The secnum package

Gau, Syu

Last Update: 2022/01/30

Abstract

The package secnum provides a marco `\setsecnum` which allows user to format section numbering intuitively.

Contents

A Example

This document uses the following setting of section numbering format.

```
\usepackage[tocdep=2]{secnum}
\setsecnum{A,:1.i}
```

B Usage

Before using the macro, load the package in preamble.

```
\usepackage{secnum}
```

1 Set numbering format

One can format the section numbering by using the marco `\setsecnum` in preamble.

`\setsecnum` `\setsecnum{`⟨*num format*⟩`}`

A typical ⟨*num format*⟩ is like this:

`A,:1.i`

It consists of some syntax abbrs of numbering formats, reffering the follows,

A	a	I	i	1
`\Alph`	`\alph`	`\Roman`	`\roman`	`\arabic`

and some separators delimiting them.

A separator can contain any character except above abbrs, the tokens "`{`", "`}`" and "`#`" (more precisely, explicit character tokens with category code 1 (begin-group) or 2 (end-group), and tokens with category code 6) and the space "␣".

Note that ⟨*num format*⟩ must end with an abbr.

TEXhackers note: This command will overwrite `secnumdepth` and `tocdepth`

2 Breaking the numbering

The comma "`,`" in above example is used as the breaking mark. When a separator contains a comma (in our example, `,:` between `A` and `1`), the shallower section levels (in our example, `\thesection`) will not be shown in titles of deeper levels (in our example, `\thesubsection` and `\thesubsubsection`). However, the numbering will appear in the reference labels with the given separator removing the comma. For instance, the next subsection is refered as B:3.

3 Package options

3.i tocdep

There is an option setting `tocdepth`, the table-of-contents depth manually.

`tocdep` `tocdep =` ⟨*integer*⟩

The ⟨*integer*⟩ refers to the table-of-contents depth, which should between 1 and 5.

TEXhackers note: If this option is used, then `\setsecnum` will not overwrite `tocdepth`.

3.ii breaking

Another option is used to change the breaking mark.

breaking `breaking = ⟨token⟩`

The ⟨*token*⟩ will be the breaking mark (the default is the comma "`,`"). It can be any character except above abbrs, the tokens "`{`", "`}`" and "`#`" (more precisely, explicit character tokens with category code 1 (begin-group) or 2 (end-group), and tokens with category code 6) and the space "␣".

C Process

The process of the macro `\setsecnum` can be explained as follows.

Step 1. The main function eats the input, saying `A,:1.i`, and stores it in a token list.

Step 2. Replace abbrs by macros. In our example, it results "`\Alph,:\arabic.\roman`"

Step 3. Split this token list into a sequence by macros. In our example, it results "`\Alph`", "`,:`", "`\arabic`", "`.`", and "`\roman`".

Step 4. Store those codes in indivial containers.

Step 5. Detect if there is `\thechapter`. Skip the chapter level if not. In our example, this is the case.

Step 6. Use the containers to redefine `\thesection`, `\thesubsection`, `\thesubsubsection` etc. In each step, detect if such level needs numbering and if there is a breaking mark in the container. In our example, the numbering formats will be redefined as

```
\renewcommand*{\thesection}{\Alph{section}}
\renewcommand*{\thesubsection}{\arabic{subsection}}
\renewcommand*{\thesubsubsection}{thesubsection.\roman{subsubsection}}
\makeatletter
  \renewcommand*{\p@subsection}{\Alph{section}:}
  \renewcommand*{\p@subsubsection}{\Alph{section}:}
\makeatother
```

D Implementation

The following is the implementation. Users can ignore.

1 Preparations

This package uses LaTeX3. Therefore, the packages expl3, xparse and l3keys2e are needed and should use `\ProvidesExplPackage` rather than `\ProvidesPackage`.

```
⟨*package⟩
⟨@@=syu⟩
\NeedsTeXFormat{LaTeX2e}
\RequirePackage{expl3,xparse,l3keys2e}
\ProvidesExplPackage{secnum}{2022/01/30}{}
  { An intuitive way to format section numbering }
```

`\g__syu_secnum_depth` This ⟨*integer*⟩ counts the depth of section levels.

```
\int_new:N \g__syu_secnum_depth
```

\g__syu_ifchapter_int This ⟨*integer*⟩ encodes if \thechapter is defined.

```
\int_new:N \g__syu_ifchapter_int
```

If \thechapter is defined, it is 1.

```
\if_cs_exist:N \thechapter
  \int_gset:Nn \g__syu_ifchapter_int 1
```

Otherwise, it is 0.

```
\else:
  \int_gset:Nn \g__syu_ifchapter_int 0
\fi:
```

\g__syu_secnum_bkm This variable is used to store the breaking mark.

```
\tl_gset:Nx \g__syu_secnum_bkmr {,}
```

We need the following variants

```
\cs_generate_variant:Nn \tl_if_in:NnTF { NV }
\cs_generate_variant:Nn \tl_remove_all:Nn { NV }
```

\g__syu_chapter_tl
\g__syu_chapter_section_tl
\g__syu_section_tl
\g__syu_section_subsection_tl
\g__syu_subsection_tl
\g__syu_subsection_subsubsection_tl
\g__syu_subsubsection_tl
\g__syu_subsubsection_paragraph_tl
\g__syu_paragraph_tl
\g__syu_paragraph_subparagraph_tl
\g__syu_subparagraph_tl

The following variables are used to store the individal formatting codes.

```
\tl_new:N \g__syu_chapter_tl
\tl_new:N \g__syu_chapter_section_tl
\tl_new:N \g__syu_section_tl
\tl_new:N \g__syu_section_subsection_tl
\tl_new:N \g__syu_subsection_tl
\tl_new:N \g__syu_subsection_subsubsection_tl
\tl_new:N \g__syu_subsubsection_tl
\tl_new:N \g__syu_subsubsection_paragraph_tl
\tl_new:N \g__syu_paragraph_tl
\tl_new:N \g__syu_paragraph_subparagraph_tl
\tl_new:N \g__syu_subparagraph_tl
```

2 Package option

```
\keys_define:nn { syu / options }{
```

tocdep Set the table-of-contents depth.

```
  tocdep .code:n = {
    \int_const:Nn \g__syu_tocdep {#1}
    \setcounter{tocdepth}{ \g__syu_tocdep }
  },
```

breaking Set the breaking mark used in ⟨*num format*⟩.

```
  breaking .code:n = {
    \tl_gset:Nx \g__syu_secnum_bkmr {#1}
  },

}
```

Passing keys to options.

```
\ProcessKeysOptions{ syu / options }
```

3 Main function

\setsecnum Here is the definition of the main function `\setsecnum`.

```
\DeclareDocumentCommand{\setsecnum}{m}{
```

Store the input in.

```
  \tl_set:Nn \l__syu_secnum_tl {#1}
```

Replace syntax abbrs by corresponding macros.

```
  \__syu_secnum_unabbr:N \l__syu_secnum_tl
```

Split into a sequence by macros.

```
  \__syu_split_by_macros:NNN
    \l__syu_secnum_tl \l__syu_secnum_seq \g__syu_secnum_depth
```

Read formatting information.

```
  \__syu_secnum_from_seq:N \l__syu_secnum_seq
```

Set the `secnumdepth` and `tocdepth`.

```
  \setcounter{secnumdepth}{
    \int_eval:n { \g__syu_secnum_depth - \g__syu_ifchapter_int }
  }
  \int_if_exist:NTF \g__syu_tocdep {
    \setcounter{tocdepth}{ \g__syu_tocdep }
  }{
    \setcounter{tocdepth}{
      \int_eval:n { \g__syu_secnum_depth - \g__syu_ifchapter_int }
    }
  }
```

Format numberings.

```
  \__syu_secnum:
}
```

4 Unabbravation

__syu_secnum_unabbr:N This function replace the abbrs in a ⟨*tl var*⟩ by expansions.

```
\cs_new_protected:Npn \__syu_secnum_unabbr:N #1 {
  \regex_replace_all:nnN {A} {\c{Alph}} #1
  \regex_replace_all:nnN {a} {\c{alph}} #1
  \regex_replace_all:nnN {I} {\c{Roman}} #1
  \regex_replace_all:nnN {i} {\c{roman}} #1
  \regex_replace_all:nnN {1} {\c{arabic}} #1
}
```

5 Split to sequence

__syu_split_by_macros:NNN This function splits a ⟨*tl var*⟩ into a ⟨*sequence*⟩ by macros and provides the number of macros it contains.

```
\cs_new_protected:Npn \__syu_split_by_macros:NNN #1 #2 #3 {
  \tl_set:Nn \l_tmpa_tl {S}
  \seq_clear:N #2
  \int_set:Nn #3 {0}
  \tl_map_inline:Nn #1 {
    \__syu_if_macro:nTF ##1 {
```

```
      \seq_put_right:NV #2 \l_tmpa_tl
      \tl_clear:N \l_tmpa_tl
      \tl_put_right:Nn \l_tmpa_tl ##1
      \seq_put_right:NV #2 \l_tmpa_tl
      \tl_clear:N \l_tmpa_tl
      \int_incr:N #3
    }{
      \tl_put_right:Nn \l_tmpa_tl ##1
    }
  }
}
```

But how to see if an ⟨*item*⟩ in the token list is a macro?

`\g__syu_macro_tl` This ⟨*tl var*⟩ stores the first five characters of the meaning of any macro, i.e. `macro` (watch out its catcode). The idea is to creat a ⟨*tl var*⟩ and then set its value to be the first five characters of its meaning.

```
\tl_new:N \g__syu_macro_tl
\tl_set:Nx \g__syu_macro_tl { \meaning \g__syu_macro_tl }
\tl_gset:Nx \g__syu_macro_tl { \tl_range:Nnn \g__syu_macro_tl {1}{5} }
```

`\__syu_if_macro:nT` `\__syu_if_macro:nF` `\__syu_if_macro:nTF` Then, define a conditional testing if the input is a macro. Note that I use `\if_meaning` rather than `\tl_if_eq:NNTF`.

```
\prg_new_protected_conditional:Npnn \__syu_if_macro:n #1 { T , F , TF }{
  \group_begin:
    \tl_set:Nx \l_tmpa_tl {\meaning #1}
    \tl_set:Nx \l_tmpa_tl {\tl_range:Nnn \l_tmpa_tl {1} {5}}
```

This is a trick to keep `\l_tmpa_tl` in the current local group

```
  \exp_after:wN
  \group_end:
```

while throwing the comparison result out.

```
    \if_meaning:w \l_tmpa_tl \g__syu_macro_tl
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
}
```

6 Read formatting info

`\__syu_secnum_from_seq:N` Read the formatting info from given ⟨*sequence*⟩.

```
\cs_new_protected:Npn \__syu_secnum_from_seq:N #1 {
```

Use `\tl_gset:Nx` since: 1, these data are global and 2: I need them eating the fully expanded results.

```
  \int_if_odd:nTF \g__syu_ifchapter_int {
    \tl_gset:Nx \g__syu_chapter_tl
      { \seq_item:Nn #1 { 2 } }
    \tl_gset:Nx \g__syu_chapter_section_tl
      { \seq_item:Nn #1 { 3 } }
    \tl_gset:Nx \g__syu_section_tl
      { \seq_item:Nn #1 { 4 } }
```

```
    \tl_gset:Nx \g__syu_section_subsection_tl
      { \seq_item:Nn #1 { 5 } }
    \tl_gset:Nx \g__syu_subsection_tl
      { \seq_item:Nn #1 { 6 } }
    \tl_gset:Nx \g__syu_subsection_subsubsection_tl
      { \seq_item:Nn #1 { 7 } }
    \tl_gset:Nx \g__syu_subsubsection_tl
      { \seq_item:Nn #1 { 8 } }
    \tl_gset:Nx \g__syu_subsubsection_paragraph_tl
      { \seq_item:Nn #1 { 9 } }
    \tl_gset:Nx \g__syu_paragraph_tl
      { \seq_item:Nn #1 { 10 } }
    \tl_gset:Nx \g__syu_paragraph_subparagrah_tl
      { \seq_item:Nn #1 { 11 } }
    \tl_gset:Nx \g__syu_subparagraph_tl
      { \seq_item:Nn #1 { 12 } }
  }{
    \tl_gset:Nx \g__syu_section_tl
      { \seq_item:Nn #1 { 2 } }
    \tl_gset:Nx \g__syu_section_subsection_tl
      { \seq_item:Nn #1 { 3 } }
    \tl_gset:Nx \g__syu_subsection_tl
      { \seq_item:Nn #1 { 4 } }
    \tl_gset:Nx \g__syu_subsection_subsubsection_tl
      { \seq_item:Nn #1 { 5 } }
    \tl_gset:Nx \g__syu_subsubsection_tl
      { \seq_item:Nn #1 { 6 } }
    \tl_gset:Nx \g__syu_subsubsection_paragraph_tl
      { \seq_item:Nn #1 { 7 } }
    \tl_gset:Nx \g__syu_paragraph_tl
      { \seq_item:Nn #1 { 8 } }
    \tl_gset:Nx \g__syu_paragraph_subparagrah_tl
      { \seq_item:Nn #1 { 9 } }
    \tl_gset:Nx \g__syu_subparagraph_tl
      { \seq_item:Nn #1 { 10 } }
  }
}
```

7 Formatting

`\__syu_secnum:` Formatting section numbering.

```
\cs_new:Nn \__syu_secnum: {
```

7.i Detect if there is `\thechapter`

When `\thechapter` is defined, start from it.

```
  \if_cs_exist:N \thechapter
    \renewcommand*{\thechapter}{ \g__syu_chapter_tl {chapter} }
```

Test if the numbering breaks before section.

```
    \tl_if_in:NVTF \g__syu_chapter_section_tl \g__syu_secnum_bkmr {
```

Remove the breaking marker.

```
      \tl_remove_all:NV
```

```
        \g__syu_chapter_section_tl \g__syu_secnum_bkmr
```
Format `\thesection`.
```
      \renewcommand*{\thesection}{ \g__syu_section_tl {section} }
```
Restore the `\p@`s.
```
      \makeatletter
        \renewcommand*{\p@section}{
          \thechapter\g__syu_chapter_section_tl
        }
        \renewcommand*{\p@subsection}{ \p@section}
        \renewcommand*{\p@subsubsection}{ \p@section }
        \renewcommand*{\p@paragraph}{ \p@section}
        \renewcommand*{\p@subparagraph}{ \p@section }
      \makeatother
    }{
```
Format `\thesection`.
```
      \renewcommand*{\thesection}{
        \thechapter\g__syu_chapter_section_tl
        \g__syu_section_tl {section}
      }
    }
```
Otherwise start from `\thesection`.
```
  \else:
    \renewcommand*{\thesection}{ \g__syu_section_tl {section} }
  \fi:
```

7.ii Subsections

Test if the subsections are needed to be numbered.
```
  \tl_if_empty:NF \g__syu_subsection_tl {
```
Test if the numbering breaks before subsection.
```
    \tl_if_in:NVTF \g__syu_section_subsection_tl \g__syu_secnum_bkmr {
```
Remove the breaking marker.
```
      \tl_remove_all:NV
        \g__syu_section_subsection_tl \g__syu_secnum_bkmr
```
Format `\thesubsection`.
```
      \renewcommand*{\thesubsection}{ \g__syu_subsection_tl {subsection} }
```
Restore the `\p@`s.
```
      \makeatletter
        \renewcommand*{\p@subsection}{
          \p@section\g__syu_section_tl{section}
          \g__syu_section_subsection_tl
        }
        \renewcommand*{\p@subsubsection}{ \p@subsection }
        \renewcommand*{\p@paragraph}{ \p@subsection}
        \renewcommand*{\p@subparagraph}{ \p@subsection }
      \makeatother
    }{
```

Format `\thesubsection`.

```
      \renewcommand*{\thesubsection}{
        \thesection\g__syu_section_subsection_tl
        \g__syu_subsection_tl {subsection}
      }
    }
}
```

7.iii Subsubsections

Test if the subsubsections are needed to be numbered.

```
  \tl_if_empty:NF \g__syu_subsubsection_tl {
```

Test if the numbering breaks before subsubsection.

```
    \tl_if_in:NVTF \g__syu_subsection_subsubsection_tl \g__syu_secnum_bkmr {
```

Remove the breaking marker.

```
      \tl_remove_all:NV
        \g__syu_subsection_subsubsection_tl \g__syu_secnum_bkmr
```

Format `\thesubsubsection`.

```
      \renewcommand*{\thesubsubsection}
        { \g__syu_subsubsection_tl {subsubsection} }
```

Restore the `\p@`s.

```
      \makeatletter
        \renewcommand*{\p@subsubsection}{
          \p@subsection\g__syu_subsection_tl {subsection}
          \g__syu_subsection_subsubsection_tl
        }
        \renewcommand*{\p@paragraph}{ \p@subsubsection}
        \renewcommand*{\p@subparagraph}{ \p@subsubsection }
      \makeatother
    }{
```

Format `\thesubsubsection`.

```
      \renewcommand*{\thesubsubsection}{
        \thesubsection\g__syu_subsection_subsubsection_tl
        \g__syu_subsubsection_tl {subsubsection}
      }
    }
}
```

7.iv Paragraphs

Test if the paragraphs are needed to be numbered.

```
  \tl_if_empty:NF \g__syu_paragraph_tl {
```

Test if the numbering breaks before paragraph.

```
    \tl_if_in:NVTF \g__syu_subsubsection_paragraph_tl \g__syu_secnum_bkmr {
```

Remove the breaking marker.

```
      \tl_remove_all:NV
        \g__syu_subsubsection_paragraph_tl \g__syu_secnum_bkmr
```

Format \theparagraph.

```
      \renewcommand*{\theparagraph}{ \g__syu_paragraph_tl {paragraph} }
```

Restore the \p@s.

```
      \makeatletter
        \renewcommand*{\p@paragraph}{
          \p@subsubsection\g__syu_subsubsection_tl {subsubsection}
          \g__syu_subsubsection_paragraph_tl
        }
        \renewcommand*{\p@subparagraph}{ \p@paragraph }
      \makeatother
    }{
```

Format \theparagraph.

```
      \renewcommand*{\theparagraph}{
        \thesubsubsection\g__syu_subsubsection_paragraph_tl
        \g__syu_paragraph_tl {paragraph}
      }
    }
}
```

7.v Subparagraphs

Test if the subparagraphs are needed to be numbered.

```
  \tl_if_empty:NF \g__syu_subparagraph_tl {
```

Test if the numbering breaks before paragraph.

```
    \tl_if_in:NVTF \g__syu_paragraph_subparagrah_tl \g__syu_secnum_bkmr {
```

Remove the breaking marker.

```
      \tl_remove_all:NV
        \g__syu_paragraph_subparagrah_tl \g__syu_secnum_bkmr
```

Format \thesubparagraph.

```
      \renewcommand*{\thesubparagraph}{ \g__syu_subparagraph_tl {subparagraph} }
```

Restore the \p@s.

```
      \makeatletter
        \renewcommand*{\p@subparagraph}{
          \p@paragraph\g__syu_paragraph_tl {paragraph}
          \g__syu_paragraph_subparagrah_tl
        }
      \makeatother
    }{
```

Format \thesubparagraph.

```
      \renewcommand*{\thesubparagraph}{
        \theparagraph\g__syu_paragraph_subparagrah_tl
        \g__syu_subparagraph_tl {subparagraph}
      }
    }
  }
}
```

⟨/package⟩