The latex-lab-bookmark package
Creating bookmarks

LaTeX Project*

v0.95 2026-02-18

Abstract

The following code implements a first draft for the creation of bookmarks with kernel methods.

1 Introduction

This package implements support for the PDF outline aka bookmarks based on the l3pdfoutline module in the PDF management. It is based on the bookmark and the hyperref package and mostly implements the same user commands, most importantly the main commands `\bookmark` and `\bookmarksetup`. [1] But it does not try to be a full replacement and to replicate every option in the same way as the other packages.

1.1 Usage

If `\DocumentMetadata` is used, the package should (currently) be loaded additionally with the `testphase` key:

```
\DocumentMetadata{testphase=bookmark}
\documentclass{...}
```

This will also load all the other tagging support code.

If only the PDF management is loaded the code should be loaded as package:

```
\RequirePackage{pdfmanagement}
\RequirePackage{latex-lab-testphase-bookmark}
```

If loaded it will replace the code used for bookmarks in the hyperref and bookmark packages. It is possible to manually create bookmarks with this package even if hyperref or bookmark are not loaded, but then the targets must be created manually too, e.g. with `\pdf_destination:nn`.

The package assumes that the input files are UTF-8 encoded!

*Initial implementation done by Ulrike Fischer

[1] There is no clash here, as documents can use only one bookmark system: mixing commands from more than one package or using manually the primitives is not really possible without breaking the tree.

1.2 User commands

\bookmark `\bookmark[⟨keyval options⟩]{⟨title text⟩}`

This is the main command. As keyval options it accepts the keys described below (which are more or less the same as the keys from the bookmark package). At least one action key is required!

\bookmarksetup `\bookmarksetup{⟨keyval options⟩}`

This can be used to change the behaviour of following bookmarks. It can also be used in the hook `bookmark`.

\bookmarksetupnext `\bookmarksetupnext{⟨keyval options⟩}`

This is a small wrapper around `\AddToHookNext{bookmark}{\bookmarksetup{#1}}` and provided for compatibility with the package bookmark. The hook it uses is executed after the optional argument of `\bookmark` has been processed and so allows to change the settings done there.

\bookmarkget ⋆ `\bookmarkget{⟨option name⟩}`

In the package bookmark this command allows to retrieve the current value of options. It is meant to be used inside the hook (as various options are set in a group it wouldn't give a really sensible output in other places). In bookmark the argument ⟨*option name*⟩ can take lots of values but in practice only `level` has been used to, e.g., make chapter bookmarks bold. So currently only support for `level` has been implemented and all other values returns an empty result.

\pdfbookmark
\subpdfbookmark
\belowpdfbookmark
\currentpdfbookmark

`\pdfbookmark[⟨level⟩]{⟨text⟩}{⟨name⟩}`

These commands are defined if hyperref is loaded and work as described in the hyperref documentation. They all not only create a bookmark with a GoTo link but also a destination `{⟨name⟩}` for this bookmark.

1.3 Configuring bookmarks connected to table of contents entries

Bookmarks can in a document be added by two methods: manually with commands like `\pdfbookmark` or `\bookmark` and automatically through heading commands. For the second method hyperref hooks into the `\addcontentsline` command which means that only headings which can at least potentially appear in a table of contents are added to the bookmarks — the standard starred headings currently not [2] The following keys allow to configure the bookmarks added with this second method[3].

bookmarksnumbered, numbered This allows to decide if the bookmarks show numbers. The code internally uses a socket `hyp/outline/numberformat`. The predeclared plugs of these socket redefine `\numberline`, `\booknumberline`, `\partnumberline` and `\chapternumberline` to gobble their argument (plug `false`), or to print it with a following space (plug `true`). Other plugs can be defined by users or packages and be assigned with `numbered=`⟨*plugname*⟩.

[2] This will change with the new template implementation of headings. In this implementation also starred headings can create bookmarks.

[3] As mentioned above hyperref is currently required for these bookmarks

bookmarkstype,lists By default hyperref and bookmark add only entries that go into the main table of contents (`toc`) to the outline. With these keys this can be changed. Differently to hyperref, the keys here accepts a list of extensions, so e.g. with `lists={toc,lof,lot}` it is possible to add entries from all three lists.

1.4 Hooks

Code to create bookmarks is typically hidden inside other commands. This makes it difficult to override settings done in their optional argument. There is therefore a hook `bookmark` to add additional options. It can, for example, change keys if you add a `\bookmarksetup{...}` to it. This hook is executed after the keys in the optional argument has been processed and so allows to overwrite them.

The package bookmark has actually two hooks: one which is filled when using `\bookmarksetupnext` and one which is filled by the `addtohook` key. This is different here: there is only one hook and there is no `addtohook` key. The hook should be filled with the standard commands, e.g., `\AddToHook`.

As an example the following code will make all bookmarks of level 0 (in a book chapters) bold:

```
\AddToHook{bookmark}
  {\ifnum\bookmarkget{level}=0 \bookmarksetup{bold}\fi}
```

1.5 Level keys

While bookmarks use relative levels, a user normally wants to set levels relative to document sectioning but implementing a suitable interface is not trivial as bookmarks can't skip levels. So e.g. if you use in this order a section, a chapter, a part, a chapter, a section, then the first section, chapter and the part will all be in the bookmarks in level 1, while the second chapter will be a child of the part and so in level 2, and the next section in level 3. This is different to the printed table of contents where both chapters and both sections look alike and so are perceived to be on the same respective level.

It is therefore not possible to map the level of a sectioning command (as seen by a user and by the `tocdepth` and `secnumdepth` counter) to a specific level in the bookmarks.

The code used here (and also in the bookmark package) tries to address this problem, by storing for every bookmark not only the actual *bookmark level* but also the (intended) *document level*. The *document level* of the last bookmark is the *current document level*. If a new bookmark then requests to be set in a specific document level, the code looks up the levels of the previous bookmarks and chooses the best fit as parent. For example, if there is a chapter (document level 0, bookmark level 1) with a subsection as child (document level 2, bookmark level 2) and the document now wants to insert a section bookmark (document level 1), it will insert it as child of the chapter and sibling of the subsection.

level The value is the requested *document level*. It can be given as integer expression (positive or negative) or with keywords like `section` if a corresponding `\toclevel@section` command exists that expands to an integer.

If the level value n is larger then the current document level, the bookmark is inserted as child. If the two values are equal, the bookmark is inserted as sibling. If n is smaller it looks up the parents until it finds a bookmark with a document

level equal to n or smaller to n. In the first case it is inserted as a sibling, in the second as a child of this bookmark.

In all cases the current document level is updated to n

bookmarksdepth, depth The value of these keys decides if a bookmark is shown at all. The value is a *document level* and accepts the same values as the `level` key. A bookmark is suppressed if its document level value is larger than the depth value.

rellevel This sets the level of a new bookmark in relation to the previous. It is in such a tree the natural way to set a level and quite easy to use and implement: a positive value (the actual value doesn't matter) produces a child, zero a sibling, a negative value goes up by the number of steps given but at most the number of steps needed to reach the root level. The value will also be used to update the current document level.

startatroot This goes up to the root, the same can be achieved by using `rellevel` with a large negative number (and this is also how it is implemented). By default it will set the current document level to zero, the value can be changed by using `startatroot=$n$`, the setting is suppressed if `keeplevel` has been set first.

1.6 Expanding commands in the bookmarks

Bookmarks can only show simple text. LaTeX commands must there be converted into something simple. hyperref uses here `\pdfstringdef`. The code here uses kernel methods, this can lead to different output, e.g., if math is used. The conversion code can be switched back to `\pdfstringdef` (if hyperref is loaded) by reassigning a socket plug (the default plug is called `default`):

```
\AssignSocketPlug{hyp/outline/title}{pdfstringdef}
```

Both methods understand the `\texorpdfstring` command.

1.7 Actions

The following actions are supported by this package (the list is quite similar to the bookmark package).

dest The value is a destination name. That is the action normally used in bookmarks. When creating a tagged PDF it will also link to the relevant structure destination.

page The value is a (absolute) page number. It creates a destination to the relevant page. It can be used together with the `gotor` key and will then create a link to the page in the external PDF. The `view` key can be used to control the view/zoom of the page. When creating a tagged PDF one should avoid this key for internal links as the page destination has no associated structure destination.

gotor The value is the file name of an external PDF (with extension).

named The value is one of `FirstPage`, `LastPage`, `NextPage`, `PrevPage`.

uri The value is url. The value is automatically percent encoded. The hash and percent chars must be properly escaped:

```
\bookmark[uri=https://www.example.com\#abc]{uri with hash}
\bookmark[uri=https://www.example.com\%abc]{uri with percent}
\bookmark[uri=https://www.köln.de]{uri with non-ascii char}
```

view The value is one of `Fit`, `FitB`, `FitH`, `FitV`, `FitBH`, `FitBV` which can be followed by a positive integer (separated by a space) or the keyword `null`. Or it can be `XYZ`. This can be followed (separated by spaces) by up to two positive integers and one decimarl 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%.

rawaction This is simply passed into the `/A` of the action. Check the PDF reference to find out what is possible ….

```
⟨@@=hyp⟩
⟨*package⟩

\ProvidesExplPackage{latex-lab-testphase-bookmark}{2026-04-21}{0.97a}
  {PDF bookmarks with kernel methods}%
```

2 Implementation

2.1 Variables

\g__hyp_outline_currentlevel_int
\g__hyp_outline_level_intarray

While bookmarks use relative levels, user set levels relative to document sectioning.

```
\int_new:N    \g__hyp_outline_currentlevel_int
\int_new:N    \l__hyp_outline_level_int
\intarray_new:Nn   \g__hyp_outline_level_intarray {2000} %TODO size??
\intarray_gset:Nnn \g__hyp_outline_level_intarray {1}{0}
\int_new:N    \l__hyp_outline_depth_int
\tl_new:N     \l__hyp_outline_title_tl
\tl_new:N     \l__hyp_outline_action_data_tl
\tl_new:N     \l__hyp_outline_action_dest_data_tl
\tl_new:N     \l__hyp_outline_action_view_tl
\bool_new:N   \l__hyp_outline_keeplevel_bool
\seq_new:N    \l__hyp_outline_type_seq
\bitset_new:Nn \l__hyp_outline_action_bitset
 {
   goto  = 1,
   gotor = 2,
   named = 3,
   uri   = 4,
   raw   = 5,
```

this two are destination types that apply to both goto and gotor

```
   namedest  = 6,
   pagedest  = 7
 }
\bitset_set_true:Nn  \l__hyp_outline_action_bitset { goto }

\int_new:N    \l__hyp_outline_tmpa_int
\int_new:N    \l__hyp_outline_rellevel_tmpa_int
```

```
\tl_new:N     \l__hyp_outline_parent_tmpa_tl
\seq_new:N    \l__hyp_outline_tmpa_seq
\tl_new:N     \l__hyp_outline_tmpa_tl
\str_new:N    \g__hyp_outline_tmpa_str
```

(*End of definition for* `\g__hyp_outline_currentlevel_int` *and* `\g__hyp_outline_level_intarray`.)

2.2 Converting text

The title text of the bookmark and other strings must be converted to strings suitable for a PDF. With hyperref one can use `\pdfstringdef` but in l3pdftools we also provide a native version which is used by default.

```
\socket_new:nn       {hyp/outline/title}{2}
\socket_new_plug:nnn {hyp/outline/title}{pdfstringdef}
 {
   \pdfstringdef#2{#1}
 }
```

This uses the pdfmanagement version of `\pdfstringdef` defined in l3pdftools. We use that socket by default. Ensure that a text without parentheses is produced by using `string-raw`!

```
\socket_new_plug:nnn {hyp/outline/title}{default}
 {
   \pdf_purify:nN {#1}#2
   \pdf_string_from_unicode:nVN {utf16/string-raw} #2 #2
 }
\socket_assign_plug:nn  {hyp/outline/title}{default}
```

2.3 Regex to check the view value

A similar regex is used in the generic driver of hyperref and should perhaps be shared.

```
\regex_const:Nn \c__hyp_outline_dest_startview_regex
  {
    \A\ *
     (?:
      (?:XYZ (?:\ +(?:(?:\d+|\d*\.\d+)|null)){3}\ )
      |
      (?:Fit\b|FitB\b)
      |
      (?:(?:FitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1})
      |
      (?:FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
     )
  }

\msg_new:nnn
  { hyp /outline }
  { invalid-view-value }
  {
    Invalid~value~'#1'~of~'#2'  \\
    is~replaced~by~'Fit'~\msg_line_context:.
  }
```

2.4 Messages

```
\msg_new:nnn {hyp/outline}{no-action}
  {
    bookmark~action~missing!\\
    Use~one~of~'dest',~'gotor',~'named',~'uri',~or~'rawaction'
  }
```

2.5 Formatting the bookmark

```
\socket_new:nn {hyp/outline/numberformat}{0}
\socket_new_plug:nnn {hyp/outline/numberformat}{false}
 {
   \text_declare_purify_equivalent:Nn\numberline\use_none:n
   \text_declare_purify_equivalent:Nn\booknumberline\use_none:n
   \text_declare_purify_equivalent:Nn\partnumberline\use_none:n
   \text_declare_purify_equivalent:Nn\chapternumberline\use_none:n
   \cs_set_eq:NN\numberline\use_none:n
   \cs_set_eq:NN\booknumberline\use_none:n
   \cs_set_eq:NN\partnumberline\use_none:n
   \cs_set_eq:NN\chapternumberline\use_none:n
 }
\cs_new:Npn \__hyp_outline_use_numberline:n #1 {#1\c_space_tl}
\socket_new_plug:nnn {hyp/outline/numberformat}{true}
 {
   \text_declare_purify_equivalent:Nn\numberline       \__hyp_outline_use_numberline:n
   \text_declare_purify_equivalent:Nn\booknumberline   \__hyp_outline_use_numberline:n
   \text_declare_purify_equivalent:Nn\partnumberline   \__hyp_outline_use_numberline:n
   \text_declare_purify_equivalent:Nn\chapternumberline\__hyp_outline_use_numberline:n
   \cs_set_eq:NN \numberline \__hyp_outline_use_numberline:n
   \cs_set_eq:NN\booknumberline   \__hyp_outline_use_numberline:n
   \cs_set_eq:NN\partnumberline   \__hyp_outline_use_numberline:n
   \cs_set_eq:NN\chapternumberline\__hyp_outline_use_numberline:n
 }
```

2.6 Key definitions

```
\keys_define:nn {hyp/outline}
 {
   ,bold .choice:
   ,bold / true .code:n = \bitset_set_true:Nn \l_pdfoutline_F_bitset {Bold}
   ,bold /false .code:n = \bitset_set_false:Nn\l_pdfoutline_F_bitset {Bold}
   ,bold .default:n = true
   ,italic .choice:
   ,italic / true .code:n = \bitset_set_true:Nn\l_pdfoutline_F_bitset {Italic}
   ,italic /false .code:n = \bitset_set_false:Nn\l_pdfoutline_F_bitset {Italic}
   ,italic .default:n = true
   ,numbered .code:n =
     { \socket_assign_plug:nn { hyp/outline/numberformat } {#1}}
   ,numbered .default:n = true
   ,numbered .initial:n = false
   ,bookmarksnumbered .meta:n = {numbered}
   ,color .tl_set:N = \l_pdfoutline_color_tl

   ,open .bool_set:N = \l_pdfoutline_open_bool
```

TODO: make this perhaps a bit safer ...

```
  ,depth .code:n =
   {
     \cs_if_exist:cTF {toclevel@#1}
       { \int_set:Nn \l__hyp_outline_depth_int { \use:c {toclevel@#1} } }
       { \int_set:Nn \l__hyp_outline_depth_int { #1 } }
   }
  ,bookmarksdepth .meta:n = { depth = #1 }
  ,depth .initial:n = {\int_use:N \c_max_int}
```

openlevel is relative to the bookmark levels, not some document level!

```
  ,openlevel .int_set:N = \l_pdfoutline_open_int
  ,bookmarksopenlevel .meta:n = { openlevel = #1 }
  ,level .code:n =
   {
     \cs_if_exist:cTF {toclevel@#1}
       { \int_set:Nn \l__hyp_outline_level_int { \use:c {toclevel@#1} } }
       { \int_set:Nn \l__hyp_outline_level_int { #1 } }
   }
  ,rellevel .code:n =
    {
      \int_set:Nn
        \l__hyp_outline_level_int
        { \g__hyp_outline_currentlevel_int + #1 }
    }
  ,keeplevel .bool_set:N = \l__hyp_outline_keeplevel_bool
  ,startatroot .code:n =
    {
      \int_set:Nn \l__hyp_outline_level_int { -1000 }
      \bool_if:NF \l__hyp_outline_keeplevel_bool
        {
          \int_gset:Nn \g__hyp_outline_currentlevel_int {#1}
          \bool_set_true:NF \l__hyp_outline_keeplevel_bool
        }
    }
  ,startatroot .default:n = 0
```

dest applies to goto or gotor actions.

```
  ,dest  .code:n =
    {
      \int_compare:nNnTF
       { \bitset_item:Nn \l__hyp_outline_action_bitset {gotor} } = {1}
       {
         \bitset_clear:N      \l__hyp_outline_action_bitset
         \bitset_set_true:Nn  \l__hyp_outline_action_bitset { gotor }
       }
       {
         \bitset_clear:N      \l__hyp_outline_action_bitset
         \bitset_set_true:Nn  \l__hyp_outline_action_bitset { goto }
       }
      \bitset_set_true:Nn  \l__hyp_outline_action_bitset { namedest }
      \tl_set:Nn\l__hyp_outline_action_dest_data_tl {#1}
    }
  ,dest .default:n = Doc-Start
  ,goto .code:n =
```

```
    {
      \bitset_clear:N      \l__hyp_outline_action_bitset
      \bitset_set_true:Nn  \l__hyp_outline_action_bitset { goto }
      \bitset_set_true:Nn  \l__hyp_outline_action_bitset { namedest }
      \tl_set:Nn\l__hyp_outline_action_dest_data_tl {#1}
    }
  ,gotor .code:n =
    {
      \int_compare:nNnTF
       { \bitset_item:Nn \l__hyp_outline_action_bitset {pagedest} } = {1}
       {
         \bitset_clear:N      \l__hyp_outline_action_bitset
         \bitset_set_true:Nn  \l__hyp_outline_action_bitset { pagedest }
       }
       {
         \bitset_clear:N      \l__hyp_outline_action_bitset
         \bitset_set_true:Nn  \l__hyp_outline_action_bitset { namedest }
       }
      \bitset_set_true:Nn  \l__hyp_outline_action_bitset  { gotor }
      \tl_set:Nn\l__hyp_outline_action_data_tl {#1}
    }
  ,named .choices:nn =
    {FirstPage, NextPage, PrevPage, LastPage}
    {
      \bitset_clear:N      \l__hyp_outline_action_bitset
      \bitset_set_true:Nn  \l__hyp_outline_action_bitset { named }
      \tl_set:Nn \l__hyp_outline_action_data_tl {/#1}
    }
```

page like dest applies to goto and gotor actions.

```
  ,page .code:n =
    {
      \int_compare:nNnTF
       { \bitset_item:Nn \l__hyp_outline_action_bitset {gotor} } = {1}
       {
         \bitset_clear:N      \l__hyp_outline_action_bitset
         \bitset_set_true:Nn  \l__hyp_outline_action_bitset { gotor }
       }
       {
         \bitset_clear:N      \l__hyp_outline_action_bitset
         \bitset_set_true:Nn  \l__hyp_outline_action_bitset { goto }
       }
      \bitset_set_true:Nn   \l__hyp_outline_action_bitset { pagedest }
      \tl_set:Nn \l__hyp_outline_action_dest_data_tl {#1}
    }
  ,page  .default:n    = 1
  ,rawaction .code:n =
   {
     \bitset_clear:N      \l__hyp_outline_action_bitset
     \bitset_set_true:Nn  \l__hyp_outline_action_bitset { uri }
     \tl_set:Nn\l__hyp_outline_action_data_tl {#1}
    }
  ,view .code:n =
    {
```

```
        \tl_set:Ne \l__hyp_outline_tmpa_tl {#1~null~null~null~}
        \exp_args:NNV
        \regex_extract_once:NnNTF
          \c__hyp_outline_dest_startview_regex
          \l__hyp_outline_tmpa_tl
          \l__hyp_outline_tmpa_seq
           {
            \tl_set:Ne \l__hyp_outline_action_view_tl {/\seq_item:Nn \l__hyp_outline_tmpa_seq {1}
           }
           {
             \msg_warning:nnnn {hyp/outline}{invalid-view-value}{#1}{view}
             \tl_set:Nn \l__hyp_outline_action_view_tl {Fit}
           }
      }
    ,uri .code:n =
     {
       \bitset_clear:N      \l__hyp_outline_action_bitset
       \bitset_set_true:Nn  \l__hyp_outline_action_bitset { uri }
       \pdf_purify:nN{#1}\l__hyp_outline_action_data_tl
       \pdf_string_from_unicode:nVN
         { utf8/URI } \l__hyp_outline_action_data_tl \l__hyp_outline_action_data_tl
     }
    ,lists .code:n =
      { \seq_set_from_clist:Nn \l__hyp_outline_type_seq { #1 } }
    ,lists .initial:n = toc
    ,bookmarkstype .meta:n = {lists={#1}}
  }
```

\bookmarksetup

```
\NewDocumentCommand\bookmarksetup{m}{\keys_set:nn{hyp/outline}{#1}}
```

(*End of definition for* \bookmarksetup*. This function is documented on page 2.*)

```
\NewHook{bookmark}
```

\bookmarksetupnext

```
\NewDocumentCommand\bookmarksetupnext{m}{\AddToHookNext{bookmark}{\bookmarksetup{#1}}}
```

(*End of definition for* \bookmarksetupnext*. This function is documented on page 2.*)

\bookmarkget

```
\cs_new:Npn\bookmarkget #1 {\use:c{__hyp_bookmarkget_#1:}}
\cs_new:Npn\__hyp_bookmarkget_level:{\int_use:N\l__hyp_outline_level_int}
```

(*End of definition for* \bookmarkget*. This function is documented on page 2.*)

\bookmark

```
\NewDocumentCommand\bookmark{O{}m}
  {
    \bool_if:NT \l_pdfoutline_active_bool
     {
       \group_begin:
       \keys_set:nn {hyp/outline} { #1 }
       \UseHook{bookmark}
```

```
      \socket_use:nnn{hyp/outline/title}{#2}\l__hyp_outline_title_tl
      \int_compare:nNnF {\l__hyp_outline_level_int} > {\l__hyp_outline_depth_int}
       {
         \__hyp_outline_bookmark_aux:
       }
      \group_end:
    }
  }
```

(*End of definition for* `\bookmark`. *This function is documented on page 2.*)

`\__hyp_outline_bookmark_aux:`

```
\cs_new_protected:Npn \__hyp_outline_bookmark_aux:
```

At first we calculate the rellevel from the requested level.

```
  {
    \__hyp_outline_calc_rellevel:NN \l__hyp_outline_level_int \l__hyp_outline_rellevel_tmpa_int
```

Update the global current level

```
    \bool_if:NF\l__hyp_outline_keeplevel_bool
     {
       \int_gset:Nn \g__hyp_outline_currentlevel_int { \l__hyp_outline_level_int }
     }
    \int_compare:nNnTF
      {
        \bitset_item:Nn \l__hyp_outline_action_bitset {goto} *
        \bitset_item:Nn \l__hyp_outline_action_bitset {namedest}
      } = {1}
      {
        \pdfoutline_goto:nee
          { \l__hyp_outline_rellevel_tmpa_int } %level
          { \l__hyp_outline_action_dest_data_tl }
          { \l__hyp_outline_title_tl            }
      }
      {
        \int_case:nnF { \bitset_to_arabic:N \l__hyp_outline_action_bitset }
          {
            { 65 } % goto (1) + pagedest (65)
            {
              \pdfoutline_action:nee
               {  \l__hyp_outline_rellevel_tmpa_int }
               { /S/GoTo~
                 /D
                  [
                   \pdf_pageobject_ref:n { \l__hyp_outline_action_dest_data_tl }~
                   \l__hyp_outline_action_view_tl
                  ]
               }
               { \l__hyp_outline_title_tl }
            }
            { 34 } % gotor (2) + namedest (32)
            {
              \pdfoutline_action:nee
```

```
              {  \l__hyp_outline_rellevel_tmpa_int }
              { /S/GoToR~
                /F ( \l__hyp_outline_action_data_tl ) %Todo check format
                /D ( \l__hyp_outline_action_dest_data_tl ) %Todo check format
              }
              { \l__hyp_outline_title_tl }
            }
            { 66 } % gotor (2) + pagedest (64)
            {
              \pdfoutline_action:nee
              {  \l__hyp_outline_rellevel_tmpa_int }
              { /S/GoToR~
                /F ( \l__hyp_outline_action_data_tl ) %Todo check format
                /D
                 [
                  \pdf_pageobject_ref:n { \l__hyp_outline_action_dest_data_tl }~
                  \l__hyp_outline_action_view_tl
                 ]
              }
              { \l__hyp_outline_title_tl }

            }
            {  4 }  % named (4)
            {
             \pdfoutline_action:nee
              {  \l__hyp_outline_rellevel_tmpa_int }
              { /S/Named~
                /N \l__hyp_outline_action_data_tl %TODO check formt
              }
              { \l__hyp_outline_title_tl }

            }
            {  8 } % uri (8)
            {
             \pdfoutline_action:nee
              {  \l__hyp_outline_rellevel_tmpa_int }
              { /S/URI~
                /URI \l__hyp_outline_action_data_tl %TODO check formt
              }
              { \l__hyp_outline_title_tl }
            }
            { 16 } % raw (16)
            {
              \pdfoutline_action:nee
              {  \l__hyp_outline_rellevel_tmpa_int }
              {
                \l__hyp_outline_action_data_tl %TODO check formt
              }
              { \l__hyp_outline_title_tl }
            }
          }
          { \msg_error:nn {hyp/outline}{no-action} }
      }
    \intarray_gset:Nnn
```

```
      \g__hyp_outline_level_intarray
      { \pdfoutline_id_ref_last: }
      { \g__hyp_outline_currentlevel_int }
  }  %
```

(*End of definition for* `\__hyp_outline_bookmark_aux:`.)

`\__hyp_outline_calc_rellevel:NN` This function computes the relative level.

```
\cs_new_protected:Npn\__hyp_outline_calc_rellevel:NN #1 #2 % #1 the requested level, #2 the retu
  {
    \int_compare:nNnTF
      { #1 }
       <
      { \g__hyp_outline_currentlevel_int }
```

if the requested level is lower, we must inspect the parents to find the best fit. starting with the previous bookmark

```
      {
        \int_set:Nn #2 { 0 }
```

get the first parent

```
        \tl_set:Ne \l__hyp_outline_parent_tmpa_tl
                   { \pdfoutline_parent_ref:n { \pdfoutline_id_ref_last: } }
```

get the document level of the first parent.

```
       \int_compare:nNnTF
         { \l__hyp_outline_parent_tmpa_tl } > { 0 }
         {
           \int_set:Nn \l__hyp_outline_tmpa_int
             {
               \intarray_item:Nn
                \g__hyp_outline_level_intarray
                {  \l__hyp_outline_parent_tmpa_tl }
             }
         }
         { \int_set_eq:NN \l__hyp_outline_tmpa_int #1 }
% if \l__hyp_outline_tmpa_int = #1
% stop, decr rellevel (this sets it as sibling of the parent)
% if \l__hyp_outline_tmpa_int < #1
% stop (this sets it as child of the parent as rellevel is unchanged)
% if \l__hyp_outline_tmpa_int > #1
% get next parent and continue.
        \int_do_while:nNnn
          { \l__hyp_outline_tmpa_int  }
           >
          { #1 }
          {
            \int_case:nn
             {
               \int_sign:n
                { \l__hyp_outline_tmpa_int-#1 }
             }
```

```
              {
                { 0 } { \int_decr:N #2 }
                { 1 }
                {
                  \int_decr:N #2
                  \tl_set:Ne \l__hyp_outline_parent_tmpa_tl
                    { \pdfoutline_parent_ref:n { \l__hyp_outline_parent_tmpa_tl } }
```

We need to check if we reached the root level.

```
                  \int_compare:nNnTF
                    { \l__hyp_outline_parent_tmpa_tl } > { 1 }
                    {
                      \int_set:Nn \l__hyp_outline_tmpa_int
                        {
                          \intarray_item:Nn
                          \g__hyp_outline_level_intarray
                          { \l__hyp_outline_parent_tmpa_tl }
                        }
                    }
                    {
                      % root
                      \int_set_eq:NN \l__hyp_outline_tmpa_int #1
                    }
                }
              }
          \int_if_zero:nT {\l__hyp_outline_tmpa_int-#1}{\int_decr:N #2}
        }
```

requested level >= current level is the easy case: we create either a sibling or a child:

```
        {
          \int_set:Nn #2
           { #1 - \g__hyp_outline_currentlevel_int }
        }
}
```

(*End of definition for* `\__hyp_outline_calc_rellevel:NN`.)

2.7 Replacing hyperref commands

```
\disable@package@load{bookmark}{\RequirePackage{hyperref}}
\DeclareHookRule{package/hyperref/after}%
  {latex-lab-testphase-bookmark}{voids}{latex-lab-testphase-sec-template}
\DeclareHookRule{package/hyperref/after}%
  {latex-lab-testphase-bookmark}{voids}{latex-lab-testphase-sec}
\DeclareHookRule{package/hyperref/after}%
  {latex-lab-testphase-bookmark}{voids}{latex-lab-testphase-toc}

\AddToHook{package/hyperref/after}
  {
    \RemoveFromHook{cmd/addcontentsline/before}[hyp]
    \AddToHookWithArguments{cmd/addcontentsline/before}[hyp/outline]
      { \__hyp_addcontentsline_bookmark:nnn {#1}{#2}{#3} }
    \providecommand\addcontentslinebookmark{}
```

```
    \providecommand\addcontentslinebookmarkOff{}
    \RenewCommandCopy\addcontentslinebookmark\__hyp_addcontentsline_bookmark:nnn
    \renewcommand\addcontentslinebookmarkOff
     {
       \bool_if:NTF \l_pdfoutline_active_bool
        {
          \def\addcontentslinebookmarkReset{\bool_set_true:N\l_pdfoutline_active_bool}
        }
        {
          \let\addcontentslinebookmarkReset\relax
        }
       \bool_set_false:N\l_pdfoutline_active_bool
     }
    \legacy_if:nF{Hy@bookmarks}{\bool_set_false:N\l_pdfoutline_active_bool}
    \renewcommand*{\pdfbookmark}[3][0]{%
    \bookmark[level=#1,dest={#3.#1}]{#2}%
    \hyper@anchorstart{#3.#1}\hyper@anchorend}
    \def\currentpdfbookmark#1#2#3{%
      \bookmark[rellevel=0,dest={#3.#1}]{#2}%
      \hyper@anchorstart{#3.#1}\hyper@anchorend}
    \def\subpdfbookmark#1#2#3{%
      \bookmark[rellevel=1,dest={#3.#1}]{#2}%
      \hyper@anchorstart{#3.#1}\hyper@anchorend}
    \def\belopdfbookmark#1#2#3{%
      \bookmark[keeplevel,rellevel=1,dest={#3.#1}]{#2}%
      \hyper@anchorstart{#3.#1}\hyper@anchorend}
```

This doesn't yet handle the package options. This must be done in hyperref.

```
    \keys_define:nn{hyp}
     {bookmarksdepth      .meta:nn = {hyp/outline}{depth=#1},
      bookmarksopen       .meta:nn = {hyp/outline}{open},
      bookmarksopenlevel  .meta:nn = {hyp/outline}{openlevel=#1},
      bookmarksnumbered   .meta:nn = {hyp/outline}{numbered}}
   }
\DeclareHookRule{package/hyperref/after}{latex-lab-testphase-bookmark}{after}{latex-
lab-testphase-toc}
```

`\__hyp_addcontentsline_bookmark:nnn` This is the command used at the begin of `\addcontentsline` which creates the bookmark.

```
\cs_new_protected:Npn \__hyp_addcontentsline_bookmark:nnn #1 #2 #3 %%#1 toc type, #2 level, #3 c
  {
    \seq_if_in:NeT \l__hyp_outline_type_seq { #1 }
      {
        \tl_if_empty:NT\@currentHref
          { \MakeLinkTarget[page] {} }
        \tl_if_exist:cF { toclevel@#2 }
          {
            \tl_new:c { toclevel@#2 }
            \tl_gset:cn { toclevel@#2 } { 0 }
            % message
          }
        \group_begin:
        \socket_use:n{hyp/outline/numberformat}
        \bookmark[level=#2,dest={\HyperDestNameFilter{\@currentHref}}]{#3}
```

```
      \group_end:
    }
  }
```

(*End of definition for* `\__hyp_addcontentsline_bookmark:nnn`.)

⟨/package⟩

⟨*latex-lab⟩
```
\ProvidesFile{bookmark-latex-lab-testphase.ltx}
        [\ltlabbookmarkdate\space v\ltlabbookmarkversion\space latex-lab wrapper bookmark]
\RequirePackage{latex-lab-testphase-bookmark}
```
⟨/latex-lab⟩