The sec-tagging package
Changes related to the tagging of sectioning commands

LaTeX Project[*]

v0.81 2023-02-12

Abstract

The following code implements a first draft for the tagging of sectioning commands.

1 Limitations

Sectioning commands are in not defined by the format but by the classes. Their implementation vary: some are defined with the help of `\@startsection`, some are like `\chapter` handcrafted, some build with the help of extension packages or as in the KOMA classes with class code that extends the `\@startsection` functionality.

The following code can therefore currently be used *only* with the standard classes or with classes which do not overwrite the changed definitions.

2 Introduction

Tagging of sectioning commands consist of two parts:

- The heading/title text of the section should be surrounded by a heading tag, typically `Hn` with some value of `n`. The number of the section command can optionally be put in a `Lbl`. The number of the `Hn` tag should reflect the “natural” level. So in an article `\section` will use `H1`, in a book `\chapter` will use `H1` and `\section` `H2`. Titles of `\part` are a bit out of this system as they are normally not part of the hierarchy: often only some chapters are grouped under a part. Their title is therefore tagged as `Title`.
- The whole section should normally be surrounded by a `Sect` tag. Parts should be surrounded by `Part`. It is a bit unclear if the headings should be inside or outside of these structures—the best practice guide puts them outside—but on the whole it sounds more logical to group the heading with the text inside the `Sect`. For the part this is actually required, as there can be only one `Title` in a structure, so the part title can’t be at the same level as the document `Title`.

  Starting such an enclosing `Sect` structure is rather easy, but closing it requires code in various place, for example the commands `\mainmatter`, `\backmatter`, `\frontmatter` and `\appendix` should typically close everything. Following sectioning commands should close all previous structures with a level equal or higher than their own level.

[*] Initial implementation done by Ulrike Fischer

3 Technical details and problems

The implementation has to take care of various details.

- As sections in LaTeX are not environments, the `<Sect>` structures can be wrongly nested with other structures. For example if a document puts a sectioning command into a list or a trivlist or a minipage then it can no longer close previous `<Sect>` structures correctly. The problem can be detected by checking the structure stack and a warning can be issued, but the author then has to close the structures manually before the list or minipage.

  Thus there have to be user interfaces to handle such cases. It should also be possible not to create all the `<Sect>` structures automatically but to tag only the headings so that the author can handle special cases manually.

- If hyperref is used, targets for links should be inserted, either with `\refstepcounter` or manually with `\MakeLinkTarget`. These targets must be in the correct structure for the structure destinations. They replace some of the current patches in hyperref.

- With lualatex the mc-commands set attributes *locally*, so the commands must be at the right grouping level.

3.1 Funktions and keys

```
\tag_tool:n
\tagtool
```

3.2 TODO

- A dedicated command to close a sectioning unit should be provided.
- A dedicated command to open a sectioning unit should be provided too.
- It should also be possible to suppress the sectioning unit in sectioning commands to allow e.g. to put an epigraph or similar in front.
- The number in `\part` and `\chapter` is currently not correctly tagged as a `Lbl` as this requires to redefine the internal (class dependant) commands too.

```
⟨*package⟩
```

4 Implementation

```
\ProvidesExplPackage {latex-lab-testphase-sec-tagging} {2022-10-25} {0.8}
  {Code related to the tagging of sectioning commands}
```

4.1 Surrounding by `Sect` structures

We use a stack to record the levels of the open `Sect`. The first item has level -100. A sectioning command will take a record from the stack. If its level is greater or equal it closes this structure and takes the next record from the stack. If the record has a smaller level then it puts it back and stops. The stack is compared with the main structure stack, if they don't match it means we can't safely close the `Sect` and so we issue a warning and do nothing.

4.1.1 Loading general kernel changes

[kernel?] Also loaded in the toc-tagging code.

```
\RequirePackage{latex-lab-kernel-changes}
```

```
⟨/package⟩
```

4.1.2 Glyphtounicode improvements

As lualatex runs with legacy encodings in the test files, we enable and load glyphtounicode. For the math we load additional definitions.

```
⟨*kernelchange⟩
\ifdefined\directlua
 \pdfvariable gentounicode =1
 \protected\def\pdfglyphtounicode {\pdfextension glyphtounicode }
 \input{glyphtounicode}
\fi
 \input{glyphtounicode-cmex}
⟨/kernelchange⟩
```

4.1.3 updating `\@currentHref`

[kernel?]

We must ensure that manual targets (e.g. in unnumbered sections) correctly update `\@currentHref`. For this we extend the kernel definition of `\MakeLinkTarget`

```
⟨*kernelchange⟩
\ExplSyntaxOn
\int_new:N\g__kernel_target_int
\RenewDocumentCommand\MakeLinkTarget{sO{}m}
 {%
  \ifvmode
    \special{}%
  \else
    \@savsf\spacefactor
    \smash{}%
    \spacefactor\@savsf
  \fi
  \int_gincr:N\g__kernel_target_int
  \tl_gset:Nx \@currentHref {target*.\int_use:N\g__kernel_target_int}
 }
\ExplSyntaxOff
⟨/kernelchange⟩
```

```
⟨*package⟩
```

4.1.4 Tagging commands

\g__tag_sec_stack_seq

The stack holds the tag and the level.

```
\seq_new:N   \g__tag_sec_stack_seq
\seq_gpush:Nn\g__tag_sec_stack_seq {{Document}{-100}}
```

\l__tag_sec_Sect_bool

This boolean controls if a Sect structure is opened.

```
\bool_new:N     \l__tag_sec_Sect_bool
\bool_set_true:N\l__tag_sec_Sect_bool
```

__tag_sec_begin:nn

This starts a sectioning structure. Currently the tag is fix, either Sect or Part, depending on the level, but this will perhaps change. The second argument is currently unused.

```
\cs_new_protected:Npn\__tag_sec_begin:nn #1 #2 %#1 level #2 keyval
  {
    \tag_struct_begin:n
      {
         tag= {\int_compare:nNnTF {#1}={-1}{Part}{Sect}}
        ,#2
      }
    \seq_gpush:Nx \g__tag_sec_stack_seq {{\g__tag_struct_tag_tl}{#1}}
  }
```

(*End definition for* __tag_sec_begin:nn.)

__tag_sec_end:n

```
\msg_new:nnn { tag } {wrong-sect-nesting}
  {
    The~structure~#1~can~not~be~closed.\\
    It~is~not~equal~to~the~current~structure~#2~on~the~main~stack
  }

\cs_new_protected:Npn\__tag_sec_end:n #1 % #1 level
  {
    \seq_get:NN \g__tag_sec_stack_seq \l__tag_tmpa_tl
    \int_compare:nNnT {#1}<{\exp_last_unbraced:NV\use_ii:nn\l__tag_tmpa_tl+1}
      {
        \seq_get:NN\g__tag_struct_tag_stack_seq \l__tag_tmpb_tl
        \exp_args:Nee
          \tl_if_eq:nnTF
            {\exp_last_unbraced:NV\use_i:nn\l__tag_tmpa_tl}
            {\exp_last_unbraced:NV\use_i:nn\l__tag_tmpb_tl}
            {
              \seq_gpop:NN \g__tag_sec_stack_seq \l__tag_tmpa_tl
              \tag_struct_end:
              \__tag_sec_end:n {#1}
            }
            {
              \msg_warning:nnxx {tag}{wrong-sect-nesting}
               { \exp_last_unbraced:NV\use_i:nn \l__tag_tmpa_tl }
               { \exp_last_unbraced:NV\use_i:nn \l__tag_tmpb_tl }
            }
      }
  }
```

(*End definition for* `\__tag_sec_end:n`*.*)

`\__tag_tool_para_split:` Runin-sectioning command must separate the heading from the following text.

```
\cs_new_protected:Npn \__tag_tool_para_split:
  {
    \tag_mc_end:
    \tag_struct_end:
    \tag_struct_begin:n{tag=\l__tag_para_tag_default_tl}
    \tag_mc_begin:n{}
    \__tag_setup_restore_para_default:
  }
```

(*End definition for* `\__tag_tool_para_split:`*.*)

`\__tag_setup_restore_para_default:` We change the para tagging in the sectioning code. This here restores the default. Currently it only resets the the tag, but perhaps more will be needed later.

```
\cs_new_protected:Npn \__tag_setup_restore_para_default:
  {
    \tl_set:Nn \l__tag_para_main_tag_tl {text-unit}
    \tl_set_eq:NN\l__tag_para_tag_tl\l__tag_para_tag_default_tl
  }
```

(*End definition for* `\__tag_setup_restore_para_default:`*.*)

`\__tag_sec_end_display:`

```
\cs_new_protected:Npn \__tag_sec_end_display:
  {
    \tag_struct_end: %P = Hn
    \__tag_setup_restore_para_default:
  }
```

(*End definition for* `\__tag_sec_end_display:`*.*)

Open sec structures should be closed at the end of the document. This should be done before tagpdf closes the Document structure.

```
\hook_gput_code:nnn{tagpdf/finish/before}{tagpdf/sec}{\__tag_sec_end:n{-10}}
\hook_gset_rule:nnnn {tagpdf/finish/before}{tagpdf/sec}{before}{tagpdf}
```

The commands `\mainmatter`, `\backmatter`, `\frontmatter` and `\appendix` close all `Sect` and `Part` structures.

```
\AddToHook{cmd/frontmatter/before}{\__tag_sec_end:n{-10}}
\AddToHook{cmd/mainmatter/before} {\__tag_sec_end:n{-10}}
\AddToHook{cmd/backmatter/before} {\__tag_sec_end:n{-10}}
\AddToHook{cmd/appendix/before}   {\__tag_sec_end:n{-10}}
```

4.2 Tagging tools

We need to provide user and package level commands

```
\cs_if_free:NT \tag_tool:n
 {
   \cs_new_protected:Npn \tag_tool:n #1
    {
      \tag_if_active:T { \keys_set:nn {tag / tool}{#1} }
    }
   \cs_set_eq:NN\tagtool\tag_tool:n
 }
\keys_define:nn { tag / tool}
  {
    ,sec-start-part .code:n =
      {
        \bool_if:NT\l__tag_sec_Sect_bool
          {
            \__tag_sec_end:n   {-1}
            \__tag_sec_begin:nn{-1}{tag=Part}
          }
         \tag_struct_begin:n{tag=part,title=#1}
```

We remap here the text-unit from the paragraph to NonStruct. It would be better to suppress it completly as with the other sectioning commands, but this would require to redefine `\@spart` and `\@part`, as there is the grouping, and these commands are all slightly different in the standard classes. So this is delayed to the time when sectioning commands are redefined with templates.

```
         \tl_set:Nn\l__tag_para_main_tag_tl {NonStruct}
         \tl_set:Nn\l__tag_para_tag_tl {Span}
      }
    ,sec-stop-part .code:n = {\__tag_sec_end_display:}
    ,sec-start-chapter .code:n =
     {
       \bool_if:NT\l__tag_sec_Sect_bool
         {
           \__tag_sec_end:n   {0}
           \__tag_sec_begin:nn{0}{tag=Sect}
         }
        \tag_struct_begin:n{tag=chapter,title=#1}
```

similar to part we remap to NonStruct for now ...

```
        \tl_set:Nn\l__tag_para_main_tag_tl {NonStruct}
        \tl_set:Nn\l__tag_para_tag_tl {Span}
     }
    ,sec-stop-chapter .meta:n = { sec-stop-part}
    ,sec-start .code:n = % #1 is a name like "section"
      {
        \bool_if:NT\l__tag_sec_Sect_bool
          {
            \__tag_sec_end:n    {\use:c{toclevel@#1}+0}
            \__tag_sec_begin:nn {\use:c{toclevel@#1}+0}{tag=Sect}
          }
        \tl_set:Nn\l__tag_para_tag_tl{#1}
      }
    ,sec-start .value_required:n = true
    ,sec-split-para .code:n = {\__tag_tool_para_split:}
    ,restore-para .code:n = {\__tag_setup_restore_para_default:}
    ,sec-stop .code:n =
      {
        \par\__tag_sec_end:n   {\use:c{toclevel@#1}+0}
      }
    ,sec-stop .value_required:n = true
    ,sec-add-grouping .bool_set:N = \l__tag_sec_Sect_bool
  }
```

5 Sectioning commands

5.1 \part and \chapter

\part and \chapter are defined by the classes. To tag them we redefine the user commands. This will probably break with various classes and with titlesec. The tagging inside relies on the para tagging. We do not yet use keyval in the optional argument, as this requires latex-dev and the naming of the keys and their key family is unclear.

```
\AddToHook{class/after}
 {
  \@ifundefined{chapter}
    {
```

This redefines \part in article class.

```
     \RenewDocumentCommand\part{ s O{#3} m }
      {
        \if@noskipsec \leavevmode \fi
        \par
        \addvspace{4ex}%
        \@afterindentfalse
```

This are the tagging commands needed at the begin. They open a Part structure and the structure for the title of the heading.

```
        % tagging start commands
        \tag_tool:n {sec-start-part=#2}
        % end tagging start commands
```

This adds a manual target if the part is unnumbered or starred. It replaces the hyperref patches.

```
        \bool_lazy_any:nT
          {
            { #1 }
            {
              \int_compare_p:nNn {\c@secnumdepth}<{-1}
            }
          }
          {
            \MakeLinkTarget[part]{}
          }
```

The main call to the underlying commands.

```
        \IfBooleanTF
          {#1}
          { \@spart {#3} }
          { \@part [#2]{#3} }
```

and now the closing command for the tagging of the title.

```
       \tag_tool:n {sec-stop-part}
      }
    }
```

Redefinitions for book and report

```
    {
     \RenewDocumentCommand\chapter{ s O{#3} m }
      {
        \if@openright\cleardoublepage\else\clearpage\fi
        \thispagestyle{plain}%
        \global\@topnum\z@
        \@afterindentfalse
```

This are the tagging commands needed at the begin. They open a Sect structure and the structure for the title of the heading.

```
        \tag_tool:n { sec-start-chapter= #2 }
```

This adds a manual target if the chapter is unnumbered or starred. It replaces the hyperref patches.

```
        \bool_lazy_any:nT
          {
            { #1 }
            {
              \int_compare_p:nNn {\c@secnumdepth}<{0}
            }
            {
              %in book target also needed in frontmatter
              \bool_lazy_and_p:nn
                { \cs_if_exist_p:c { @mainmattertrue } }
                { ! \legacy_if_p:n { @mainmatter } }
            }
          }
          {
            \MakeLinkTarget[chapter]{}
            %todo public function for this update!
            \tl_if_blank:VF \@currentHref
              {
                \prop_gput:Nxx \g__tag_struct_dest_num_prop {\@currentHref}{\tag_get:n{struct
              }
          }
```

The main call to the underlying commands.

```
        \IfBooleanTF
          {#1}
          { \@schapter {#3} }
          { \@chapter [#2]{#3} }
```

and now the closing command for the tagging of the title.

```
        \tag_tool:n {sec-stop-chapter}
      }
```

and similar for \part

```
    \RenewDocumentCommand\part{ s O{#3} m }
     {
       \if@openright
         \cleardoublepage
       \else
         \clearpage
       \fi
       \thispagestyle{plain}%
       \if@twocolumn
         \onecolumn
         \@tempswatrue
       \else
         \@tempswafalse
       \fi
       \null\vfil
```

This are the tagging commands needed at the begin. They open a Part structure and the structure for the title of the heading.

```
      \tag_tool:n {sec-start-part=#2}
```

This adds a manual target if the part is unnumbered or starred. It replaces the hyperref patches.

```
       \bool_lazy_any:nT
         {
           { #1 }
           {
             \int_compare_p:nNn {\c@secnumdepth}<{-1}
           }
           {
             %in book target also needed in frontmatter
             \bool_lazy_and_p:nn
               { \cs_if_exist_p:c { @mainmattertrue } }
               { ! \legacy_if_p:n { @mainmatter } }
           }
         }
         {
           \MakeLinkTarget[part]{}
           \tl_if_blank:VF \@currentHref
            {
              \prop_gput:Nxx \g__tag_struct_dest_num_prop {\@currentHref}{\tag_get:n{struct_
            }
         }
```

The main call to the underlying commands.

```
       \IfBooleanTF
         {#1}
         { \@spart {#3} }
         { \@part [#2]{#3} }
```

and now the closing command for the tagging of the title.

```
       \tag_tool:n{sec-stop-part}
     }
   }
}
```

5.2 Sectioning commands based on `\@startsection`

The tagging of relies again on the para tagging: we simply exchange the tag name by the one given as #1. This assumes that a tag with the name of the sectioning type is defined. We don't try to pass the title, this will be done together with the new keyval handling in the user command.

5.2.1 Hyperref code

hyperref has to insert anchors. If the sectioning is numbered this is done by `\refstepcounter` (and so in vmode). For unnumbered section hyperref injects the anchor in hmode before the text, it also inserts a kern to compensate the indent.

This means that the target of numbered and unnumbered sectioning commands differ, both regarding the location and in relation to the tagging structure: The anchor from the `\refstepcounter` is outside of the structure created by the heading title if the para tags are used, while the other anchors are inside and so the structure destinations are different.

We unify this by suppressing the anchor from the refstepcounter. Also we only go back if the indent is positive.

At first suppress all hyperref patches related to sectioning:

```
\def\hyper@nopatch@sectioning{}
```

`\@hyp@section@target@nnn` A simple internal command. There is no need for something public, as packages defining their own version of `\@startsection` will probably need something slightly different based on `\MakeLinkTarget`.

```
\cs_new_protected:Npn \@hyp@section@target@nnn #1 #2 #3 %#1 optarg #2 name/counter, #3 indent
 {
   \makebox[0pt][l]
    {
      \skip_set:Nn \@tempskipa {#3}
      \dim_compare:nNnF {\@tempskipa}<{0pt}{\kern-\@tempskipa}
      \MakeLinkTarget#1{#2}
    }
   \tl_if_blank:VF \@currentHref
    {
      \prop_gput:Nxx \g__tag_struct_dest_num_prop {\@currentHref}{\tag_get:n{struct_num}}
    }
 }
```

(*End definition for* `\@hyp@section@target@nnn`*. This function is documented on page* **??**.)

5.3 Adaption of the heading commands

We add to `\@startsection` the commands to open the `Sect` structure and to change the para tag.

```
\def\@startsection#1#2#3#4#5#6{%
  \if@noskipsec \leavevmode \fi
  \par
  \@tempskipa #4\relax
  \@afterindenttrue
  \ifdim \@tempskipa <\z@
    \@tempskipa -\@tempskipa \@afterindentfalse
  \fi
  \if@nobreak
    \everypar{}%
  \else
    \addpenalty\@secpenalty\addvspace\@tempskipa
  \fi
  \tag_tool:n { sec-start=#1}%new
  \@ifstar
    {\@ssect{#3}{#4}{#5}{#6}}%
    {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}}
```

To be able to correctly tag the number we need a special `\@hangfrom` variant. This is a bit tricky: As the paragraph starts after the `\setbox` luatex attributes are not set yet and numbers are unmarked if one doesn't pay attention. The code assumes that we are in vmode!

```
\cs_new_protected:Npn \@kernel@tag@hangfrom #1
  {
     \tagstructbegin{tag=\l__tag_para_tag_tl}
     \tagstructbegin{tag=Lbl}
     \setbox\@tempboxa
       \hbox
         {
```

In lua mode we have to set the attributes inside the box!

```
           \bool_lazy_and:nnT
            {\tag_if_active_p:}
            {\g__tag_mode_lua_bool}
            {\tagmcbegin{tag=Lbl}}
           {#1}
         }
```

We stop tagging now, to avoid that the `\noindent` triggers the paratagging. We do not disable paratagging completely, to avoid that the numbering goes wrong.

```
   \tag_stop:n{hangfrom}
   \hangindent \wd\@tempboxa\noindent
```

Restart tagging and insert the box.

```
   \tag_start:n{hangfrom}
   \tagmcbegin{}\box\@tempboxa\tagmcend\tagstructend\tagmcbegin{}}
```

This command is used to tag the numbers of runin. We do not try to avoid the empty container from the paratagging, this would require more changes.

```
\cs_new_protected:Npn \@kernel@tag@svsec
  {
    \tag_mc_end_push:
    \tag_struct_begin:n{tag=Lbl}
    \tag_mc_begin:n{}
    \@svsec
    \tag_mc_end:
    \tag_struct_end:
    \tag_mc_begin_pop:n{}
  }
```

`\@sect` is only changed to replace the hyperref patches and to use the new `\@kernel@tag@hangfrom` and `\@kernel@tag@svsec`

```
\def\@sect#1#2#3#4#5#6[#7]#8{%
```

```
  \ifnum #2>\c@secnumdepth
    \def\@svsec{\@hyp@section@target@nnn{[section]}{}{#3}}
  \else
    \LinkTargetOff
    \refstepcounter{#1}%
    \LinkTargetOn
    \protected@edef\@svsec{\@hyp@section@target@nnn{}{#1}{#3}\@seccntformat{#1}\relax}%
  \fi
  \@tempskipa #5\relax
  \ifdim \@tempskipa>\z@
    \begingroup
    \tagtool{para-flattened=true} % or \bool_set_true\l__tag_para_flattened_bool
      #6{%
         \ifnum #2>\c@secnumdepth
          \@hangfrom {\hskip #3\relax\@svsec}%
         \else
          \@kernel@tag@hangfrom{\hskip #3\relax\@svsec}%
         \fi
          \interlinepenalty \@M #8\@@par}%
    \endgroup
    \csname #1mark\endcsname{#7}%
    \addcontentsline{toc}{#1}{%
      \ifnum #2>\c@secnumdepth \else
        \protect\numberline{\csname the#1\endcsname}%
      \fi
      #7}%
  \else
    \def\@svsechd{%
      #6{\hskip #3\relax
      \ifnum #2>\c@secnumdepth
       \@svsec
      \else
       \@kernel@tag@svsec
      \fi  #8}%
      \csname #1mark\endcsname{#7}%
      \addcontentsline{toc}{#1}{%
        \ifnum #2>\c@secnumdepth \else
          \protect\numberline{\csname the#1\endcsname}%
        \fi
        #7}}%
  \fi
  \@xsect{#5}}
```

similar for `\@ssect`

```
\def\@ssect#1#2#3#4#5{%
  \@tempskipa #3\relax
  \ifdim \@tempskipa>\z@
    \begingroup
    \tagtool{para-flattened=true}
      #4{%
        \@hangfrom{\hskip #1\relax\@hyp@section@target@nnn{[section]}{}{#1}}%
          \interlinepenalty \@M #5\@@par}%
    \endgroup
  \else
    \def\@svsechd{#4{\hskip #1\relax\@hyp@section@target@nnn{[section]}{}{#3}\relax #5}}%
  \fi
  \@xsect{#3}}
```

At last `\@xsect` needs code in two places. For display headings it has to restore the default para code, for run in headings it has to separated the heading from the following text.

```
\def\@xsect#1{%
  \@tempskipa #1\relax
  \ifdim \@tempskipa>\z@
    \par \nobreak
    \vskip \@tempskipa
    \tag_tool:n {restore-para}
    \@afterheading
  \else
    \@nobreakfalse
    \global\@noskipsectrue
    \everypar{%
      \if@noskipsec
        \global\@noskipsecfalse
       {\setbox\z@\lastbox}%
        \clubpenalty\@M
        \begingroup \@svsechd \endgroup
        \unskip
        \tag_tool:n {sec-split-para}
        \@tempskipa #1\relax
        \hskip -\@tempskipa
      \else
        \clubpenalty \@clubpenalty
        \everypar{}%
      \fi}%
  \fi
  \ignorespaces}
⟨/package⟩
```

```
⟨*latex-lab⟩
\ProvidesFile{sec-tagging-latex-lab-testphase.ltx}
        [2023-02-12 v0.81 code related to the tagging of sectioning commands]
\RequirePackage{latex-lab-testphase-sec-tagging}
⟨/latex-lab⟩
```