# The package modroman*

Le TEXnicien de surface

April 9, 2010

**Abstract**

**Documentation anglaise pour l'utilisateur final de l'extension modroman. La documentation française est disponible sous le nom de `modroman-fr`.**

---

This is the English documentation of modroman for the final user.

This short package provides macros which enable one to write roman numerals with some modifications.

It provides macros with which one can obtain 'i' for 1 and 'xviij' for 18; 'dcccclxxxxviiij' or 'DCCCCLXXXXVIIII' for 999.

With option UPOURV — opposite of VPOURV default option — one obtains 'xuj' for 15. With option IFINAL — opposite of JFINAL default option — one can obtain 'xiiii' for 14.

Some of the macros are used as TEX `\romannumeral`, others, purely expandable, are devoted to format a counter — as LATEX `\roman` — or a number.

It also provides a macro — `\printntimes{`⟨*number*⟩`}{`⟨*text*⟩`}` — which produces ⟨*number*⟩ times the `{`⟨*text*⟩`}` as e.g. `**********` obtained with `\printntimes{10}{*}`.

## Contents



*This document corresponds to the file modroman v1, dated 2010/04/09.

# 1 Introduction

## 1.1 Motivation

The ways of writing numbers with roman numerals are more diverse than it could be thought when one considers the TEX `\romannumeral` and LATEX `\roman` and `\Roman`. Many other forms were used at a time or another through history. I saw, many years ago, the form viij in a manuscript. The first versions — from 0.1 to 0.4 — of this package provided just that form, with, user willing, the use of a `u` to denote a group of 5 as in 'xuij' for 17.

The interested reader could cast a glance on the page wikipedia devotes to roman numerals to see that the world is not always as simple as one would like it to be.

## 1.2 Technical Remarks

The code of the first versions enable one to number the pages with *modified* roman numerals but one could not then use the reference tools such as `\label` and `\ref` or even hope to see the page numbers correctly written in the table of contents.

As the years go by I don't really become more clever but, for I read not a few documentations of packages, I end with knowing a bit more and I happen to stumble upon the solution to a problem I had just caught a glimpse of.

I don't forget what I owe to T. Lachand-Robert in [1] – numerous ideas, detailed examples, clear explanations – but I now use other sources – `source2e` [2] to be accurate. That is where I have found the *trick* which enables me to provide this new version of `modroman`. The reader who would like to know more should have a look at the definition of the macro `\Roman` and its auxiliary macros.

While rewriting the code I happen to understand that I was able to go a bit farther than I have gone previously without a tremendous extra cost. That's why one will now find more macros and more presentations of the roman numerals and an additional macro.

This version 1 keeps the compatibility with the previous version but the code has been completely rewritten and the package now provides to the user in addition to `\modroman` and `\modromannumeral` about fifteen other macros.

However, the main change introduced by this version 1 is the fact that from now on `\modroman` — and its pals — is purely expandable — see page *ij* for more details. One can then use it to number pages and obtain the correct thing in the table of content – which was not the case until now. Caution, I do *not* say that `\modromannumeral` is purely expandable – it is not.

## 1.3 Purely Expandable Macros

One could, if one understands French, read the thread 'test de développabilité pure ?' on the news group `fr.comp.text.tex` to see that that notion is not as simple as one could think at n-th sight `:-)` however, here, when I will say that a macro is 'purely expandable' I will understand what follows.

Let's assume that the macro `\thing` is such that `\thing{`⟨*12*⟩`}` gives 'xij' — does it ring a bell? — then

1. the macro `\THING` defined by `\edef\TRUC{\truc{12}}` is such that:

(a) `\THING` gives ‘xij’ and

(b) `\meaning\THING` gives ‘`macro:->xij`’

2. moreover, if one defines `\Axij` then the construct `\csname A\truc{12} \endcsname` truly calls the macro `\Axij`.

# 2 Usage

## 2.1 The Macros

Macros the name of which ends with `numeral` are to be used as TEX `\romannumeral`. They must be followed by a number and eat the spaces which are after it, e. g. `\longromannumeral 368␣␣AND` gives ‘ccclxviijAND’.

Macros the name of which begins with `\nb` take a number as argument such as `\nbLongRoman{127}` which gives ‘CXXVII’.

Macros the name of which doesn’t begin with `\nb` but ends with `roman` are used as LATEX `\roman`: their only argument is the name of a counter. With `\newcounter{machin}`, `\setcounter{machin}{124}`, `\shortroman{machin}` one obtains ‘cxxiv’.

In what follows ⟨*nbr*⟩ denotes a number, ⟨*ctr*⟩ denotes the LATEX name of a counter such as `page` or `chapter`.

Here comes a presentation of all the macros available with this package. They are grouped by family where a family is defined with respect to the obtained presentation of roman numerals.

After the macro’s syntax, there will be [**PD**]to mean that the macro is purely expandable — see page *ij* —, [**LaTeX**]to mean that it is used the LATEX way, [**TeX**]to mean that it is used as the TEX `\romannumeral`.

The examples are governed by the default options: JFINAL, VPOURV, COURT, MIN.

### 2.1.1 shortroman Family

`\shortroman` `\shortroman{`⟨*ctr*⟩`}` [**PD**][**LaTeX**]

`\shortromannumeral` `\shortromannumeral` ⟨*nbr*⟩ [**TeX**]

`\nbshortroman` `\nbshortroman{`⟨*nbr*⟩`}` [**PD**][**LaTeX**]

---

**Examples**
`\nbshortroman{1}` ⟶ i
`\nbshortroman{11}` ⟶ xj
`\nbshortroman{444}` ⟶ cdxliv
`\nbshortroman{888}` ⟶ dccclxxxviij
`\nbshortroman{1999}` ⟶ mcmxcix

---

### 2.1.2 longroman Family

`\longroman` `\longroman{`⟨*ctr*⟩`}` **[PD][LaTeX]**

`\longromannumeral` `\longromannumeral` ⟨*nbr*⟩ **[TeX]**

`\nblongroman` `\nblongroman{`⟨*nbr*⟩`}` **[PD][LaTeX]**

---

**Examples**
`\nblongroman{1}` ⟶ i
`\nblongroman{11}` ⟶ xj
`\nblongroman{444}` ⟶ ccccxxxxiiij
`\nblongroman{888}` ⟶ dccclxxxviij
`\nblongroman{1999}` ⟶ mdcccclxxxxviiij

---

### 2.1.3 LongRoman Family

`\LongRoman` `\LongRoman{`⟨*ctr*⟩`}` **[PD][LaTeX]**

`\LongRomannumeral` `\LongRomannumeral` ⟨*nbr*⟩ **[TeX]**

`\nbLongRoman` `\nbLongRoman{`⟨*nbr*⟩`}` **[PD][LaTeX]**

---

**Examples**
`\nbLongRoman{1}` ⟶ I
`\nbLongRoman{11}` ⟶ XI
`\nbLongRoman{444}` ⟶ CCCCXXXXIIII
`\nbLongRoman{888}` ⟶ DCCCLXXXVIII
`\nbLongRoman{1999}` ⟶ MDCCCCLXXXXVIIII

---

### 2.1.4 roman Family

TeX provides `\romannumeral` and LaTeX `\roman`. I complete the family with `\nbroman`.

`\nbroman` `\nbroman{`⟨*nbr*⟩`}` **[PD][LaTeX]**

---

**Examples**
`\nbroman{1}` ⟶ i
`\nbroman{11}` ⟶ xi
`\nbroman{444}` ⟶ cdxliv
`\nbroman{888}` ⟶ dccclxxxviii
`\nbroman{1999}` ⟶ mcmxcix

---

### 2.1.5 Roman Family

LaTeX provides `\Roman`. I complete the family with `\nbRoman` and `\Romannumeral`.

`\Romannumeral` `\Romannumeral` ⟨*nbr*⟩ [**TEX**]

`\nbRoman` `\nbRoman{`⟨*nbr*⟩`}` [**PD**][**LATEX**]

---

**Examples**
`\nbRoman{1}` ⟶ I
`\nbRoman{11}` ⟶ XI
`\nbRoman{444}` ⟶ CDXLIV
`\nbRoman{888}` ⟶ DCCCLXXXVIII
`\nbRoman{1999}` ⟶ MCMXCIX

---

### 2.1.6 modroman Family

The output of the macros `\modroman`, `\modromannumeral`, and `\nbmodroman` is determined by the chosen options. By default:

`\modroman` `\modroman{`⟨*ctr*⟩`}` = `\shortroman{`⟨*ctr*⟩`}` [**PD**][**LATEX**]

`\modromannumeral` `\modromannumeral`⟨*nbr*⟩ = `\shortromannumeral`⟨*nbr*⟩ [**TEX**]

`\nbmodroman` `\nbmodroman{`⟨*nbr*⟩`}` = `\nbshortroman{`⟨*nbr*⟩`}` [**PD**][**LATEX**]

---

**Examples**
`\nbmodroman{1}` ⟶ i
`\nbmodroman{11}` ⟶ xj
`\nbmodroman{444}` ⟶ cdxliv
`\nbmodroman{888}` ⟶ dccclxxxviij
`\nbmodroman{1999}` ⟶ mcmxcix

---

### 2.1.7 Other macros

One can redefine the behaviour of families `\shortroman` and `\longroman` with the macro `\RedefineMRmdclxvij`.

`\RedefineMRmdclxvij` `\RedefineMRmdclxvij[`⟨*ISOL*⟩`]{`⟨*M*⟩`}{`⟨*D*⟩`}{`⟨*C*⟩`}{`⟨*L*⟩`}{`⟨*X*⟩`}{`⟨*V*⟩`}{`⟨*I*⟩`}{`⟨*J*⟩`}`

The above arguments determine the look of the roman numerals produced after — one should take care of the side effects and one would have to limit the scope of the redefinition ot a group if necessary — by the families `\shortroman`, `\longroman`, and, if it is linked to one of the two preceding, `\modroman`.

The optional argument ⟨*ISOL*⟩ gives the look of the isolated `i` i. e. the number 1. If one doesn't give the argument the look of the isolated `i` is the look of the non-final `i` determined by ⟨*I*⟩

The look of the final `i` is given by the argument ⟨*J*⟩. All other arguments give the look of the corresponding (lowercase) digit, e. g. ⟨*M*⟩ gives the look of `m`.

\printntimes `\printntimes{⟨nbr⟩}{⟨text⟩}`

Here are two examples using the macro.
`$\ast$ \texttt{\printntimes{15}{*-*}} $\ast$`
gives
∗ *-**-**-**-**-**-**-**-**-**-**-**-**-**-* ∗

```
\newcommand\truc{\par\centering ***\par}
\newcommand\saut{\par\noindent\hrulefill\par}
\saut\texttt{\printntimes{5}{\truc}}\saut
```
gives

---

***
***
***
***
***

---

## 2.2 The Options

There exist – since the version 0.2 – options VPOURV and UPOURV. The default option is VPOURV with which `\modromannumeral5` is written as 'v'. With the option UPOURV the same `\modromannumeral5` is written as 'u'. It was a special requirement from *one* person posting on `fr.comp.text.tex`. *The (almost) French 'vpourv' stands for 'v for v'.*

For sake of symetry, I add, with this version 1, two antithetical options JFINAL – final j – the default, and IFINAL – final i – by which one can choose if the last i of the number will be written as a `j` or not.

I add also two pairs of antithetic options. First MIN — i. e. *minuscule* lowercase, default option — and MAJ — i. e. *majuscule* uppercase — then COURT — short, default option — and LONG.

Last, with this version 1, I add the option SANSMOD – without modification – which makes the macros of the `\modroman` family aliases of those of `\roman` family.

The last five options determine the behaviour of the macros of the family `\modroman`.

UPOURV With that option, the roman numeral 'v' is turn into an 'u' and one obtains, e.g., 'xuij' for 17.

VPOURV That option, enforced by default, is the opposite of the previous one. With it, one obtains xvij for 17.

The next three options appear with the version 1 of the package.

JFINAL With that default option, if the processed number is greater than 1 and if the last roman *digit* is an i then it is turned into a j. See the examples above.

IFINAL That option is the opposite of the previous one. When enforced, one obtains xvii for 17.

SANSMOD With that option `\modroman`, `\modromannumeral` and `\nbmodroman` are just aliases – obtained with `\let` – of `\roman`, `\romannumeral`, and `\nbroman` respectively.

If one choses options VPOURV, , COURT, MIN, and IFINAL together, one enforces the option SANSMOD.

The following table shows which family is linked to the `\modroman` family according to the chosen options when SANSMOD is not enforced.

| | COURT | LONG |
|---|---|---|
| MIN | `\shortroman` | `\longroman` |
| MAJ | `\Roman` | `\LongRoman` |

# References


[1] T. Lachand-Robert. *La maîtrise de TEX et LATEX*. Masson, Paris, Milan, Barcelone, 1995. ISBN : 2-225-84832-7.

[2] Johannes Braams, David Carlisle, Alan Jeffrey, Leslie Lamport, Frank Mittelbach, Chris Rowley, Rainer Schöpf. *The LATEX 2ε sources*. 2009/09/24.


---

In the preambule of this document, there is
`\renewcommand\thepage{\textit{\modroman{page}}}`
hence the page numbering.

---

Here ends the documentation of modroman.