# The 1, 2, many package*

Ulrich M. Schwarz†

April 9, 2005

**Abstract**

In the discrete branches of mathematics and the computer sciences, it will only take some seconds until you're faced with a set like $\{1, \ldots, m\}$. Only some people write $1..m$, or $\{j : 1 \leq j \leq m\}$, and that journal you're submitting to might want something else entirely. 1, 2, many provides an interface that makes changing from one to another a one-line change.

## Contents



## 1 Command overview

To use 1, 2, many in your LaTeX document, place 12many.sty into your local texmf tree and load it by `\usepackage{12many}`. Three in-document commands are provided: `\nto{`⟨*from*⟩`}{`⟨*to*⟩`}` takes two arguments and typesets the range from the first to the second, ends inclusive: `\nto{3}{4}` might yield something like $\{3, \ldots, 4\}$.

`\nto`

`\ito` `\ito` is an alias to `\nto{1}`, i.e. `\ito{3}` is `\nto{1}{3}` and yields $\{1, \ldots, 3\}$.

`\oto` `\oto` is an alias to `\nto{0}`, i.e. `\ito{3}` is `\nto{0}{3}` and yields $\{0, \ldots, 3\}$.

`\setOTMstyle` To select the style, use `\setOTMstyle[`⟨*params*⟩`]{`⟨*style*⟩`}`, where style is a style name, and params are style-specific options in a key=val fashion. For a list

*This documents 1, 2, many v0.3 of 2005/05/09

†ulmi@users.sarovar.org

of pre-defined styles and their options, see section 2. It's as simple as that! If you don't need to change the default parameters of a style, you can also pass it as a package option: `\usepackage[laue]{12many}`

# 2 Pre-defined styles

The following styles are predefined by 1, 2, many:

## 2.1 Style `set`

The "proper" way of specifying a range: $\{i \in \mathrm{N}_0 | 2 \leq i \leq 233\}$. Supported parameters:

**var** The variable name to use. Default: the popular scratch integer $i$.

**naturals** The way you write the set of natural numbers, 0 included. Defaults to the (rather ugly) `\mathrm{N}_0` $\mathrm{N}_0$, but depending on your style, you'll use blackboard, boldface or even fraktur here.

**where** This goes between the naturals set symbol and the lower bound. I've seen both colons and bars used here, and you might want to do fancy extra spacing. The default is `|`

## 2.2 Style `dots`

The somewhat less formal enumeration style: $\{2, \ldots, 233\}$. Supported parameters:

**dots** What goes between the bounds. The default, `,\ldots,`, is the most formal one but takes up a lot of horizontal space, so you might use something like `,\ldotp\ldotp` here.

## 2.3 Style `nude`

An even less formal enumeration style, popular with economics and CS people, but certainly not mathematicians: $2..233$. Supported parameters:

**dots** Works just like in the case of dots-style, but for nude, it defaults to `\ldotp\ldotp`.

## 2.4 Style `laue`

A style championed by a local maths professor: $\underline{233}$. (Note that this is the one-to variant. The other variants are derived from this: $(\{0\} \cup \underline{233})$ and $(\underline{233} \setminus \underline{1})$. Also note that the outer parens are added by 1, 2, many to make sure the semantics do not change.) Supported parameters:

**setminus** Used for the variants that do not start at 0 or 1. Defaults to `\setminus`.

**setplus** Used to add in the {0}. Defaults to `\cup`.

**ybelow** How far below the baseline the rule is. I don't think you'd need to change this—the default of 0.3 ex looks good to me in Computer Modern, Palatino and Utopia/Fourier.

**strokewidth** The width of the rule, with a default of 0.08 ex.

**innersidegap** The rule protudes beyond the number above it, by default by 0.05 em.

**outersidegap** There is an additional space after the hook, by default 0.05 em.

# 3 Creating new styles

If none of the styles above float your boat, you can still define your own. To this end, 1, 2, many provides three commands:

`\newOTMstyle` `\newOTMstyle[`⟨*params*⟩`]{`⟨*name*⟩`}{`⟨*definition*⟩`}` takes three arguments: optional parameters, the style name and its definition. The definition has access to the bounds that are passed in the parameters `#1` (lower) and `#2` (upper). For example, the dots style is defined like this:

```
\newOTMstyle[dots={,\ldots,}]{dots}{%
  \{#1\getOTMparameter{dots}#2\}%
}
```

*Please be careful when declaring parameters with the optional arguments: spaces are not stripped. If you declare `foo=bar,␣baz=bam`, `\getOTMparameter{baz}` won't find anything.*

`\renewOTMstyle` This macro is just the same as `\newOTMstyle`, only a newcommand in a crucial place is changed to a renewcommand, i.e. this can be used to change an existing style.

`\newOTMparameter` This macro declares parameters and their defaults to a style. For example, we have already seen that the dots style accesses its dots with `\getOTMparameter{dots}`. This parameter was declared with

```
\newOTMparameter{dots}{dots}{,\ldots,}
```

i.e., "(style) `dots` has a parameter `dots` that defaults to ,...,".

## 3.1 Configuration files

1, 2, many will also look for your new style declarations in a file called `12many.cfg`. This file is read in before the package parameter is evaluated, so it's legal to name a style that will only be defined in `12many.cfg`.

# Change History



# 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; numbers in roman refer to the code lines where the entry is used.