The secnum package

Gau, Syu

Last Update: 2021/08/28

Abstract
The package secnum provides a macro \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 Unabbrevation .................................................................................. 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.
\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{⟨num format⟩}

A typical ⟨num format⟩ is like this:

A,1.i

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

A a I i l
\Alph \alph \Roman \roman \arabic

and some separators delimiting them.

The separators 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 “␣”.

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: for deep levels (in our case, deeper than sections), the numbering before “,” will be hided.

Note that the breaking mark must immediately follows an abbr.

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 “␣”.

2
C  Process

The process of the macro \texttt{\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 “\texttt{Alph,\arabic.\roman}”

Step 3. Split this token list into a sequence by macros. In our example, it results “\texttt{Alph, “,\arabic} and “,\roman”.

Step 4. Store those codes in individual containers.

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

Step 6. Use the containers to redefine \texttt{\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

\begin{verbatim}
\renewcommand*{\thesection}{\Alph{section}}
\renewcommand*{\thesubsection}{\arabic{subsection}}
\renewcommand*{\thesubsubsection}{\thesubsection.\roman{subsubsection}}
\end{verbatim}

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 \texttt{\ProvidesExplPackage} rather than \texttt{\ProvidesPackage}.

\begin{verbatim}
(*package) (\@=syu)
\NeedsTeXFormat{LaTeX2e}
\RequirePackage{expl3}
\ProvidesExplPackage{secnum}{2021/08/28}{}
{ An intuitive way to format section numbering }
\RequirePackage{xparse,l3keys2e}
\l__syu_secnum_tl
\l__syu_secnum_seq
\int_new:N \l__syu_secnum_depth
\g__syu_chapter_tl
\g__syu_section_tl
\g__syu_subsection_tl
\g__syu_subsubsection_tl
\g__syu_paragraph_tl
\g__syu_subparagraph_tl
\tl_new:N \l__syu_secnum_tl
\tl_new:N \l__syu_secnum_seq
\int_new:N \l__syu_secnum_depth
\g__syu_chapter_tl
\g__syu_section_tl
\g__syu_subsection_tl
\g__syu_subsubsection_tl
\g__syu_paragraph_tl
\g__syu_subparagraph_tl
\end{verbatim}
\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.
\ifcs_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:

\l__syu_secnum_bkm  This variable is used to store the breaking mark.
\tl_new:N \g__syu_secnum_bkmr
\tl_gset:Nx \g__syu_secnum_bkmr {,}

Note that one needs the following variants
\cs_generate_variant:Nn \tl_if_in:NnTF { NV }
\cs_generate_variant:Nn \tl_remove_all:Nn { NV }

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 \langle num format\rangle.
breaking .code:n =
  \{ 
  \tl_gset:Nx \g__syu_secnum_bkmr {#1}
  \},

  \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

4
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 and tocdepth.
\int_set:Nn \l__syu_secnum_depth
\seq_count:N \l__syu_secnum_seq
\setcounter{secnumdepth}
\setcounter{tocdepth}{ \int_eval:n \l__syu_secnum_depth - \g__syu_if_thechapter_int }
\setcounter{tocdepth}{ \int_eval:n \l__syu_secnum_depth - \g__syu_if_thechapter_int }
\int_if_exist:NTF \g__syu_tocdep \setcounter{tocdepth}{ \g__syu_tocdep }
\setcounter{tocdepth}{ \int_eval:n \l__syu_secnum_depth - \g__syu_if_thechapter_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: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
But how to see if an \textit{item} in the token list is a macro?

\texttt{\g__syu_macro_tl} This \texttt{tl var} stores the first five characters of the meaning of any macro, \ie \texttt{macro} (watch out its catcode). The idea is to creat a \texttt{tl var} 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} }
\if_meaning:w \l_tmpa_tl \g__syu_macro_tl
\prg_return_true:
\else:
\prg_return_false:
\fi:
\group_end:
}
\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}
\prg_new_protected:Npn \__syu_secnum_from_seq:N #1
{ 
\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
}\end{verbatim}

6 Read formatting info

\texttt{\__syu_secnum_from_seq:N} Read the formatting info from given \texttt{sequence}.

\begin{verbatim}
\cs_new_protected:Npn \__syu_secnum_from_seq:N #1
{/ 
\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
}\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}
 \{ \_\_syu_chapter_tl \{chapter\} \}
Test if the numbering breaks before section.
\tl_if_in:NVTF \_\_syu_section_tl \_\_syu_secnum_bkmr
 \{ 
 \tl_remove_all:NV \_\_syu_section_tl \_\_syu_secnum_bkmr
 \renewcommand*{\thesection}
 \{ \_\_syu_section_tl \{section\} \}
 \}
\}{
\renewcommand*{\thesection}
 \{ \thechapter
 \_\_syu_section_tl \{section\}
 \}

Otherwise start from \thesection.
\else:
 \renewcommand*{\thesection}
 \{ \_\_syu_section_tl \{section\} \}
\fi:

7.ii Subsections
Test if the subsections are needed to be numbered.
\tl_if_empty:NTF \_\_syu_subsection_tl
 \{}

Test if the numbering breaks before subsection.
\tl_if_in:NVTF \_\_syu_subsection_tl \_\_syu_secnum_bkmr
 \{ 
 \tl_remove_all:NV \_\_syu_subsection_tl \_\_syu_secnum_bkmr

\}
7.iii Subsubsections

Test if the subsubsections are needed to be numbered.
\begin{verbatim}
\tl_if_empty:NTF \g__syu_subsubsection_tl {}
\end{verbatim}

Test if the numbering breaks before subsubsection.
\begin{verbatim}
\tl_if_in:NVTF \g__syu_subsubsection_tl \g__syu_secnum_bkmr
{ \g__syu_subsubsection_tl \g__syu_secnum_bkmr
\renewcommand*{\thesubsubsection}
{ \g__syu_subsubsection_tl {subsubsection} }
}
{ \renewcommand*{\thesubsubsection}
{ \thesubsection \g__syu_subsubsection_tl {subsubsection} }
}
\end{verbatim}

7.iv Paragraphs

Test if the paragraphs are needed to be numbered.
\begin{verbatim}
\tl_if_empty:NTF \g__syu_paragraph_tl {}
\end{verbatim}

Test if the numbering breaks before paragraph.
\begin{verbatim}
\tl_if_in:NVTF \g__syu_paragraph_tl \g__syu_secnum_bkmr
{ \g__syu_paragraph_tl \g__syu_secnum_bkmr
\renewcommand*{\theparagraph}
{ \g__syu_paragraph_tl {paragraph} }
}
{ \renewcommand*{\theparagraph}
{ \thesubsubsection \g__syu_paragraph_tl {paragraph} }
}
\end{verbatim}
7.v Subparagraphs

Test if the subparagraphs are needed to be numbered.

\tl_if_empty:NTF \g__syu_subparagraph_tl
{}
{
Test if the numbering breaks before paragraph.

\tl_if_in:NVTF \g__syu_subparagraph_tl \g__syu_secnum_bkmr
{}
{%\tl_remove_all:NV \g__syu_subparagraph_tl \g__syu_secnum_bkmr
\renewcommand*{\thesubparagraph}
{ \g__syu_subparagraph_tl {subparagraph} }
%
{ \renewcommand*{\thesubparagraph}
{ \theparagraph
\g__syu_subparagraph_tl {subparagraph}
}
}
}

{/package}