
3322 /BC[\Fld@bordercolor]%
3323 \fi
3324 \ifx\Fld@bcolor\relax
3325 \else
3326 /BG[\Fld@bcolor]%
3327 \fi
3328 >>%
3329 \fi
3330 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3331 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3332 \Fld@choices
3333 \Fld@additionalactions
3334 }
3335 \ExplSyntaxOn
3336 \def\PDFForm@Radio
3337 {
3338 /Subtype/Widget
3339 ~\Fld@annotflags
3340 ~\Fld@pageobjref
3341 ~\Fld@annotnames
3342 /FT/Btn
3343 \Fld@flags
3344 /H/P
3345 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3346 /MK<<
3347 \ifnum\Fld@rotation=\z@
3348 \else
3349 /R~\Fld@rotation
3350 \fi
3351 \ifx\Fld@bordercolor\relax
3352 \else
3353 /BC[\Fld@bordercolor]%
3354 \fi
3355 \ifx\Fld@bcolor\relax
3356 \else
3357 /BG[\Fld@bcolor]%
3358 \fi

```

```

3359     /CA(\Hy@escapestring{\Fld@radiosymbol})%
3360     >>
3361     /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3362         \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3363     \ifx\Fld@default\@empty
3364         /V/Off%
3365         /DV/Off%
3366     \else
3367         /V/\Fld@default
3368         /DV/\Fld@default
3369     \fi
3370     \Fld@additionalactions
3371 }
3372 \ExplSyntaxOff
3373 \ExplSyntaxOn
3374 % Does an appeareance dict make sense here?
3375 \def\PDFForm@Text
3376 {
3377     /Subtype/Widget
3378     ~\Fld@annotflags
3379     ~\Fld@pageobjref
3380     ~\Fld@annotnames
3381     /FT/Tx
3382     ~\Fld@flags
3383     /Q~\Fld@align
3384     /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3385     \bool_if:nT
3386     {
3387         !\int_compare_p:nNn {\Fld@rotation} = {0}
3388         ||
3389         \tl_if_exist_p:N \Fld@bordercolor
3390         ||
3391         \tl_if_exist_p:N \Fld@bcolor
3392     }
3393     {
3394         /MK
3395         <<
3396         \int_compare:nNnF {\Fld@rotation} = {0}
3397         {
3398             /R~\Fld@rotation
3399         }
3400         \tl_if_exist:NT \Fld@bordercolor
3401         {
3402             /BC[\Fld@bordercolor]
3403         }
3404         \tl_if_exist:NT \Fld@bcolor
3405         {
3406             /BG[\Fld@bcolor]
3407         }
3408         >>
3409     }
3410 /DA
3411 (
3412     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf

```



```

3413     \tl_if_empty:NF {\c_space_tl\Fld@color}
3414   )
3415   /DV(\Hy@escapestring{\Fld@default})
3416   /V(\Hy@escapestring{\Fld@value})
3417   ~\Fld@additionalactions
3418   \int_compare:nNnT { \Fld@maxlen}>{0}
3419   {
3420     /MaxLen~\Fld@maxlen
3421   }
3422 }
3423 \ExplSyntaxOff
3424
3425 \def\PDFForm@Submit{%
3426   /Subtype/Widget%
3427   \Fld@annotflags
3428   \Fld@pageobjref
3429   \Fld@annotnames
3430   /FT/Btn%
3431   \Fld@flags
3432   /H/P%
3433   /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3434   \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3435     \ifx\Fld@bordercolor\relax\else 1\fi
3436     \space
3437   \else
3438     /MK<<%
3439     \ifnum\Fld@rotation=\z@
3440     \else
3441       /R \Fld@rotation
3442     \fi
3443     \ifx\Fld@bordercolor\relax
3444     \else
3445       /BC[\Fld@bordercolor]%
3446     \fi
3447     >>%
3448   \fi
3449   /A<<%
3450   /S/SubmitForm%
3451   /F<<%
3452   /FS/URL%
3453   /F(\Hy@escapestring{\Form@action})%
3454   >>%
3455   \Fld@submitflags
3456   >>%
3457   \Fld@additionalactions
3458 }
3459 \ExplSyntaxOn
3460 \def\PDFForm@Reset{%
3461   /Subtype/Widget%
3462   \Fld@annotflags
3463   \Fld@pageobjref
3464   \Fld@annotnames
3465   /FT/Btn%
3466   \Fld@flags

```

```

3467         /H/P%
3468         /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3469         \ifcase0\ifnum\Fld@rotation=\z@   \else 1\fi
3470             \ifx\Fld@bordercolor\relax\else 1\fi
3471             \space
3472         \else
3473             /MK<<%
3474             \ifnum\Fld@rotation=\z@
3475             \else
3476                 /R~\Fld@rotation
3477             \fi
3478             \ifx\Fld@bordercolor\relax
3479             \else
3480                 /BC[\Fld@bordercolor]%
3481             \fi
3482             >>%
3483         \fi
3484         /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3485         /A<</S/ResetForm>>%
3486         \Fld@additionalactions
3487     }%
3488
3489
3490     %these patterns are used in hyperref checks.
3491     %it is unclear if they are really useful and if a backend support is
3492     %needed.
3493     \str_case:VnF \c_sys_backend_str
3494     {
3495         { pdfmode }
3496         {
3497             \def\HyPat@ObjRef
3498             {
3499                 [0-9]*[1-9][0-9]*~0~R
3500             }
3501         }
3502         { dvipdfmx }
3503         {
3504             \def\HyPat@ObjRef
3505             {
3506                 @[\~]+
3507             }
3508         }
3509         { xdvipdfmx }
3510         {
3511             \def\HyPat@ObjRef
3512             {
3513                 @[\~]+
3514             }
3515         }
3516     }
3517     { %also set in hyperref sty, so probably not needed.
3518         \def\HyPat@ObjRef/{.+}
3519     }
3520

```

```

3521
3522 \ExplSyntaxOff
3523 % UF: removed Hy@writebookmark
3524 % \Hy@currentbookmarklevel{0}
3525 % \Hy@numberline
3526 % \__hypwritetorep
3527 % counter{bookmark@seq@number}
3528 % removed \HyPsd@SanitizeForOutFile, not needed
3529 % removed \currentpdfbookmark, defined by bookmark,
3530 % should use \newcommand there
3531 % removed \subpdfbookmark, defined by bookmark,
3532 % should use \newcommand there
3533 % removed \belowpdfbookmark, defined by bookmark,
3534 % should use \newcommand there
3535 % removed \pdfbookmark, defined by bookmark,
3536 % \BOOKMARK
3537 % \@BOOKMARK
3538 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3539 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3540 %% removed \ReadBookmarks / unneeded with bookmark.
3541 %% removed \Hy@OutlineName
3542 %% removed \check@bm@number
3543 %% removed \calc@bm@number
3544
3545 \ifHy@implicit
3546 \else
3547 \expandafter\endinput
3548 \fi
3549 \newlength\Hy@SectionHShift
3550 \def\Hy@SectionAnchorHref#1{%
3551 \ifx\protect\@typeset@protect
3552 \Hy__hypSectionAnchor{#1}%
3553 \fi
3554 }
3555 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%
3556 \leavevmode
3557 \hbox to 0pt{%
3558 \kern-\Hy@SectionHShift
3559 \Hy@raisedlink{%
3560 \hyper@anchorstart{#1}\hyper@anchorend
3561 }%
3562 \hss
3563 }%
3564 }
3565 \@ifundefined{hyper@nopatch@sectioning}
3566 {
3567 \let\H@old@ssect\@ssect
3568 \def\@ssect#1#2#3#4#5{%
3569 \Hy@MakeCurrentHrefAuto{section*}%
3570 \setlength{\Hy@SectionHShift}{#1}%
3571 \begingroup
3572 \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3573 \toks\tw@\expandafter{%
3574 \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%

```

```

3575     #5%
3576   }%
3577   \edef\x{\endgroup
3578     \the\toks0{\the\toks\tw0}%
3579   }\x
3580 }
3581 \let\H@old@schapter\@schapter
3582 \def\@schapter#1{%
3583   \begingroup
3584     \let\mkboth@gobbletwo
3585     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3586     \Hy@raisedlink{%
3587       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3588     }%
3589   \endgroup
3590   \H@old@schapter{#1}%
3591 }
3592 \@ifundefined{chapter}{}{%
3593   \let\Hy@org@chapter\@chapter
3594   \def\@chapter{%
3595     \def\Hy@next{%
3596       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3597       \Hy@raisedlink{%
3598         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3599       }%
3600     }%
3601     \ifnum\c@secnumdepth>\m@ne
3602       \@ifundefined{if@mainmatter}%
3603       \iftrue{\csname if@mainmatter\endcsname}%
3604       \let\Hy@next\relax
3605     \fi
3606     \fi
3607     \Hy@next
3608     \Hy@org@chapter
3609   }%
3610 }
3611 \let\H@old@part\@part
3612 \begingroup\expandafter\expandafter\expandafter\endgroup
3613 \expandafter\ifx\csname chapter\endcsname\relax
3614   \let\Hy@secnum@part\z@
3615 \else
3616   \let\Hy@secnum@part\m@ne
3617 \fi
3618 \def\@part{%
3619   \ifnum\Hy@secnum@part>\c@secnumdepth
3620     \phantomsection
3621     \fi
3622   \H@old@part
3623 }
3624 \let\H@old@spart\@spart
3625 \def\@spart#1{%
3626   \Hy@MakeCurrentHrefAuto{part*}%
3627   \Hy@raisedlink{%
3628     \hyper@anchorstart{\@currentHref}\hyper@anchorend

```

```

3629 }%
3630 \H@old@spart{#1}%
3631 }
3632 \let\H@old@sect\@sect
3633 \def\@sect#1#2#3#4#5#6[#7]#8{%
3634 \ifnum #2>\c@secnumdepth
3635 \expandafter\@firstoftwo
3636 \else
3637 \expandafter\@secondoftwo
3638 \fi
3639 {%
3640 \Hy@MakeCurrentHrefAuto{section*}%
3641 \setlength{\Hy@SectionHShift}{#3}%
3642 \begingroup
3643 \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[#7]}%
3644 \toks\tw@\expandafter{%
3645 \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3646 #8%
3647 }%
3648 \edef\x{\endgroup
3649 \the\toks@{\the\toks\tw@}%
3650 }\x
3651 }{%
3652 \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[#7]{#8}%
3653 }%
3654 }
3655 }{}
3656 \expandafter\def\csname Parent-4\endcsname{}
3657 \expandafter\def\csname Parent-3\endcsname{}
3658 \expandafter\def\csname Parent-2\endcsname{}
3659 \expandafter\def\csname Parent-1\endcsname{}
3660 \expandafter\def\csname Parent0\endcsname{}
3661 \expandafter\def\csname Parent1\endcsname{}
3662 \expandafter\def\csname Parent2\endcsname{}
3663 \expandafter\def\csname Parent3\endcsname{}
3664 \expandafter\def\csname Parent4\endcsname{}
3665 %%
3666 %% End of file 'hgeneric-testphase.def'.
3667 </package>

3668 <*colorscheme>
3669 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3670 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3671 % linkcolor=
3672 %,filecolor=
3673 %,urlcolor=
3674 %,menucolor=
3675 %,runcolor=
3676 %,linkbordercolor=
3677 %,filebordercolor=
3678 %,urlbordercolor=
3679 %,menubordercolor=
3680 %,runbordercolor=
3681
3682 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }

```

```

3683 {
3684     linkcolor      = [rgb]{1,0,0}, %red
3685     filecolor      = [rgb]{0,1,1}, %cyan
3686     urlcolor       = [rgb]{1,0,1}, %magenta
3687     menucolor      = [rgb]{1, 0, 0}, %red
3688     runcolor       = [rgb]{0,1,1}, %cyan
3689 %-----
3690     linkbordercolor = [rgb]{1, 0 ,0 },
3691     filebordercolor = [rgb]{0, .5, .5},
3692     urlbordercolor  = [rgb]{0, 1, 1},
3693     menubordercolor = [rgb]{1, 0, 0},
3694     runbordercolor  = [rgb]{0, .7, .7}
3695 }
3696
3697 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3698 {
3699     linkcolor      = [rgb]{0,0.2,0.6},
3700     filecolor      = [rgb]{0.8,0,0.8},
3701     urlcolor       = [rgb]{0.8,0,0.8},
3702     menucolor      = [rgb]{0,0.2,0.6},
3703     runcolor       = [rgb]{0.8,0,0.8},
3704 %----- %-----
3705     linkbordercolor = [rgb]{0,0.2,0.6},
3706     filebordercolor = [rgb]{0.8,0,0.8},
3707     urlbordercolor  = [rgb]{0.8,0,0.8},
3708     menubordercolor = [rgb]{0,0.2,0.6},
3709     runbordercolor  = [rgb]{0.8,0,0.8}
3710 }
3711
3712 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3713 { %two colors: intern/extern
3714     linkcolor      = [rgb]{0.79216, 0, 0.12549},
3715     filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3716     urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3717     menucolor      = [rgb]{0.79216, 0, 0.12549 },
3718     runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3719 %----- %-----
3720     linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3721     filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3722     urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3723     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3724     runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3725 }
3726
3727 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3728 { %all darkgray
3729     linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3730     filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3731     urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3732     menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3733     runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3734 %----- %-----
3735     linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3736     filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },

```

```

3737     urlbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3738     menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3739     runbordercolor = [rgb]{0.4 ,0.4 ,0.4 }
3740 }
3741
3742 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3743 { %dvipsnam.def
3744     linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3745     filecolor      = [rgb]{1, 0, 0}, %Red
3746     urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3747     menucolor      = [rgb]{1, 0, 0}, %Red
3748     runcolor       = [rgb]{1, 0, 0}, %Red
3749     %----- %-----
3750     linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3751     filebordercolor = [rgb]{1, 0, 0}, %Red
3752     urlbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3753     menubordercolor = [rgb]{1, 0, 0}, %Red
3754     runbordercolor = [rgb]{1, 0, 0} %Red
3755 }
3756
3757 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3758 { %dvipsnam.def
3759     linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3760     filecolor      = [rgb]{0, 1, 0}, %Green
3761     urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3762     menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3763     runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3764     %----- %-----
3765     linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3766     filebordercolor = [rgb]{0, 1, 0}, %Green
3767     urlbordercolor = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3768     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3769     runbordercolor = [rgb]{0.64, 0.08, 0.98} %Mulberry
3770 }
3771
3772
3773 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3774 {
3775     linkcolor      = [rgb]{0.50196, 0, 0.02353},
3776     filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3777     urlcolor       = [rgb]{0.54118, 0, 0.52941},
3778     menucolor      = [rgb]{0.44706, 0.45882, 0},
3779     runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3780     %----- %-----
3781     linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3782     filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3783     urlbordercolor = [rgb]{0.724708, 0.4, 0.717646},
3784     menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3785     runbordercolor = [rgb]{0.444706, 0.680002, 0.67765}
3786 }
3787
3788 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3789 {
3790     linkcolor      = [rgb]{0,0,0},

```

```

3791     filecolor      = [rgb]{0,0,0},
3792     urlcolor       = [rgb]{0,0,0},
3793     menucolor      = [rgb]{0,0,0},
3794     runcolor       = [rgb]{0,0,0},
3795 %----- %-----
3796     linkbordercolor = [rgb]{0,0,0},
3797     filebordercolor = [rgb]{0,0,0},
3798     urlbordercolor  = [rgb]{0,0,0},
3799     menubordercolor = [rgb]{0,0,0},
3800     runbordercolor  = [rgb]{0,0,0}
3801 }
3802 </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	
\#	280, 782
\\$	279
\%	783
\-	2190
\.	543, 547, 549
@curropt commands:	
\@curropt:	2709, 2776
\[2190
\]	20, 21, 31, 32, 33, 41, 45, 55, 75, 82, 89, 96, 103, 118, 123, 124, 132, 133, 142, 143, 144, 145, 146, 153, 161, 281, 282, 932, 2711, 2778, 3118
_	541, 543, 547, 549
\]	2190
A	
\A	541, 2190
\Acrobatmenu	18, 174
\addcontentsline	13
\AddToDocumentProperties	389, 2185, 2286
\AddToHook	427, 440
\AddToHookNext	199
\advance	2641, 2716, 2760, 2761
allcolors (hypersetup key)	1051
\author	2
B	
\b	545
\begingroup	210, 278, 376, 2547, 2676, 2705, 2717, 2751, 3140, 3571, 3583, 3612, 3642
\belowpdfbookmark	3533
\bgroup	275, 376
\BOOKMARK	3536
bookmarkstype (hypersetup key)	13
bool commands:	
\bool_if:NTF	268, 297, 356, 700, 726, 745, 765, 776, 813, 878, 943, 1021, 1031, 1275, 1286
\bool_if:nTF	893, 1960, 1985, 3277, 3385
\bool_lazy_and:nnTF	424
\bool_lazy_or:nnTF	1305, 1354
\bool_new:N	204, 205, 527, 531, 535
\bool_set_true:N	536
bordercolormodel (hypersetup key)	13, 1072
box commands:	
\box_dp:N	637, 2861, 2912, 2947
\box_ht:N	636, 2860, 2911, 2946
\box_new:N	457, 538
\box_set_dp:Nn	2157
\box_set_ht:Nn	2164
\box_use:N	1292, 2863, 2921, 2950
\box_use_drop:N	1297
\box_wd:N	635, 2859, 2910, 2945
\l_tmpa_box	2857, 2859, 2860, 2861, 2863, 2896, 2910, 2911, 2912, 2921, 2943, 2945, 2946, 2947, 2950
C	
\catcode	279, 280
\char	2795, 2992, 3004
\chardef	159
\cite	33
clist commands:	
\clist_item:nn	2217
\clist_map_function:nN	125, 134
\clist_map_inline:nn	2350
\clist_map_variable:nNn	2708, 2775
color commands:	
\color_export:nnN	40, 432, 980, 1093
\color_select:n	988, 1024, 1296, 2902, 2906
\color_select:nn	994
\color_set:nn	2, 394, 1005
\color_set:nnn	2, 393, 1011
color names:	
hyp/annot/file	524
hyp/annot/link	524
hyp/annot/menu	524
hyp/annot/run	524
hyp/annot/url	524
colorfile (hypersetup key)	1051
colorlink (hypersetup key)	1051
colorlinks (hypersetup key)	1037
colormenu (hypersetup key)	1051
colorrun (hypersetup key)	1051
colorscheme (hypersetup key)	1, 1463
colorurl (hypersetup key)	1051
cs commands:	
\cs_generate_variant:Nn	156, 157, 158, 391, 456, 579, 810, 980, 997, 1014
\cs_gset:Npn	1267
\cs_gset_eq:NN	410, 421
\cs_if_exist:NTF	13
\cs_if_exist_p:N	425

<code>\ExplSyntaxOn</code>	1288
.... 8, 11, 2579, 2636, 2660, 3020,	2141
3192, 3213, 3266, 3335, 3373, 3459	1439
extension (hypersetup key)	1439
13, 1505	1439
F	1439
<code>\fbox</code>	1439
2814, 2902, 2906	1439
<code>\fi</code>	1439
7, 2535,	1439
2562, 2565, 2574, 2674, 2682, 2683,	1439
2714, 2732, 2735, 2736, 2742, 2745,	1439
2754, 2756, 2763, 2766, 3093, 3115,	1439
3125, 3127, 3133, 3137, 3144, 3145,	1439
3154, 3158, 3162, 3166, 3170, 3174,	1439
3178, 3182, 3186, 3190, 3207, 3211,	1439
3310, 3311, 3312, 3319, 3323, 3327,	1439
3329, 3331, 3350, 3354, 3358, 3362,	1439
3369, 3434, 3435, 3442, 3446, 3448,	1439
3469, 3470, 3477, 3481, 3483, 3548,	1439
3553, 3605, 3606, 3617, 3621, 3638	1439
file (hypersetup key)	1439
10, 1536	1439
file commands:	1439
<code>\file_input:n</code>	1439
9	1439
fileborderstyle (hypersetup key)	1439
14, 1165	1439
filecolor (hypersetup key)	1439
1051	1439
final (hypersetup key)	1439
1487	1439
<code>\fontencoding</code>	1439
2791, 2988, 3000	1439
<code>\fontfamily</code>	1439
2790, 2987, 2999	1439
<code>\fontseries</code>	1439
2792, 2989, 3001	1439
<code>\fontshape</code>	1439
2793, 2990, 3002	1439
fp commands:	1439
<code>\fp_eval:n</code>	1439
2115, 2143, 2159	1439
G	1439
<code>\gdef</code>	1439
281, 282	1439
<code>\GetDocumentProperties</code>	1439
3	1439
<code>\global</code>	1439
2641, 3083	1439
group commands:	1439
<code>\group_begin:</code>	1439
266, 295, 318,	1439
335, 354, 569, 706, 718, 732, 738,	1439
770, 799, 818, 866, 883, 925, 948,	1439
974, 1023, 1278, 1295, 2419, 2789,	1439
2839, 2882, 2930, 2963, 2986, 2998	1439
<code>\group_end:</code>	1439
309, 326,	1439
343, 371, 576, 714, 718, 750, 754,	1439
796, 799, 863, 866, 923, 925, 972,	1439
974, 1033, 1298, 1301, 2426, 2796,	1439
2875, 2923, 2956, 2993, 3005, 3015	1439
H	1439
<code>\hbox</code>	1439
376, 3557	1439
hbox commands:	1439
<code>\hbox_overlap_right:n</code>	1439
1292	1439
<code>\hbox_set:Nn</code>	1439
2857, 2896, 2943	1439
<code>\hbox_set:Nw</code>	1439
1279	1439
<code>\hbox_set_end:</code>	1439
1288	1439
<code>\hbox_set_to_wd:Nnn</code>	1439
2141	1439
hidefile (hypersetup key)	1439
hideurl (hypersetup key)	1439
hideurl (hypersetup key)	1439
hook commands:	1439
<code>\hook_gput_code:nnn</code>	1439
403, 414, 1017, 1027, 1271, 1282	1439
<code>\hook_new:n</code>	1439
392, 554, 649	1439
<code>\hook_new_pair:nn</code>	1439
261, 289, 313, 330, 348	1439
<code>\hook_use:n</code>	1439
265, 285, 294, 310, 317, 327, 334,	1439
344, 353, 372, 571, 655, 664, 711, 734	1439
<code>\href</code>	1439
4, 5, 19, 263	1439
<code>\hreflaunch</code>	1439
5	1439
<code>\hrefpdf</code>	1439
5, 19, 314, 384	1439
<code>\hrefrun</code>	1439
5, 331, 385	1439
<code>\hrefurl</code>	1439
5, 19, 291, 383	1439
<code>\hss</code>	1439
3562	1439
Hy internal commands:	1439
<code>\Hy_hyescapeform</code> ..	1439
3104, 3117, 3123	1439
<code>\Hy_hypSectionAnchor</code> ...	1439
3552, 3555	1439
HyField internal commands:	1439
<code>\HyField_hypAddToFields</code> .	1439
2541, 2568	1439
hyp commands:	1439
<code>\l_hyp_annot_colorfile_bool</code> ...	1439
525	1439
<code>\l_hyp_annot_colorlink_bool</code> ...	1439
525	1439
<code>\l_hyp_annot_colormenu_bool</code> ...	1439
525	1439
<code>\l_hyp_annot_colorrrun_bool</code>	1439
525	1439
<code>\l_hyp_annot_colorurl_bool</code>	1439
525	1439
<code>\l_hyp_annot_ocgcolorfile_bool</code> .	1439
529	1439
<code>\l_hyp_annot_ocgcolorlink_bool</code> .	1439
529	1439
<code>\l_hyp_annot_ocgcolormenu_bool</code> .	1439
529	1439
<code>\l_hyp_annot_ocgcolorrrun_bool</code> ..	1439
529	1439
<code>\l_hyp_annot_ocgcolorurl_bool</code> ..	1439
529	1439
<code>\l_hyp_current_dest_name_tl</code>	1439
648, 707, 733	1439
hyp internal commands:	1439
<code>\g__hyp_AcroForm_CoFields_prop</code> ..	1439
2505, 2523, 2592, 2594, 2609	1439
<code>\g__hyp_AcroForm_Fields_prop</code> ...	1439
2506, 2516, 2587	1439
<code>\l_hyp_annot_GoTo_bool</code> 700, 726, 745	1439
<code>\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</code>	1439
533	1439
<code>\l_hyp_annot_GoToR_bool</code>	1439
813	1439
<code>\l_hyp_annot_Launch_bool</code>	1439
878	1439

\l_hyp_annot_Named_bool	943	_hyp_href_pdf_aux:nn	320, 323
\c_hyp_annot_types_seq		\l_hyp_href_pdf_destination_tl	207, 218, 257, 325
.	469, 525, 529, 1051, 1352, 1450	\l_hyp_href_pdf_page_tl 208, 226, 843	
\l_hyp_annot_URI_bool	765	_hyp_href_run_aux:nn	337, 340
\g_hyp_bordercolormodel_str		\l_hyp_href_run_parameter_tl	209, 230, 342
.	434, 524, 1076, 1095	_hyp_href_url_aux:n	363, 366
_hyp_check_link_nesting:TF	609, 617,	_hyp_href_url_aux:nn	304, 306
.	619, 703, 729, 747, 768, 816, 881, 946	\l_hyp_href_url_encode_bool	204, 215, 253, 268, 297, 356
_hyp_citebordercolor_hook_-init:	399, 412, 421	_hyp_href_url_format:	210, 216, 258, 369
_hyp_citecolor_hook_init:	397, 401, 410	\l_hyp_href_url_ismap_bool	205, 229, 776
_hyp_clist_display:n	118, 125, 134	\l_hyp_href_url_protocol_tl	206, 217, 256, 308, 370
_hyp_color_select:n 40, 981, 981, 997		_hyp_if_outer_link:	603
_hyp_color_select_aux:wn	981, 985, 992	_hyp_if_outer_link:TF	619
_hyp_color_set:nn	40, 41, 397,	\l_hyp_link_Contents_tl	26, 35, 36, 504, 512, 519, 710, 781
.	399, 431, 998, 998, 1014, 1056, 1092	_hyp_link_goto_begin:nw	671, 712, 735
_hyp_color_set_aux:nwn	998, 1002, 1009	_hyp_link_goto_end:	693, 713, 749
\c_hyp_colorscheme_daleif_prop	3697	\g_hyp_linknestlevel_int	602, 605, 702, 720, 728, 756,
\c_hyp_colorscheme_henryford_-prop	3788	767, 801, 815, 868, 880, 926, 945, 975
\c_hyp_colorscheme_julian_prop	3712	g_hyp_linknestlevel_int	602
\c_hyp_colorscheme_phelype_prop	3773	\c_hyp_map_annot_hyp_prop	469
\c_hyp_colorscheme_szabolcsA_-prop	3742	\c_hyp_map_hyp_annot_prop	469, 1015,
\c_hyp_colorscheme_szabolcsB_-prop	3757	1079, 1119, 1146, 1154, 1165, 1192,
\c_hyp_colorscheme_tivv_prop	3727	1200, 1269, 1377, 1413, 1423, 1536
\l_hyp_dest_box	32,	_hyp_ocg_init: 1211, 1211, 1267, 1277	
.	538, 635, 636, 637, 2141, 2157, 2164	\l_hyp_optlang_regex 2189, 2190, 2219	
\l_hyp_dest_name_tpa_tl	462,	l_hyp_page/Trans	552
.	674, 675, 681, 685, 687, 690, 836, 849	_hyp_PageLabels_gpush: 586, 586, 599	
\l_hyp_dest_pdfremotestartview_-tl	488, 844, 1948, 1952	\l_hyp_para_tpa_tl 465, 892, 895, 906	
\g_hyp_dest_pdfstartpage_tl	488, 1959, 1961, 1968, 1986, 1993	\l_hyp_para_tpa_tl_uuu\l_hyp_-text_tpa_str_uuu\g_hyp_text_-tpa_str	462
\g_hyp_dest_pdfstartview_tl	488, 1961, 1968, 1979, 1983, 1986, 1993	_hyp_property_record:nn	449, 450, 456, 2645
\l_hyp_dest_pdfview_tl	523, 653, 662,	_hyp_secondoftwoewithopt:wnn	382, 383, 384, 385
.	2110, 2113, 2120, 2123, 2124, 2125,	_hyp_setup_info_date_key:nn	2272, 2305, 2306
.	2126, 2127, 2128, 2133, 2137, 2173	_hyp_setup_info_key:nn	2191,
\c_hyp_dest_startview_regex	539, 1946, 1977	2265, 2266, 2267, 2268, 2270, 2271
\c_hyp_dest_undefined_tl	468, 680, 681	_hyp_store_metadata:nn	386, 391, 1555, 2233,
_hyp_destination:nn	32, 622, 622, 653, 662	2251, 2285, 2300, 2309, 2325, 2393
\l_hyp_filename_tpa_tl	462,	_hyp_text_cleanup:N	559, 559, 573
.	819, 821, 826, 827, 832, 887, 888, 906		

\l__hyp_text_enc_dest_print_tl ..	colorlink	1051
..... 491, 835	colorlinks	1037
\l__hyp_text_enc_dest_tl	colormenu	1051
..... 32, 491, 629, 686	colorrun	1051
\l__hyp_text_enc_file_print_tl ..	colorscheme	1, 1463
..... 491, 886	colorurl	1051
\l__hyp_text_enc_info_print_tl ..	debug	1487
..... 491, 511, 518, 584	destlabel	13
\l__hyp_text_enc_para_print_tl ..	draft	1487
..... 491, 891	extension	13, 1505
\l__hyp_text_enc_uri_print_tl ...	file	10, 1536
..... 270,	fileborderstyle	14, 1165
273, 299, 302, 358, 361, 491, 773, 1547	filecolor	1051
__hyp_text_pdfstring:nnN	final	1487
.... 509, 516, 567, 567, 579, 581,	hidefile	1439
584, 627, 684, 771, 833, 884, 889, 1547	hidelink	1439
__hyp_text_pdfstring_info:nN ...	hidelinks	1439
.... 582, 582, 2222, 2226, 2245, 2255	hidemenu	1439
__hyp_text_purify:nN . 555, 555, 572	hiderun	1439
__hyp_text_string_from_unicode:nN	hideurl	1439
..... 563, 563, 574	hypertextnames	1505
\g__hyp_text_tmpa_str . 467, 575, 577	link	10, 1536
\l__hyp_text_tmpa_str	linkborder	14
..... 466, 572, 573, 574, 575	linkborderstyle	14, 1165
\l__hyp_tmpa_box 457, 1279, 1292, 1297	linkcolor	1051
\l__hyp_tmpa_int	linkfileprefix	1505
457	linktoc	1505
\l__hyp_tmpa_seq	linktocpage	1505
.... 457, 1946, 1948, 1977, 1979,	localanchorname	1505
2100, 2101, 2105, 2107, 2131, 2138,	menu	10, 1536
2139, 2140, 2149, 2151, 2161, 2166,	menuborder	14
2219, 2220, 2226, 2596, 2598, 2604	menuborderstyle	14, 1165
\l__hyp_tmpa_str	menucolor	1051
.... 457, 2222, 2226, 2228, 2230,	naturalnames	1505
2245, 2246, 2248, 2255, 2256, 2260	nested-links	10
\l__hyp_tmpa_tl	nesting	14
.. 457, 630, 634, 642, 1096, 1100,	ocgcolorfile	1305
1547, 1548, 1553, 1944, 1946, 1975,	ocgcolorlink	1305
1977, 2107, 2108, 2115, 2217, 2219	ocgcolorlinks	1305
\l__hyp_uri_tmpa_tl ... 462, 774, 775	ocgcolormenu	1305
hyp/anchor	ocgcolorrun	1305
hyp/annot/file (color name)	ocgcolorurl	1305
hyp/annot/link (color name)	pageanchor	1505
hyp/annot/menu (color name)	pdfauthor	2189
hyp/annot/run (color name)	pdfborder	14
hyp/annot/url (color name)	pdfborderstyle	14, 1165
hyp/text/pdfstring	pdfcreationdate	14, 2272
554	pdfcreator	2189
\hypercalcbp	pdfencoding	1477
12, 18, 162	pdfinfo	2339
\HyperDestNameFilter	pdfkeywords	2189
13, 628, 685	pdflang	14, 2178
\hypersetup 1, 2, 5, 9, 10, 13, 19, 68, 101, 181	pdflinkmargin	14
\hypersetup keys:	pdfmetadate	14, 2272
allcolors		
1051		
bookmarkstyle		
13		
bordercolormodel		
13, 1072		
colorfile		
1051		

pdfmoddate	14, 2272	
pdfproducer	2189	
pdfremotestartview	12	
pdfstartview	12	
pdfsubject	2189	
pdftitle	2189	
pdftrapped	2311	
pdfversion	1477	
pdfview	12, 2096	
plainpages	1505	
run	10, 1536	
runborder	14	
runborderstyle	14, 1165	
runcolor	1051	
unicode	1477	
url	10, 1536	
urlborder	14	
urlborderstyle	14, 1165	
urlcolor	1051	
verbose	1487	
hypertextnames (hypersetup key)	1505	
hypListbox internal commands:		
_hypListbox	2764, 2823	
hypRadio internal commands:		
_hypRadio	2749, 2771	
hypwritetorep internal commands:		
_hypwritetorep	3526	
I		
\ifcase	3310, 3434, 3469	
\ifcsname	3087	
\ifdim	2714, 2753, 2754	
\ifnum	3141, 3310, 3316, 3347, 3434, 3439, 3469, 3474, 3601, 3619, 3634	
\iftrue	3603	
\ifx	2545, 2683, 2743, 3118, 3130, 3134, 3151, 3155, 3159, 3163, 3167, 3171, 3175, 3179, 3183, 3187, 3204, 3208, 3311, 3312, 3320, 3324, 3331, 3351, 3355, 3362, 3363, 3435, 3443, 3470, 3478, 3551, 3613	
\immediate	2529, 2532	
int commands:		
\int_compare:nNnTF	605, 2105, 2131, 2815, 3233, 3286, 3396, 3418	
\int_compare_p:nNn	3279, 3387	
\int_eval:n	842	
\int_gdecr:N	720, 756, 801, 868, 926, 975	
\int_gincr:N	702, 728, 767, 815, 880, 945	
\int_max:nn	843	
\int_new:N	460, 602	
iow commands:		
\iow_newline:	3036, 3042, 3055	
K		
\kern	3558	
keys commands:		
\keys_define:nn	187, 195, 213, 251, 395, 613, 1037, 1053, 1060, 1072, 1081, 1106, 1121, 1140, 1167, 1186, 1309, 1323, 1335, 1358, 1364, 1379, 1408, 1439, 1452, 1463, 1477, 1487, 1492, 1505, 1517, 1538, 1543, 2096, 2178, 2193, 2236, 2274, 2289, 2307, 2311, 2339, 2391, 2397, 2410, 2430	
\l_keys_key_str	192, 2260, 2495	
\keys_set:nn	184, 221, 240, 267, 296, 319, 336, 355, 405, 416, 1469, 1476, 2316, 2343, 2346, 2347, 2348, 2349, 2420, 2495	
\keys_set_known:nn	2498	
\kvsetkeys	183, 2583	
L		
\label	9, 13	
\LayoutCheckField	2976	
\LayoutChoiceField	2746	
\LayoutPushButtonField	2852, 2869	
\LayoutTextField	2684	
\leavevmode	2685, 2782, 3556	
legacy commands:		
\legacy_if:nTF	169, 438, 445, 594, 2621, 2848, 2889, 2939, 2972, 3256	
\let	383, 384, 385, 722, 791, 803, 859, 870, 919, 928, 977, 2508, 2509, 2536, 2632, 2633, 2634, 2666, 2667, 2701, 2702, 2744, 3083, 3090, 3092, 3097, 3114, 3148, 3567, 3581, 3584, 3593, 3604, 3611, 3614, 3616, 3624, 3632	
link (hypersetup key)	10, 1536	
linkborder (hypersetup key)	14	
linkborderstyle (hypersetup key)	14, 1165	
linkcolor (hypersetup key)	1051	
linkfileprefix (hypersetup key)	1505	
linktoc (hypersetup key)	1505	
linktocpage (hypersetup key)	1505	
localanchorname (hypersetup key)	1505	
\long	7	
M		
\MakeButtonField	2857, 2872, 2896, 2943, 2954	
\MakeChoiceField	2832	
\MakeFieldObject	2499	
\MakeRadioField	2814	
\MakeTextField	2693	
\mbox	1289	

menu (hypersetup key)	10, 1536	\pdf_destination:nn	27, 156, 641
menuborder (hypersetup key)	14	\pdf_destination:nnnn	634
menuborderstyle (hypersetup key)	14, 1165	\pdf_emc:	1293, 1299
menucolor (hypersetup key)	1051	\pdf_link_user:nnn	2829
mode commands:		\pdf_name_from_unicode_e:n	246, 823, 953
\mode_if_horizontal:TF	624, 646	\pdf_object_if_exist:nTF	821
\mode_leave_vertical:	264, 293,	\pdf_object_new:n	1213,
316, 333, 352, 673, 785, 851, 908,		1214, 1215, 1216, 3023, 3024, 3025	
951, 2826, 2854, 2871, 2893, 2935, 2978		\pdf_object_ref:n	157, 832,
msg commands:		1219, 1221, 1243, 1244, 1247, 1251,	
\msg_error:nn	426, 2868, 2953	1256, 1261, 1266, 2614, 2616, 3081	
\msg_info:nnn	2203, 2208	\pdf_object_ref_last:	853, 2425
\msg_line_context:	83	\pdf_object_unnamed_write:nn	852, 2421
\g_msg_module_name_prop	12	\pdf_object_write:nnn	1217,
\msg_new:nnn	51, 58, 63, 67, 71, 78,	1223, 1233, 1245, 3026, 3068, 3075	
85, 92, 99, 106, 112, 119, 128, 138, 149		\pdf_pageobject_ref:n	
\msg_new:nnnn	15, 26, 37	158, 1968, 1993, 2653	
\msg_warning:nn	171, 1483	\pdf_string_from_unicode:nnN	565
\msg_warning:nnn	180, 677, 969	\pdf_version:	1629, 1748, 1777, 1811, 1826,
\msg_warning:nnnn		1852, 1877, 1912, 1928, 2033, 2060	
1327, 1368, 1581, 1625,		\pdf_version_compare:NnTF	
1659, 1682, 1744, 1773, 1807, 1822,		1619, 1710, 1732, 1768, 1801, 1816,	
1835, 1848, 1873, 1908, 1924, 1951,		1842, 1867, 1895, 1918, 2023, 2050	
1982, 2015, 2029, 2056, 2090, 2172		\pdf_version_compare_p:Nn	
\msg_warning:nnnnn	191,	897, 1306, 1355	
1399, 1433, 1525, 1601, 1638, 1725,		\pdf_version_major:	
1758, 1783, 1859, 1886, 1937, 2042,		168, 1307, 1330, 1356, 1372	
2069, 2333, 2437, 2456, 2475, 2486		\pdf_version_minor:	167, 1330, 1372
N			
naturalnames (hypersetup key)	1505	pdfannot commands:	
nested-links (hypersetup key)	10	\pdfannot_box:nnnn	2688, 2799, 2858, 2909, 2944, 3008
nesting (hypersetup key)	14	\pdfannot_box_ref_last:	2543, 2559
\newcommand	165, 3530, 3532, 3534	\pdfannot_dict_put:nnn	709, 781, 786, 853, 909, 954, 1097,
\newcount	2637, 2822	1132, 1156, 1178, 1202, 1384, 1415	
\NewDocumentCommand	2499	\pdfannot_dict_remove:nn	1087,
\NewExpandableDocumentCommand	382	1127, 1148, 1173, 1194, 1392, 1425	
\newlength	3549	\pdfannot_link:nnn	787, 854, 910, 955
\noexpand	2720, 2721, 2722, 2723, 3101	\pdfannot_link_goto_begin:nw	690
\nolinkurl	4	\pdfannot_link_goto_end:	695
O			
ocgcolorfile (hypersetup key)	1305	\pdfannot_link_margin:n	8, 1663
ocgcolorlink (hypersetup key)	1305	\c_pdfannot_link_types_seq	533
ocgcolorlinks (hypersetup key)	1305	pdfauthor (hypersetup key)	2189
ocgcolormenu (hypersetup key)	1305	\pdfbookmark	3535
ocgcolorrun (hypersetup key)	1305	pdfborder (hypersetup key)	14
ocgcolorurl (hypersetup key)	1305	pdfborderstyle (hypersetup key)	14, 1165
P			
pageanchor (hypersetup key)	1505	pdfcreationdate (hypersetup key)	14, 2272
\paperwidth	3	pdfcreator (hypersetup key)	2189
\PassOptionsToPackage	447, 1497, 1502	\pdfdest	11
pdf commands:		pdfdict commands:	
\pdf_bdcobject:nn	1291, 1294	\pdfdict_new:n	552, 759, 806, 872, 937

\pdfdict_put:nnn	238, 245, 553, 760, 761, 775, 778, 807, 808, 823, 829, 839, 849, 873, 874, 888, 903, 938, 939, 952, 1689, 1690, 1694, 1695, 2434, 2444, 2448, 2451, 2453, 2472, 2481, 2483
\pdfdict_remove:nn	235, 900, 1699, 1700
\pdfdict_use:n	786, 852, 909, 915, 954, 959, 2423
pdfencoding (hypersetup key)	1477
\pdfescapestring	3090
pdffile commands:	
\pdffile_embed_file:nnn	810, 824
PDFForm internal commands:	
\PDFForm_hypName	3129, 3132, 3136, 3139
pdfinfo (hypersetup key)	2339
pdfkeywords (hypersetup key)	2189
pdflang (hypersetup key)	14, 2178
pdflinkmargin (hypersetup key)	14
pdfmanagement commands:	
\pdfmanagement_add:nn	1672
\pdfmanagement_add:nnn	588, 1243, 1244, 1266, 1553, 1573, 1589, 1593, 1613, 1621, 1650, 1705, 1715, 1739, 1753, 1765, 1770, 1795, 1803, 1818, 1844, 1869, 1903, 1920, 1966, 1991, 2005, 2025, 2052, 2081, 2184, 2204, 2209, 2230, 2248, 2259, 2283, 2298, 2320, 2406, 2425, 2589, 2606, 2613, 2615, 2617, 2623
\pdfmanagement_if_active_p:	425
\pdfmanagement_remove:nn	1550, 1569, 1577, 1597, 1609, 1634, 1646, 1654, 1668, 1677, 1721, 1736, 1755, 1792, 1831, 1856, 1882, 1899, 1933, 1963, 1988, 2001, 2010, 2038, 2065, 2077, 2086, 2213, 2242, 2280, 2295, 2403, 2416, 2627
\pdfmanagement_show:n	2590
pdfmeta commands:	
\pdfmeta_standard_verify:nnTF	949, 2845, 2936
\pdfmeta_standard_verify:nTF	2619, 3197
pdfmetadate (hypersetup key)	14, 2272
pdfmoddate (hypersetup key)	14, 2272
pdfproducer (hypersetup key)	2189
pdfremotestartview (hypersetup key)	12
pdfstartview (hypersetup key)	12
\pdfstringdef	8, 9, 23, 28, 3146
\pdfstringdefDisableCommands	383, 384, 385
pdfsubject (hypersetup key)	2189
pdftitle (hypersetup key)	2189
pdftrapped (hypersetup key)	2311
pdfversion (hypersetup key)	1477
pdfview (hypersetup key)	12, 2096
pdfxform commands:	
\pdfxform_if_exist:nTF	2785, 2897, 2981
\pdfxform_new:nnn	2501, 2787, 2900, 2904, 2983, 2995
\pdfxform_ref:n	2809, 2810, 2916, 2917, 3228, 3229
\phantom	3004
\phantomsection	13, 19, 3620
plainpages (hypersetup key)	1505
prg commands:	
\prg_do_nothing:	410, 421
\prg_generate_conditional_variant:Nnn	2648
\prg_new_conditional:Npnn	603
\prg_return_false:	606
\prg_return_true:	607
prop commands:	
\prop_const_from_keyval:Nn	471, 479, 3682, 3697, 3712, 3727, 3742, 3757, 3773, 3788
\prop_gput:Nnn	12, 2516, 2523
\prop_if_empty:NTF	2592
\prop_item:Nn	2609
\prop_map_inline:Nn	1015, 1079, 1119, 1146, 1154, 1165, 1192, 1200, 1269, 1377, 1413, 1423, 1467, 1536, 2587, 2594
\prop_new:N	2505, 2506
property commands:	
\property_if_recorded:nn	2648
\property_if_recorded:nnTF	2651
\property_record:nn	453
\property_ref:nn	2655
\property_ref_undefined_warn:nn	2646
\protect	722, 791, 803, 859, 870, 919, 928, 977, 3551
\providecommand	163, 164, 2530, 2533, 2698, 3086
\ProvidesFile	3
R	
\ReadBookmarks	3540
\refstepcounter	12
regex commands:	
\regex_const:Nn	539
\regex_extract_once:NnN	2219
\regex_extract_once:NnNTF	1946, 1977
\regex_new:N	2189
\regex_set:Nn	2190

<code>\relax</code>	722, 791, 803, 859, 870, 919, 928, 977, 2701, 2702, 2743, 3083, 3130, 3134, 3204, 3208, 3311, 3312, 3320, 3324, 3351, 3355, 3435, 3443, 3470, 3478, 3604, 3613	<code>\str_gset:Nn</code>	1076
<code>\RemoveFromHook</code>	1559	<code>\str_gset_eq:NN</code>	575
<code>\RenewDocumentCommand</code>	174	<code>\str_head:n</code>	2322, 2327
<code>\RequirePackage</code>	6, 32, 442, 3538	<code>\str_if_eq:nnTF</code>	631, 2228, 2246, 2256, 3195
<code>run</code> (hypersetup key)	10, 1536	<code>\str_if_eq_p:nn</code>	895, 1307, 1356
<code>runborder</code> (hypersetup key)	14	<code>\str_lowercase:n</code>	2101, 2251, 2300, 2323, 2328
<code>runborderstyle</code> (hypersetup key)	14, 1165	<code>\str_new:N</code>	461, 466, 467, 524
<code>runcolor</code> (hypersetup key)	1051	<code>\str_set:Nn</code>	557
S		<code>\str_set_convert:Nnnn</code>	8
<code>\selectfont</code>	2794, 2991, 3003	<code>\str_set_eq:NN</code>	577
seq commands:		<code>\str_tail:n</code>	2323, 2328
<code>\seq_const_from_clist:Nn</code>	469	<code>\str_uppercase:n</code>	2316, 2322, 2327
<code>\seq_count:N</code>	2105, 2131	<code>\string</code>	2530, 2533, 2554, 2572, 3106, 3110, 3121
<code>\seq_get_right:NN</code>	2107	<code>\subpdfbookmark</code>	3531
<code>\seq_if_empty:NTF</code>	2220	sys commands:	
<code>\seq_item:Nn</code>	1948, 1979, 2101, 2149, 2151, 2161, 2166, 2226	<code>\c_sys_backend_str</code>	3493
<code>\seq_map_inline:Nn</code>	525, 529, 533, 1051, 1352, 1450, 2604	T	
<code>\seq_new:N</code>	459	TeX and L ^A T _E X 2 _ε commands:	
<code>\seq_put_right:Nn</code>	2138, 2139, 2140, 2596	<code>\@BOOKMARK</code>	3537
<code>\seq_set_split:Nnn</code>	2100	<code>\@CheckBox</code>	2959
<code>\seq_sort:Nn</code>	2598	<code>\@ChoiceMenu</code>	2699
<code>\setbox</code>	376	<code>\@Form</code>	2581
<code>\setlength</code>	3570, 3641	<code>\@PushButton</code>	2837
<code>\setpdflinkmargin</code>	8	<code>\@Reset</code>	2926
<code>\settowidth</code>	2713	<code>\@Submit</code>	2878
<code>\show</code>	2773	<code>\@TextField</code>	2664
socket commands:		<code>\@bookmarksopenlevel</code>	1564
<code>\socket_assign_plug:nn</code>	521, 522	<code>\@bsphack</code>	452
<code>\socket_new:nn</code>	505, 506	<code>\@chapter</code>	3593, 3594
<code>\socket_new_plug:nnn</code>	507, 514	<code>\@currDisplay</code>	2713, 2781
<code>\socket_use:nn</code>	780	<code>\@currValue</code>	2779, 2809
<code>\socket_use:nnn</code>	708	<code>\@currentHref</code>	3574, 3587, 3598, 3628, 3645
sort commands:		<code>\@curropt</code>	2698, 2708, 2711, 2775, 2778
<code>\sort_return_same:</code>	2602	<code>\@empty</code>	2508, 2509, 2545, 2632, 2633, 2634, 2666, 2667, 2683, 3151, 3155, 3159, 3163, 3167, 3171, 3175, 3179, 3183, 3187, 3331, 3362, 3363
<code>\sort_return_swapped:</code>	2601	<code>\@endForm</code>	2632
<code>\space</code>	2781, 3313, 3330, 3331, 3361, 3362, 3436, 3468, 3471	<code>\@esphack</code>	454
<code>\spacefactor</code>	624, 646	<code>\@firstofone</code>	2536, 3092, 3097, 3114
<code>\stockwidth</code>	3	<code>\@firstoftwo</code>	3635
str commands:		<code>\@for</code>	2709, 2776
<code>\c_colon_str</code>	932	<code>\@gobbletwo</code>	3584
<code>\c_hash_str</code>	782	<code>\@hyper@launch</code>	39, 932
<code>\c_percent_str</code>	783	<code>\@ifnextchar</code>	275
<code>\str_case:nnTF</code>	2199, 3493	<code>\@ifundefined</code>	3106, 3565, 3592, 3602
<code>\str_case_e:nnTF</code>	2101	<code>\@mainaux</code>	2529, 2532, 2552, 2571
<code>\str_compare:nNnTF</code>	2600	<code>\@mkboth</code>	3584
		<code>\@ne</code>	2641

\@nil	3104, 3117, 3123	\Fld@height	2669, 2690, 2693, 2704, 2762, 2801, 2814, 2832, 2881, 2929, 2965, 3010
\@part	3611, 3618	\Fld@keystroke@code	3151, 3153
\@pdfauthor	23	\Fld@listcount	2772, 2780, 2815, 2822
\@pdfborder	163	\Fld@mappingname	3134, 3136, 3208, 3210
\@pdfborderstyle	164	\Fld@maxlen	3418, 3420
\@savsf	624, 646	\Fld@menulength	2706, 2712, 2759
\@schapter	3581, 3582	\Fld@name 2665, 2700, 2838, 2961, 3129, 3203
\@secondoftwo	3637	\Fld@onblur@code	3171, 3173
\@sect	3632, 3633	\Fld@onclick@code	3296
\@spart	3624, 3625	\Fld@onenter@code	3183, 3185
\@ssect	3567, 3568	\Fld@onexit@code	3187, 3189
\@tempdima	2707, 2714, 2716, 2753, 2754, 2755, 2759, 2760, 2761, 2762	\Fld@onfocus@code	3167, 3169
\@tempdimb	2713, 2714	\Fld@onmousedown@code	3175, 3177
\@typeset@protect	3551	\Fld@onmouseup@code	3179, 3181
\BKM@color	435	\Fld@pageobjref 2634, 2649, 3086, 3218, 3271, 3304, 3340, 3379, 3428, 3463
\c@secnumdepth	3601, 3619, 3634	\Fld@radiosymbol	3359
\calc@bm@number	3543	\Fld@rotation	3233, 3235, 3279, 3286, 3288, 3310, 3316, 3318, 3347, 3349, 3387, 3396, 3398, 3434, 3439, 3441, 3469, 3474, 3476
\check@bm@number	3542	\Fld@submitflags	3455
\define@key	429	\Fld@validate@code	3159, 3161
\Fld@additionalactions	3193, 3263, 3297, 3333, 3370, 3417, 3457, 3486	\Fld@value 2667, 2683, 2702, 2743, 2744, 3416
\Fld@align	3222, 3308, 3383	\Fld@width	2668, 2682, 2689, 2693, 2703, 2742, 2753, 2755, 2800, 2814, 2832, 2850, 2880, 2891, 2928, 2940, 2964, 2974, 3009
\Fld@altname	3130, 3132, 3204, 3206	\Fld@X@additionalactions 3150, 3195, 3198
\Fld@annotflags	3217, 3270, 3303, 3339, 3378, 3427, 3462	\Form@action	3453
\Fld@annotnames	3202, 3219, 3272, 3305, 3341, 3380, 3429, 3464	\H@old@part	3611, 3622
\Fld@bcolor	3241, 3243, 3324, 3326, 3355, 3357, 3391, 3404, 3406	\H@old@schapter	3581, 3590
\fld@bcolor	3312	\H@old@sect	3632, 3643, 3652
\Fld@bordercolor	3237, 3239, 3281, 3290, 3292, 3311, 3320, 3322, 3351, 3353, 3389, 3400, 3402, 3435, 3443, 3445, 3470, 3478, 3480	\H@old@spart	3624, 3630
\Fld@borderstyle	3223, 3276, 3309, 3345, 3384, 3433, 3484	\H@old@ssect	3567, 3572
\Fld@borderwidth	2760, 2761, 3223, 3276, 3309, 3345, 3384, 3433, 3484	\href@	275, 281
\Fld@calculate@code	2545, 3163, 3165	\href@split	281, 282
\Fld@calculate@sortkey	2556	\Hy@abspage	598
\Fld@cbsymbol	3245	\Hy@activeanchorfalse	669
\Fld@charsize 2759, 3249, 3330, 3361, 3412, 3468	\Hy@activeanchortrue	660
\Fld@checkequals	2711, 2778	\Hy@AtBeginDocument	2510, 2526
\Fld@choices	3332	\Hy@bookmarkstyle	1565
\Fld@color	3250, 3252, 3331, 3362, 3413	\Hy@chapapp	3585, 3596
\Fld@default 2666, 2683, 2701, 2744, 2773, 2774, 2962, 3363, 3367, 3368, 3415	\Hy@colorlink	34
\Fld@flags	3221, 3274, 3307, 3343, 3382, 3431, 3466	\Hy@currentbookmarklevel	3524
\Fld@format@code	3155, 3157	\Hy@DisableOption	173
		\Hy@drafttrue	1496

\Hy@escapeform	2687, 2784, 2828, 2856,	\HyField@AddToFields .	2539, 2694,
2895, 2942, 2980, 3088, 3095, 3098		2816, 2833, 2864, 2920, 2949, 3013	
\Hy@escapestring	3090,	\HyField@afields	2508
3092, 3097, 3100, 3101, 3104, 3105,		\HyField@AfterAuxOpen	2510, 2536, 2569
3114, 3117, 3153, 3157, 3161, 3165,		\HyField@AuxAddToCoFields	
3169, 3173, 3177, 3181, 3185, 3189,		2521, 2533, 2554	
3245, 3296, 3359, 3415, 3416, 3453		\HyField@AuxAddToFields	
\Hy@finaltrue	1501	2514, 2530, 2572	
\Hy@FormObjects	2586, 3021, 3083	\HyField@cofields	2509
\Hy@gttemp	3146, 3148	\HyField@FlagsCheckBox	2971
\Hy@href	275	\HyField@FlagsChoice	2752
\Hy@href@nextactionraw	239	\HyField@FlagsPushButton	
\Hy@href@page	227	2847, 2887, 2938	
\Hy@linkfileprefix	1510	\HyField@FlagsRadioButton	2748
\Hy@linktoc	1521	\HyField@FlagsSubmit	2888
\Hy@MakeCurrentHref	200	\HyField@FlagsText	2681
\Hy@MakeCurrentHrefAuto		\HyField@PDFChoices	2825
3569, 3585, 3596, 3626, 3640		\HyField@SetKeys	2677, 2718,
\Hy@next	3595, 3604, 3607	2721, 2740, 2840, 2883, 2931, 2966	
\Hy@numberline	166, 3525	\HyPat@ObjRef .	3497, 3504, 3511, 3518
\Hy@org@chapter	3593, 3608	\hyper@link	283
\Hy@OutlineName	3541	\hyper@anchor	648
\Hy@OutlineRerunCheck	3539	\hyper@anchorend	
\Hy@pdfmajorversion	168	648, 3560, 3587, 3598, 3628	
\Hy@pdfminorversion	167	\hyper@anchorstart	
\Hy@pdfstringtrue	29, 570	648, 3560, 3587, 3598, 3628	
\Hy@pdfversion	3141	\hyper@link	34, 698
\Hy@pstringdef	580, 3121	\hyper@linkend	34, 743
\Hy@PutCatalog	586	\hyper@linkfile	325, 811
\Hy@raisedlink	3559, 3586, 3597, 3627	\hyper@linklaunch ..	38, 342, 876, 935
\Hy@RestoreLastskip	645	\hyper@linknamed	39, 176, 941
\Hy@ReturnAfterFi	7, 3122	\hyper@linkstart	34, 724
\Hy@safe@activetrue	784, 2548	\hyper@linkurl	308, 369, 763
\Hy@SaveLastskip	625	\hyper@normalise	
\Hy@secnum@part	3614, 3616, 3619	275, 304, 320, 337, 363, 376	
\Hy@SectionAnchorHref	3550, 3574, 3645	\HyPL@Labels	588, 598
\Hy@SectionHShift		\HyPL@storePageLabel	586
3549, 3558, 3570, 3641		\HyPsd@SanitizeForOutFile	3528
\Hy@StepCount	2712, 2780	\if@filesw	2528, 2551, 2570
\Hy@temp	2549, 2564, 3103, 3104	\ifFld@combo	2727, 2757
\Hy@unicodefalse	3143	\ifFld@hidden	2682, 2742
\Hy@VerboseAnchor	626	\ifFld@multiline	2670
\Hy@VerboseLinkStart	705, 731	\ifFld@popdown	2728
\Hy@VerboseLinkStop		\ifFld@radio	2724, 2747
715, 751, 794, 861, 921, 965		\ifHy@implicit	3545
\Hy@VersionChecked	159	\ifHy@pdfescapeform	3089, 3099
\Hy@WrapperDef	622	\ifHy@unicode	3142
\Hy@xspace@end	712, 793, 860, 920, 964	\kv@set@family@handler	179
\HyAnn@AbsPageLabel		\m@ne	3601, 3616
2633, 2639, 2686,		\OBJ@OCG@view	165
2783, 2827, 2855, 2894, 2941, 2979		\p@	2716
\HyAnn@Count	2637, 2638, 2641, 2642,	\pdf@ifdraftmode	2584
2643, 2644, 2645, 2646, 2651, 2655		\PDF@SetupDoc	160
		\PDFForm@Check	2980, 3012, 3214

