The romanbar package

H.-Martin Münch
<Martin.Muench at Uni-Bonn.de>

2025-01-28 v1.0g

Abstract

This LaTeX package allows to write Roman numbers (or any other text) with horizontal bars. Additionally, commands for converting Arabic numbers into Roman ones are provided and an `\ifnumeric` test function.

Contents

1 Introduction

This LaTeX package allows to write Roman numbers (or any other text) with horizontal bars. Additionally, commands for converting Arabic numbers into Roman ones are provided and an `\ifnumeric` test function.

2 Usage

Just load the package placing

```
\usepackage{romanbar}
```

in the preamble of your LaTeX 2_ε source file.
`\Romanbar{...}` then produces a Roman number with bars (please see the example file). `\romannum{...}` turns an Arabic number into a lowercase Roman one, and `\Romannum{...}` turns an Arabic number into an uppercase Roman one. `\ifnumeric{test}{true}{false}` tests for `test` being numeric.

Options

`options` The romanbar package takes no options.

3 Alternatives

- the original code written by Prof. Enrico Gregorio (https://profs.sci.univr.it/~gregorio/), https://tex.stackexchange.com/a/24084

```
\def\barroman#1{\sbox0{#1}\dimen0=\dimexpr\wd0+1pt\relax
  \makebox[\dimen0]{\rlap{\vrule width\dimen0 height 0.06ex depth 0.06ex}%
    \rlap{\vrule width\dimen0 height\dimexpr\ht0+0.03ex\relax
            depth\dimexpr-\ht0+0.09ex\relax}%
    \kern.5pt#1\kern.5pt}}

\barroman{I} \barroman{XI}
```

- Some fonts provide single characters for Roman numerals, cf. e. g. https://tex.stackexchange.com/q/38695. With TeX-engines like LuaLaTeX and XeLaTeX unicode can be used directly if an according font is used (e. g. `\usepackage{fontspec}\usepackage{libertine}`).

(You programmed or found another alternative, which is available at https://www.ctan.org/?
OK, send an e-mail to me with the name, location at https://www.ctan.org/, and a short notice, and I will probably include it in the list above.)

4 Example

```
⟨*example⟩
\NeedsTeXFormat{LaTeX2e}[2024-11-01]
\documentclass[british]{article}[2024/06/29]% v1.4n
\usepackage[extension=pdf,%
 plainpages=false,%
 pdfpagelabels=true,%
 hyperindex=false,%
 pdflang={en},%
 pdftitle={romanbar package example},%
 pdfauthor={H.-Martin Muench},%
 pdfsubject={Example for the romanbar package},%
 pdfkeywords={LaTeX, romanbar, roman, Roman, bars},%
 pdfview=Fit,pdfstartview=Fit,%
 pdfpagelayout=SinglePage%
]{hyperref}[2024-11-05]% v7.01l
\usepackage{romanbar}[2025-01-28]% v1.0g
\listfiles
\begin{document}
\pagenumbering{arabic}
\section*{Example for romanbar}

This example demonstrates the use of package\newline
\textsf{romanbar}, v1.0g as of 2025-01-28.\newline
There are no options to be used.\newline

\noindent For more details please see the documentation!

\noindent This package provides the command \verb|\Romanbar|
to print bars below and over the following:

\begin{description}
\item[-] Roman numbers: \verb|\Romanbar{MMXXV}| prints \Romanbar{MMXXV}

\item[-] Arabic numbers turned into upper-case Roman numbers:\newline
          \verb|\Romanbar{2025}| prints \Romanbar{2025}

\item[-] negative Arabic numbers turned into upper-case Roman numbers
          \newline
          (although historically there were no negative Roman numbers):
          \newline
          \verb|\Romanbar{-25}| prints \Romanbar{-25}

\item[-] zero Arabic number ($0$; although historically
          there was no Roman zero):\newline
          \verb|\Romanbar{0}| prints \Romanbar{0}

\item[-] some arbitrary text:
          \verb|\Romanbar{Caesar}| prints \Romanbar{Caesar}\newline
          (with descenders: \Romanbar{AgjpqyW})

\newcounter{example}
\setcounter{example}{25}

\item[-] some counter's value:
          \verb|\Romanbar{\theexample}| prints \Romanbar{\theexample}
          \newline
          (where the value of \texttt{example} is \theexample)

\item[-] Arabic numbers, without turning them into upper-case Roman
          numbers:\newline
          \verb|\Romanbar{\relax 2025}| prints \Romanbar{\relax  2025}
```

```
\end{description}

Special care was taken for "L" (50), e.\,g.~in 555/DLV: \Romanbar{555}.

\verb|\romannum{...}| converts an Arabic number into a lower-case Roman one,
e.\,g.~\verb|\romannum{2025}| prints \romannum{2025}, and
\verb|\Romannum{...}| converts an Arabic number into an upper-case Roman one,
e.\,g.~\verb|\Romannum{2025}| prints \Romannum{2025}.
\end{document}
```
⟨/example⟩

5 The implementation

We start off by checking that we are loading into LaTeX 2ε and announcing the name and version of this package.

⟨∗package⟩

```
\NeedsTeXFormat{LaTeX2e}[2024-11-01]
\ProvidesPackage{romanbar}[2025-01-28 v1.0g Roman numbers with bars (HMM)]
```

A short description of the romanbar package:

```
%% Allows to write Roman numbers (or any other text) with bars
%% and to convert Arabic numbers into Roman ones.
```

There are no options to be processed.

```

\RequirePackage{trimclip}[2020/08/19]% v1.2

\newcounter{romanbarcount}
\newdimen\romanbardimH
\newdimen\romanbardimD
\newdimen\romanbardimW

```

`\gobbleminus` A command to gobble a possible minus sign is needed and `\gobbleminus` from https://texfaq.org/FAQ-isitanum is used.

```
\def\gobbleminus#1{\ifx-#1\else#1\fi}
%% from https://texfaq.org/FAQ-isitanum

```

`\ifnumeric` We want to test whether the argument passed to `\Romanbar` is numeric and therefore define `\ifnumeric{test}{true}{false}`, where `test` is to be analysed, `true` is the code to be executed, if `test` is numeric, and `false` is the code to be executed, if `test` is not numeric. This is done similar to https://tex.stackexchange.com/a/17119.

```
\newcommand{\ifnumeric}[3]{%
%% similar to https://tex.stackexchange.com/a/17119
\sbox\z@{\c@romanbarcount=0\gobbleminus#1\relax}%
\ifdim\wd0>\z@\relax#3% is not numeric
\else#2% is numeric
\fi%
}

```

`\r@iseL` We define `\r@iseL` to raise any L (50), otherwise a good deal of the horizontal part of the letter would be swallowed by the lower bar under the "number".

```
\def\r@iseL#1{\ifx @#1% then terminate
\else%
\if L#1\raisebox{0.05ex}{\begin{clipbox}{0pt 0pt 0pt 0.05ex}L\end{clipbox}}%
\else #1\fi%
```

```
\expandafter\r@iseL%
\fi%
}
```

\Romanbar We define the \Romanbar command.

```
\DeclareRobustCommand{\Romanbar}[1]{%
\ifnumeric{#1}{% is numeric
```

If its parameter is numeric, the temporary counter romanbarcount is set to the value.

```
\setcounter{romanbarcount}{#1}%
```

Regularly there are no non-positive Roman numbers, therefore we do some trick similar to \XRoman from the https://ctan.org/pkg/pageslts package.

```
%% similar to \XRoman from the https://ctan.org/pkg/pageslts package
\ifnum\theromanbarcount<1%
  \ifnum #1>0\relax%
    \Roman@bar{\Roman{#1}}%
  \else%
    \ifnum #1<0\relax%
      -\Roman@bar{\expandafter\@slowromancap\romannumeral\number-#1@}%
    \else%
      \Roman@bar{0}%
    \fi%
  \fi%
\else\Roman@bar{\@Roman\c@romanbarcount}%
\fi%
}{% is not numeric
\Roman@bar{#1}%
}%
}
```

\Roman@bar Whether it is numeric or not, in the end \Roman@bar is applied to it.

```
\DeclareRobustCommand{\Roman@bar}[1]{% #1 is in Roman, i.e. MMXXIII
%% (or any other text, "Caesar" would work, too);
%% similar to code of  by Prof. Enrico Gregorio (egreg) at
%% https://tex.stackexchange.com/a/24084
\@bsphack%
\edef\romanbartmp{#1}%
%% height:
\sbox0{\textsf{\romanbartmp}}%
%% top line:
%% + top of top line:
\setlength{\romanbardimH}{\dimexpr\ht0+0.05ex\relax}%
%% + bottom of top line:
\setlength{\romanbardimD}{\dimexpr-\ht0+0.05ex\relax}%
%% width:
\sbox0{\textsf{\expandafter\r@iseL\romanbartmp @}}%
\setlength{\romanbardimW}{\dimexpr\wd0+1pt\relax}%
\@esphack%
\makebox[\romanbardimW]{%
\rlap{\vrule width\romanbardimW height\romanbardimH depth\romanbardimD}% top line
\rlap{\vrule width\romanbardimW height 0.06ex       depth 0.03ex}%    bottom line
\kern0.5pt\textsf{\expandafter\r@iseL\romanbartmp @}\kern0.5pt}%
}
```

\romannum While it is not needed for the bars, it is sometimes asked for a command to convert an Arabic number into a Roman one. \roman{...} can only be applied to a counter, i. e. \roman{25} does not work, but

```
\newcounter{examplecounter}
```

```
\setcounter{examplecounter}{25}
\roman{examplecounter}
```

would be necessary. Further non-negative values would not work. `\romannum` accepts any number as argument. (If the argument is no number, an error message is given and the argument printed (executed) without applying anything to it.)

```
\DeclareRobustCommand{\romannum}[1]{%
\ifnumeric{#1}{% is numeric
\setcounter{romanbarcount}{#1}%
%% similar to \xroman from the https://ctan.org/pkg/pageslts package
\ifnum\c@romanbarcount<1\relax%
  \ifnum #1>0\relax%
    \roman{#1}%
  \else%
    \ifnum #1<0\relax%
      -\romannumeral\number-#1@%
    \else%
      0%
    \fi%
  \fi%
\else\@roman\c@romanbarcount%
\fi%
}{% is not numeric
\PackageError{romanbar}{%
Argument of \string\romannum\space is not a number}{%
The command \string\romannum\space converts an Arabic number into a %
lower-case Roman one,\MessageBreak%
but the used argument of \string\romannum\space is not an Arabic number %
but\MessageBreak%
`#1',\MessageBreak%
which will now be printed unchanged.}%
#1%
}%
}

```

`\Romannum` The same for upper-case Roman numbers, `\Romannum` instead of `\Roman`:

```
\DeclareRobustCommand{\Romannum}[1]{%
\ifnumeric{#1}{% is numeric
\setcounter{romanbarcount}{#1}%
%% similar to \XRoman from the http://ctan.org/pkg/pageslts package
\ifnum\c@romanbarcount<1\relax%
  \ifnum #1>0\relax%
    \Roman{#1}%
  \else%
    \ifnum #1<0\relax%
      -\expandafter\@slowromancap\romannumeral\number-#1@%
    \else%
      0%
    \fi%
  \fi%
\else\@Roman\c@romanbarcount%
\fi%
}{% is not numeric
\PackageError{romanbar}{%
Argument of \string\Romannum\space is not a number}{%
The command \string\Romannum\space converts an Arabic number into an %
upper-case Roman one,\MessageBreak%
but the used argument of \string\Romannum\space is not an Arabic number %
but\MessageBreak%
`#1'.}%
#1%
```

```
}%
}

⟨/package⟩
```

6 Installation

6.1 Downloads

Everything is available at https://www.CTAN.org/, but may need additional packages themselves.

`romanbar.dtx` For unpacking the `romanbar.dtx` file and constructing the documentation it is required:

- LaTeX 2ε [2024-11-01] or newer, https://www.ctan.org/starter
- document class ltxdoc, 2024/02/08, v2.1j, https://www.ctan.org/pkg/ltxdoc
- package holtxdoc, 2019/12/09, v0.30, https://www.ctan.org/pkg/holtxdoc
- package hypdoc, 2023-10-26, v1.19, https://www.ctan.org/pkg/hypdoc

`romanbar.sty` The `romanbar.sty` for LaTeX 2ε (i. e. each document using the romanbar package) requires:

- TeXFormat LaTeX 2ε [2024-11-01] or newer, https://www.ctan.org/starter
- package trimclip, 2020/08/19, v1.2, from the adjustbox bundle, https://www.ctan.org/pkg/adjustbox

`romanbar-example.tex` The `romanbar-example.tex` requires the same files as all documents using the romanbar package and additionally:

- class article, 2024/06/29, v1.4n , from classes: https://www.ctan.org/pkg/classes
- package romanbar, 2025-01-28, v1.0g, https://www.ctan.org/pkg/romanbar (Well, it is the example file for this package, and because you are reading the documentation for the romanbar package, it can be assumed that you already have some version of it – is it the current one?)

`Oberdiek` `holtxdoc` All packages of Heiko Oberdiek's bundle 'oberdiek' (especially holtxdoc) are also available in a TDS compliant ZIP archive:
https://mirrors.ctan.org/macros/latex/contrib/oberdiek.zip.
It is probably best to download and use this, because the packages in there are quite probably both recent and compatible among themselves.

`hyperref` hyperref is not included in that bundle and needs to be downloaded separately, https://mirrors.ctan.org/macros/latex/contrib/hyperref.zip.

`Münch` A hyperlinked list of my (other) packages can be found at https://www.ctan.org/author/muench-hm.

6.2 Package, unpacking TDS

Package. This package is available on CTAN:

http://mirrors.ctan.org/macros/latex/contrib/romanbar/romanbar.dtx
The source file.

http://mirrors.ctan.org/macros/latex/contrib/romanbar/romanbar.pdf
The documentation.

http://mirrors.ctan.org/macros/latex/contrib/romanbar/romanbar-example.pdf

The compiled example file, as it should look like.

http://mirrors.ctan.org/macros/latex/contrib/romanbar/README
The README file.

There is also a romanbar.tds.zip available:

https://mirrors.ctan.org/install/macros/latex/contrib/romanbar.tds.zip

Everything in TDS compliant, compiled format.

which additionally contains

romanbar.ins	The installation file.
romanbar.drv	The driver to generate the documentation.
romanbar.sty	The .style file.
romanbar-example.tex	The example file.

For required other packages, please see the preceding subsection.

Unpacking. The .dtx file is a self-extracting docstrip archive. The files are extracted by running the ..dtx through plain TEX:

```
tex romanbar.dtx
```

About generating the documentation see paragraph 6.4 below.

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as texmf tree):

```
romanbar.sty          → tex/latex/romanbar/romanbar.sty
romanbar.pdf          → doc/latex/romanbar/romanbar.pdf
romanbar-example.tex  → doc/latex/romanbar/romanbar-example.tex
romanbar-example.pdf  → doc/latex/romanbar/romanbar-example.pdf
romanbar.dtx          → source/latex/romanbar/romanbar.dtx
```

If you have a docstrip.cfg that configures and enables docstrip's TDS installing feature, then some files can already be in the right place, see the documentation of docstrip.

6.3 Refresh file name databases

If your TEX distribution (TEX Live, MiKTEX, ...) relies on file name databases, you must refresh these. For example, TEX Live users run texhash or mktexlsr.

6.4 Some details for the interested

Unpacking with LATEX. The .dtx chooses its action depending on the format:

plain TEX: Run docstrip and extract the files.

LATEX: Generate the documentation.

If you insist on using LATEX for docstrip (really, docstrip does not need LATEX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{romanbar.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by a configuration file `ltxdoc.cfg`. For instance, put the following line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfLaTeX:

```
pdflatex romanbar.dtx
makeindex -s gind.ist romanbar.idx
pdflatex romanbar.dtx
makeindex -s gind.ist romanbar.idx
pdflatex romanbar.dtx
```

6.5 Compiling the example

The example file, romanbar-example.tex, can be compiled via `(pdf)latex romanbar-example.tex`.

7 Acknowledgements

I would like to thank Prof. Enrico Gregorio (egreg) for his answer at https://tex.stackexchange.com/a/24084 and Heiko Oberdiek for providing a lot of useful packages (from which I also got everything I know about creating a file in `.dtx` format, OK, say it: copying).

8 History

[2011/07/25 v1.0a]

- First code by Prof. Enrico Gregorio (egreg) at https://tex.stackexchange.com/a/24084

[2011/07/26 v1.0b]

- Packed into a `.sty` file.
- Removed the requirement of ε-TeX for the package.

[2011/08/25 v1.0c]

- Renamed romanbar package.
- Minor details.

[2011/09/16 v1.0d]

- Made commands robust.
- Minor details.

[2011/12/16 v1.0e]

- Additionally accepts now Arabic numbers and converts them to Roman ones with bars.
- Also non-positive numbers are now accepted.

[2012/01/01 v1.0f]

- Added `\romannum` and `\Romannum` to convert Arabic to Roman numbers (without bars).
- `\r@iseL`.
- Created a `dtx`, example, and README.
- Uploaded to CTAN.

[2025-01-28 v1.0g]

- Converted to UTF-8.
- Updated to LaTeX format 2024-11-01.
- Documentation updates.
- Using the trimclip package for improved handling of "L".
- Again using ε-TeX for the package. Version 1.0b removed the requirement of ε-TeX for this package. Now it is 2025 and ε-TeX can be assumed standard. Therefore ε-TeX is used for this package. Also own counters and own dimens instead of scratch ones are used now.

When you find a mistake or have a suggestion for an improvement of this package, please send an e-mail to the maintainer, thanks! (Please see BUG REPORTS in the README.)

9 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.