

The `abbrev` package*

Thai Son Hoang
ECS, University of Southampton
<T dot S dot Hoang at ecs dot soton dot ac dot uk>

May 16, 2017

Abstract

This package provides macros for typesetting abbreviations. It was developed initially at the Swiss Federal Institute of Technology Zurich (ETH-Zurich) and subsequently at the University of Southampton.

Contents

1	Introduction	1
2	Usage	1
2.1	Creating Abbreviations	1
2.2	Resetting Abbreviations	2
2.3	Creating a List of Abbreviations	2
3	Implementation	2

1 Introduction

This package was developed in order to ease the typesetting of abbreviations in \LaTeX . Each abbreviation is associated with a set of macros for typesetting. Abbreviations are collected and can be displayed in list of abbreviations by making use of package `nomenc1`.

2 Usage

Just like any other package, you need to request this package with a `\usepackage` command in the preamble.

So in the simpler case, one just types

```
\usepackage{abbrev}
```

to load the package

The rest of this section is to give descriptions of the main (meta-)macro for creating abbreviation macros and their expected behaviour.

*This document corresponds to `abbrev` v2.0, dated 2017/05/16.

2.1 Creating Abbreviations

`\newabbrev` Macro `\newabbrev` is a meta-macro that creating several abbreviation macros. The signature of `\newabbrev` is as follows.

```
\newabbrev[csname]{abbreviation}{expansion string}  
             {plural-csname}{plural-abbreviation}{plural-expansion-string}
```

The first optional argument `csname` is the *control sequence* name that will be use to create the abbreviation macro, i.e., the marco will be `\csname`. If the first optional argument is omitted, the second argument representing the `abbreviation` will be used as the control sequence. The third argument is the *full expansion string* of for the abbreviation. A plural macro for the abbreviation is also created. By default, the plural control sequence name, the plural abbreviation and the plural expansion string correspond to singular version with an additional `s`. In the case where it should be defined differently, it can be declared as the additional argument `plural-csname`, `plural-abbreviation`, `plural-expansion-string`.

An invocation of the above `\newabbrev` with control sequence `csname` will create the following macros:

- `\csname`: Singular version of the abbreviation macro.
- `\csnames`: Plural version of the abbreviation macro (with an additional `s`).

First time usage of `\csname` will be replaced by “full string (abbreviation)”. Subsequent uses of `\csname` will be replaced by “abbreviation”. Macro `\csnames` is the plural version macro.

2.2 Resetting Abbreviations

`\resetabbrev` The above expansion behaviour can be reset by calling `\resetallabbrev`. Afterwards, `\csname` and `\csnames` will be expanded the first time that they are used. Individual abbreviation `\csname` can be reset by `\resetabbrev{csname}` (Here `csname` is the control sequence corresponding to the singular version of the macro.

More information can be found in the accompanying sample document.

2.3 Creating a List of Abbreviations

The list of abbreviations can be created by making use of package `nomenc1`. Typically, one includes the following in the document preamble.

```
\usepackage{nomenc1}  
\renewcommand{\nomname}{List of Abbreviations}  
\makenomenclature
```

The following command from `nomenc1` is used print the list of abbreviations.

```
\printnomenclature
```

Finally, use `makeindex` to compile and generate the list of abbreviations appropriately.

```
makeindex filename.nlo -s nomenc1.ist -o filename.nls  
where filename is the name of the main .tex file.
```

3 Implementation

The implementation is quite straightforward. We first request the `etoolbox` package for implementation purpose, `xspace` for utilising the spacing, and `nomenc1` for creating list of abbreviations.

```
\RequirePackage{etoolbox}
\RequirePackage{xspace}
\RequirePackage{nomenc1}
```

`\newabbrev` The `newabbrev` makes use of the worker macro `newfullabbrev` for creating abbreviations macros.

```
\newcommand{\newabbrev}[3][]{\%
  \newfullabbrev[#1]{#2}{#3}{-}{-}
}
```

`\newfullabbrev` The `newfullabbrev` makes use of the worker macro `abbrev@create` for creating abbreviations macros.

```
\newcommand{\newfullabbrev}[6][]{\%
  \expandafter\ifstrequal\expandafter{#1}{-}{
    \def\abbrev@sgcsname{#2}
  }{
    \def\abbrev@sgcsname{#1}
  }
  \def\abbrev@sgabbrev{#2}
  \def\abbrev@sgexpn{#3}

  \expandafter\ifstrequal\expandafter{#4}{-}{
    \edef\abbrev@plcsname{\abbrev@sgcsname s}
  }{
    \def\abbrev@plcsname{#4}
  }

  \expandafter\ifstrequal\expandafter{#5}{-}{
    \edef\abbrev@plabbrev{\abbrev@sgabbrev s}
  }{
    \def\abbrev@plabbrev{#5}
  }

  \expandafter\ifstrequal\expandafter{#6}{-}{
    \edef\abbrev@plexpn{\abbrev@sgexpn s}
  }{
    \def\abbrev@plexpn{#6}
  }
  \begingroup\edef\z{\endgroup\noexpand\abbrev@create
    {\abbrev@sgcsname}{\abbrev@sgabbrev}{\abbrev@sgexpn}
    {\abbrev@plcsname}{\abbrev@plabbrev}{\abbrev@plexpn}}\z
}
```

`\abbrev@create` The main worker meta-macro for creating the abbreviation macros is implemented as follows. It create a private toggle variable to be used to distinguish the first time usage of the abbreviation. The abbreviation is add to `abbrev@list` (for resetting

abbreviation macros later). It then call `abbrev@init` to create the initialisation macro, then call the initialisation macro afterward.

```
\newcommand{\abbrev@create}[6]{%
  \newtoggle{#1@toggle}
  \listadd{\abbrev@list}{#1}
  \abbrev@init{#1}{#2}{#3}{#4}{#5}{#6}
  \csuse{#1@init}
}
```

`\abbrev@init` This macro create the initialisation meta-macro for creating the abbreviation macros.

```
\newcommand{\abbrev@init}[6]{%
  \expandafter\def\csname#1@init\endcsname{%
    \togglefalse{#1@toggle}
    \expandafter\def\csname#1\endcsname{%
      \iftoggle{#1@toggle}{%
        #2%
      }{%
        \nomenclature{#2}{#3}%
        \toggletrue{#1@toggle}%
        #3 (#2)%
      }%
    \expandafter\def\csname#1\endcsname{#2\space}%
    \space%
  }

  \expandafter\def\csname#4\endcsname{%
    \iftoggle{#1@toggle}{%
      #5%
    }{%
      \nomenclature{#2}{#3}%
      \toggletrue{#1@toggle}%
      #6 (#5)%
    }%
    \expandafter\def\csname#4\endcsname{#5\space}%
    \space%
  }%
}
```

`\resetallabbrevs` Reset the all abbreviation macros to full expansion mode by going through `abbrev@list` and call the initialisation macro for each abbreviation.

```
\newcommand{\resetallabbrevs}{
  \renewcommand*{\do}[1]{
    \csuse{##1@init}
  }
  \dolistloop{\abbrev@list}
}
```

`\resetabbrev` Reset the abbreviation macro to full expansion mode by calling the initialisation macro for each input abbreviation.

```
\newcommand{\resetabbrev}[1]{
```

```

\csuse{#1@init}
}

```

Change History

v1.0		<code>\newfullabbrev</code> : Macro created . . . 3
	<code>\newabbrev</code> : Macro created 3	<code>\resetabbrev</code> : Added macro to
	General: Initial version 1	reset individual abbreviation . . . 4
v2.0		<code>\resetallabbrevs</code> : Renamed
	<code>\abbrev@create</code> : Allow explicit	macro to resetallabbrevs 4
	declaration of plural version . . . 3	General: Change the APIs to
	<code>\newabbrev</code> : Allow explicit	address plural version of
	declaration of plural version . . . 3	abbreviations 1

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the definition; numbers in *roman* refer to the pages where the entry is used.

A	<code>\abbrev@sgexpn</code> 3	<code>\newfullabbrev</code> . . . 3, <u>3</u>
<code>\abbrev@create</code> . . . 3, <u>3</u>		<code>\noexpand</code> 3
<code>\abbrev@init</code> 3, <u>4</u>	B	
<code>\abbrev@list</code> 3, 4	<code>\begingroup</code> 3	
<code>\abbrev@plabbrev</code> . . . 3	E	R
<code>\abbrev@plcsname</code> . . . 3	<code>\edef</code> 3	<code>\resetabbrev</code> 2, <u>4</u>
<code>\abbrev@plexpn</code> 3	<code>\endgroup</code> 3	<code>\resetallabbrevs</code> . . . <u>4</u>
<code>\abbrev@sgabbrev</code> . . . 3	N	Z
<code>\abbrev@sgcsname</code> . . . 3	<code>\newabbrev</code> 1, <u>3</u>	<code>\z</code> 3