The **secnum** package

Gau, Syu

*Last Update: 2022/01/30*

**Abstract**

The package **secnum** provides a macro \texttt{\setsecnum} which allows user to format section numbering intuitively.

**Contents**

A Example ........................................... 1

B Usage .................................................. 1
   1 Set numbering format ............................. 1
   2 Breaking the numbering ........................... 2
   3 Package options .................................. 2

C Process ............................................. 3

D Implementation .................................... 3
   1 Preparations ..................................... 3
   2 Package option ................................... 4
   3 Main function .................................... 4
   4 Unabbravation .................................... 5
   5 Split to sequence ................................ 5
   6 Read formatting info ............................ 6
   7 Formatting ....................................... 7

A Example

This document uses the following setting of section numbering format.

\texttt{\usepackage[tocdep=2]{secnum}}

\texttt{\setsecnum{A,:1.i}}

B Usage

Before using the macro, load the package in preamble.

\texttt{\usepackage{secnum}}
1 Set numbering format

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

\setsecnum\setsecnum{(\textit{num format}})

A typical \textit{⟨num format⟩} is like this:

\textit{A,:1.i}

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

<table>
<thead>
<tr>
<th>A</th>
<th>a</th>
<th>I</th>
<th>i</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{\Alph}</td>
<td>\textit{\alph}</td>
<td>\textit{\Roman}</td>
<td>\textit{\roman}</td>
<td>\textit{\arabic}</td>
</tr>
</tbody>
</table>

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 “\textit{\textstyle{\text{φ}}}”.

Note that \textit{⟨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, \textit{,:} between \textit{A} and \textit{1}), the shallower section levels (in our example, \textit{\thesection}) will not be shown in titles of deeper levels (in our example, \textit{\thesubsection} and \textit{\thesubsubsubsection}). However, the numbering will appear in the reference labels with the given separator removing the comma. For instance, the next subsection is referred as \textit{B:3}.

3 Package options

3.1 tocdep

There is an option setting \textit{tocdepth}, the table-of-contents depth manually.

\texttt{tocdep \texttt{tocdep = (integer)}}

The \textit{(integer)} refers to the table-of-contents depth, which should between 1 and 5.

\TeXhackers note: If this option is used, then \texttt{\setsecnum} will not overwrite \texttt{tocdepth}.

3.ii breaking

Another option is used to change the breaking mark.
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 \thesecnum, \thesubsecnum, \thesubsubsecnum 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*{\thesecnum}{\Alph{secnum}}
\renewcommand*{\thesubsecnum}{\arabic{subsection}}
\renewcommand*{\thesubsubsecnum}{\roman{subsubsection}}
\makeatletter
\renewcommand*{\p@subsection}{\Alph{section}:}
\renewcommand*{\p@subsubsecnum}{\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.

\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 \chapter{} is defined.
\int_new:N \g__syu_ifchapter_int
If \chapter{} 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_section_tl
\g__syu_subsection_tl
\g__syu_subsubsection_tl
\g__syu_paragraph_tl
\g__syu_paragraph_subparagraph_tl
\g__syu_subparagraph_tl
The following variables are used to store the individual 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
\tl_new:N \g__syu_ifchapter_int
\tl_new:N \g__syu_secnum_bkm
\tl_gset:Nx \g__syu_secnum_bkmr {,}
\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 dept.
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

Here is the definition of the main function \setsecnum.

\DeclareDocumentCommand{\setsecnum}{m}{
\tl_set:Nn \l__syu_secnum_tl {#1}
Replace syntax abbrs by corresponding macros.
\__syu_secnum_unbr: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 \texttt{secnumdepth} and \texttt{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_unbr:n
This function replace the abbrs in a \texttt{tl var} by expansions.
\cs_new_protected:Npn \__syu_secnum_unbr: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 \texttt{tl var} into a \texttt{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 \l__syu_split_by_macros:NNN \g__syu_split_by_macros_depth
\__syu_if_macro:nTF #1 {\__syu_split_by_macros:NNN #1 #2 #3}
But how to see if an item in the token list is a macro? This \langle \text{tl var} \rangle stores the first five characters of the meaning of any macro, i.e. \texttt{macro} (watch out its catcode). The idea is to create a \langle \text{tl var} \rangle and then set its value to be the first five characters of its meaning.

\begin{verbatim}
\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} }
\end{verbatim}

Then, define a conditional testing if the input is a macro. Note that I use \texttt{\if_meaning} rather than \texttt{\tl_if_eq:NNTF}.

\begin{verbatim}
\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} }
\exp_after:wN \group_end:
\if_meaning:w \l_tmpa_tl \g__syu_macro_tl \prg_return_true:
\else:
\prg_return_false:
\fi:
\end{verbatim}

This is a trick to keep \texttt{\l_tmpa_tl} in the current local group while throwing the comparison result out.

\begin{verbatim}
\exp_after:wN \\if_meaning:w \l_tmpa_tl \g__syu_macro_tl \prg_return_true:
\else:
\prg_return_false:
\fi:
\end{verbatim}

6 Read formatting info

\begin{verbatim}
\cs_new_protected:Npn \__syu_secnum_from_seq:N { }
\int_if_odd:nTF \g__syu_ifchapter_int { }
\tl_gset:Nx \g__syu_ifchapter_tl { \seq_item:Nn \g__syu_ifchapter_tl {1} {2} }
\tl_gset:Nx \g__syu_ifchapter_section_tl { \seq_item:Nn \g__syu_ifchapter_section_tl }{3} }
\tl_gset:Nx \g__syu_ifchapter_section_tl { \seq_item:Nn \g__syu_ifchapter_section_tl }{4} }
\end{verbatim}

6 Read formatting info from given \texttt{sequence}.

\begin{verbatim}
\cs_new_protected:Npn \__syu_secnum_from_seq:N { }
\int_if_odd:nTF \g__syu_ifchapter_int { }
\tl_gset:Nx \g__syu_ifchapter_tl { \seq_item:Nn \g__syu_ifchapter_tl {1} {2} }
\tl_gset:Nx \g__syu_ifchapter_section_tl { \seq_item:Nn \g__syu_ifchapter_section_tl {1} {3} }
\tl_gset:Nx \g__syu_ifchapter_section_tl { \seq_item:Nn \g__syu_ifchapter_section_tl }{4} }
\end{verbatim}

6 Read formatting info from given \texttt{sequence}.

\begin{verbatim}
\cs_new_protected:Npn \__syu_secnum_from_seq:N { }
\int_if_odd:nTF \g__syu_ifchapter_int { }
\tl_gset:Nx \g__syu_ifchapter_tl { \seq_item:Nn \g__syu_ifchapter_tl }{2} }
\tl_gset:Nx \g__syu_ifchapter_section_tl { \seq_item:Nn \g__syu_ifchapter_section_tl }{3} }
\tl_gset:Nx \g__syu_ifchapter_section_tl { \seq_item:Nn \g__syu_ifchapter_section_tl }{4} }
\end{verbatim}
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.
\ifcs_exist:N \thechapter
\renewcommand*{\thechapter}{ \g__syu_chapter_tl {chapter} }
\fi

Test if the numbering breaks before section.
\tl_if_in:NVT \__syu_chapter_section_tl \__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
}\makeatother
\renewcommand*{\p@subsection}{ \p@section }
\renewcommand*{\p@subsubsection}{ \p@section }
\renewcommand*{\p@paragraph}{ \p@section }
\renewcommand*{\p@subparagraph}{ \p@section }
\makeatother
\fi:

Format \thesection.
\renewcommand*{\thesection}{
  \thechapter\g__syu_chapter_section_tl
  \g__syu_section_tl {section}
}\makeatother

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
}\makeatother
\renewcommand*{\p@subsubsection}{ \p@subsection }
\renewcommand*{\p@paragraph}{ \p@subsection }
\renewcommand*{\p@subparagraph}{ \p@subsection }
\makeatother
\}

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}{ 
\thesubsection\g__syu_subsection_subsubsection_tl 
\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
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_subparagraph_tl \g__syu_secnum_bkmr {
Remove the breaking marker.
\tl_remove_all:NV
\g__syu_paragraph_subparagraph_tl \g__syu_secnum_bkmr

Format \thesubparagraph.
\renewcommand*{\thesubparagraph}{ \g__syu_subparagraph_tl \g__syu_paragraph_tl {paragraph}}
Restore the \p@s.
\makeatletter
\renewcommand*{\p@subparagraph}{ \p@paragraph \g__syu_paragraph_tl {paragraph}}
\makeatother

Format \thesubparagraph.
\renewcommand*{\thesubparagraph}{ \g__syu_subparagraph_tl \g__syu_paragraph_tl {paragraph}}
}