The secnum package

Gau, Syu

Last Update: 2020/01/01

Abstract

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

Contents

A Usage

Before using the macro, load the package in preamble.

```
\usepackage{secnum}
```

Then, one can format the section numbering by using the marco \setsecnum in preamble.

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

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

```
1.1.1
```

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, which can be any character except the abbrs and special characters such as barces "{}", comma ",", space "␣", etc.

B Process

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

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

Step 2. Replace abbrs by macros. In our example, it results "`\Roman.\arabic.\alph`"

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

Step 4. Store those codes in indivial containers.

Step 5. Use them to renew `\thesection`, `\thesubsection`, `\thesubsubsection` etc. provided there is no `\chapter`.

C Implementation

The following is the implementation. Users can ignore.

Preparations

This document class uses LaTeX3. Therefore, the packages expl3, xparse and l3keys2e are needed and should use `\ProvidesExplClass` rather than `\ProvidesClass`.

```
⟨*package⟩
⟨@@=syu⟩
\NeedsTeXFormat{LaTeX2e}
\RequirePackage{expl3}
\ProvidesExplPackage{secnum}{2020/01/01}{}
  { An intuitive way to format section numbering }
\RequirePackage{xparse}
```

`\l__syu_secnum_tl` `\l__syu_secnum_seq` The two variables are used to store the formatting information.

```
\tl_new:N \l__syu_secnum_tl
\seq_new:N \l__syu_secnum_seq
```

`\g__syu_chapter_tl` `\g__syu_section_tl` `\g__syu_subsection_tl` `\g__syu_subsubsection_tl` `\g__syu_paragraph_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_section_tl
\tl_new:N \g__syu_subsection_tl
\tl_new:N \g__syu_subsubsection_tl
\tl_new:N \g__syu_paragraph_tl
\tl_new:N \g__syu_subparagraph_tl
```

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

```
\int_new:N \g__syu_if_thechapter_int
```

If `\thechapter` is defined, it is 1.

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

Otherwise, it is 0.

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

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:NN \l__syu_secnum_tl \l__syu_secnum_seq
```

Read formatting information.

```
    \syu_secnum_from_seq:N \l__syu_secnum_seq
```

Set the `secnumdepth`.

```
    \setcounter{secnumdepth}{ \seq_count:N \l__syu_secnum_seq }
```

Format numberings.

```
    \syu_secnum:
  }
```

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
  }
```

Split to sequence

\syu_split_by_macros:NN This function split a ⟨*tl var*⟩ into a ⟨*sequence*⟩ by macros.

```
\cs_new_protected:Npn \syu_split_by_macros:NN #1 #2
  {
    \tl_clear:N \l_tmpa_tl
    \seq_clear:N #2
    \tl_map_inline:Nn #1
      {
        \tl_put_right:Nn \l_tmpa_tl ##1
        \__syu_if_macro:nT ##1
          {
            \seq_put_right:NV #2 \l_tmpa_tl
            \tl_clear:N \l_tmpa_tl
          }
      }
  }
```

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:
  }
```

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.

```
    \tl_gset:Nx \g__syu_chapter_tl
      { \seq_item:Nn #1 { \g__syu_if_thechapter_int } }
    \tl_gset:Nx \g__syu_section_tl
      { \seq_item:Nn #1 { 1 + \g__syu_if_thechapter_int } }
    \tl_gset:Nx \g__syu_subsection_tl
      { \seq_item:Nn #1 { 2 + \g__syu_if_thechapter_int } }
    \tl_gset:Nx \g__syu_subsubsection_tl
      { \seq_item:Nn #1 { 3 + \g__syu_if_thechapter_int } }
    \tl_gset:Nx \g__syu_paragraph_tl
      { \seq_item:Nn #1 { 4 + \g__syu_if_thechapter_int } }
    \tl_gset:Nx \g__syu_subparagraph_tl
      { \seq_item:Nn #1 { 5 + \g__syu_if_thechapter_int } }
  }
```

Formatting

`\syu_secnum:` Formatting section numbering.

```
\cs_new:Nn \syu_secnum:
  {
```

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

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

Otherwise start from `\thesection`.

```
    \else:
      \renewcommand*{\thesection}
        { \g__syu_section_tl {section} }
    \fi:
```

The rest levels.

```
      \renewcommand*{\thesubsection}
        { \thesection
          \g__syu_subsection_tl {subsection} }
      \renewcommand*{\thesubsubsection}
        { \thesubsection
          \g__syu_subsubsection_tl {subsubsection} }
      \renewcommand*{\theparagraph}
        { \thesubsubsection
          \g__syu_paragraph_tl {paragraph} }
      \renewcommand*{\thesubparagraph}
        { \theparagraph
          \g__syu_subparagraph_tl {subparagraph} }
  }
```

⟨/package⟩