

# crossrefenum

Smart typesetting of enumerated cross-references for various T<sub>E</sub>X formats

Bastien Dumont

**V. 1.2**

Last update of this manual: 2025/08/07

Contact: bastien [dot] dumont [at] posteo [dot] net

Copyright (C) 2022-2025 Bastien Dumont. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license should have been distributed with this file. If not, see <https://www.gnu.org/licenses/fdl-1.3.html>.

# Contents

<b>1 Loading</b>	<b>4</b>
<b>2 Summary</b>	<b>5</b>
<b>3 Basic invocation</b>	<b>8</b>
<b>4 Customization</b>	<b>10</b>
4.1 General principles	10
4.2 Prefixes, delimiters and separators	11
4.3 Collapsible and non-collapsible types	13
4.4 Double types	14
<b>5 How to extend crossrefenum with other types and formats</b>	<b>19</b>
<b>6 Compatibility issues</b>	<b>22</b>

**7 Changes ..... 23**

7.1 2022/11/11: Version 1.0 ..... 23

7.2 2023/02/20: Version 1.0.1 ..... 23

7.3 2023/02/20: Version 1.0.2 ..... 23

7.4 2024/04/13: Version 1.1 ..... 23

7.5 2025/08/07: Version 1.2 ..... 23

*crossrefenum* lets T<sub>E</sub>X manage the formatting of bunches of cross-references for you. It features:

- Automatic collapsing of references;
- Support for references by various criteria:
  - page and note number;
  - in ConT<sub>E</sub>Xt: line number;
  - in L<sup>A</sup>T<sub>E</sub>X, when used in conjunction with *reledmac*: edpage and edline.
- Handling of references combining two criteria (e.g. by page and note number);
- Extension mechanisms to add support to other types of references without modifying the internal macros.

Note that sorting is not supported. I assume that users know in what order the labels they refer to appear in their document.

It is written in Plain T<sub>E</sub>X as much as possible in order to make it compatible with a wide array of formats. For the moment, it works out of the box with ConT<sub>E</sub>Xt and L<sup>A</sup>T<sub>E</sub>X. Its only (optional) dependency is [expkv](#), but it may require other packages to work properly in some formats (e.g. [zref](#) in L<sup>A</sup>T<sub>E</sub>X).

The file `main-test.pdf` provides a showcase of the abilities of *crossrefenum*.

# 1 Loading

To load *crossrefenum*, provided that `crossrefenum.tex`, `crossrefenum.sty` and `t-crossrefenum.tex` are installed in a directory where T<sub>E</sub>X will find them (presumably under the `tex/generic/crossrefenum` directory of one of your *texmf* trees), you can do:

- `\input{crossrefenum} (generic);`
- `\usepackage{crossrefenum}` (L<sup>A</sup>T<sub>E</sub>X; must be called after *nameref* if you use *hyperref*);
- `\usemodule[crossrefenum]` (ConT<sub>E</sub>Xt).

# 2 Summary

- `\crossrefenum`: the main macro
- `\crfnmsetup`: configuration macro (default of per type)

Configuration key (with <code>\crfnmsetup</code> )	Configuration macro (with <code>\def</code> )	Meaning	Example	Page
<i>&lt;type&gt; is a single type</i>			page	
sg	<code>\crfnm&lt;type&gt;</code>	Singular prefix	{p. }	11
pl	<code>\crfnm&lt;type&gt;s</code>	Plural prefix	{pp. }	11
delimiter	<code>\crfnm&lt;type&gt;EnumDelim</code>	Delimiter between references	{, }	12
before last reference	<code>\crfnm&lt;type&gt;BeforeLast InEnum</code>	Delimiter before the last reference	{ and }	12
range separator	<code>\crfnm&lt;type&gt;RangeSep</code>	Separator between the two values in a range	–	13

collapsible?	<code>\crfnm&lt;type&gt;Collapsible</code>	Should consecutive numbers (e.g. 2, 3, 4) be merged into a range?	yes or no	<a href="#">13</a>
<i>&lt;type&gt; is a double type</i>			pagenote	
subtypes separator	<code>\crfnm&lt;type&gt;SubtypesSep</code>	Separator between the two types of references in a double reference (e.g. between the page and note numbers in a reference to a note including the page number)	{, }	<a href="#">14</a>
print prefix of first subtype	<code>\crfnm&lt;type&gt;PrintFirst Prefix</code>	Should the numbers of the first subtype be prefixed always or only for the first reference in an enumeration?	always or once	<a href="#">14</a>
group subtypes?	<code>\crfnm&lt;type&gt;Group Subtypes</code>	Should all the values for each subtype be printed separately?	yes or no	<a href="#">17</a>



order	<code>\crfnm&lt;type&gt;Order</code>	Whether the subtypes are printed in the same order as in the name of the double type	normal or inverted	18
<i>&lt;type&gt; is a single type; the following options apply when it is used as the second subtype of a double type</i>			page	
formatting when second subtype	<code>\crfnm&lt;type&gt;FormatInSecond</code>	{ } or a macro that takes the prefixes and numbers as its argument	<code>\textbf</code>	15
print prefix when second subtype?	<code>\crfnm&lt;type&gt;PrintPrefixInSecond</code>	Should the prefix be printed?	yes or no	15
delimiter when second subtype	<code>\crfnm&lt;type&gt;EnumDelimInSecond</code>	Delimiter between references	{, }	16
before last reference when second subtype	<code>\crfnm&lt;type&gt;BeforeLastInSecond</code>	Delimiter before the last reference	{ and }	16
continuous numbering ?	<code>\crfnm&lt;type&gt;NumberingContinuousAcrossDocument</code>	Is the numbering for this type continuous (i.e. not reset at every page/chapter/etc.)?	yes or no	17

### 3 Basic invocation

The macro `\crossrefenum` has the following syntax:

```
\crossrefenum[<type>][<print prefix?>]{<enumeration>}
```

- *type* is the type of the references. Built-in possible values are:
  - For L<sup>A</sup>T<sub>E</sub>X and ConT<sub>E</sub>Xt: page, note, pagenote;
  - For ConT<sub>E</sub>Xt only: line, pageline;
  - For L<sup>A</sup>T<sub>E</sub>X with *reledmac*: edpage, edline, edpageline.
- *print prefix?* indicates whether the prefix (like “p. ”) should be printed or not: true if set to `withprefix` or `yes`;
- *enumeration* is a group containing one or more *single* labels (e.g. `mylabel`) or *ranges* (e.g. `lbl-begin` to `lbl-end`). Ranges cannot be used with `note` and `pagenote` types. Two syntaxes are supported: lists of groups and comma-delimited lists (see below).

*type* and *print prefix?* are optional. *type* defaults to `page` and *print prefix?* to `withprefix`.

Here are some valid invocations with the comma-delimited syntax:

- `\crossrefenum[note][withprefix]{lblone, lbltwo, lblthree}`
- `\crossrefenum[edline][noprefix]{lblone, lbltwo, lblthree}`
- `\crossrefenum[page]{lblone to lbltwo, lblthree}`
- `\crossrefenum[noprefix]{lblone, lbltwo, lblthree}` (*type* defaults to `page`)

- `\crossrefenum[note]{\lblone, \lbltwo, \lblthree}` (*print prefix?* defaults to *withprefix*)
- `\crossrefenum{\lblone, \lbltwo, \lblthree}` (*type* defaults to *page* and *print prefix?* defaults to *withprefix*)
- `\crossrefenum{only-one}`

The same invocations with the group-based syntax:

- `\crossrefenum[note][withprefix]{\{\lblone\}\{\lbltwo\}\{\lblthree\}}`
- `\crossrefenum[edline][noprefix]{\{\lblone\}\{\lbltwo\}\{\lblthree\}}`
- `\crossrefenum[page]{\{\lblone to \lbltwo\}\{\lblthree\}}`
- `\crossrefenum[noprefix]{\{\lblone\}\{\lbltwo\}\{\lblthree\}}`
- `\crossrefenum[note]{\{\lblone\}\{\lbltwo\}\{\lblthree\}}`
- `\crossrefenum{\{\lblone\}\{\lbltwo\}\{\lblthree\}}`
- `\crossrefenum{\{only-one\}}` (even if the enumeration is limited to one item, it can be inside its own group)

## 4 Customization

There are two configuration interfaces: one based on key-value lists, the other on (re)defining macros. These two interfaces can be used concurrently. If you are not interested in using the key-value interface or prefer not to depend on the `expkv` package, you can deactivate it by defining `\crfNmNoKV` to anything other than `\relax` before loading `crossrefenum`.

After having explained the general principles, we will present the options related to single types before we turn to the options specific to double types and their subtypes. For each configuration option, I will show first how to use the key-value interface using `\crfNmsetup`, and second how to do the same thing with the low-level macros.

### 4.1 General principles

In this manual, “default” means “not type-specific”. In this respect, “default” settings may be set by the user.

The key-value interface has the following syntax:

```
\crfNmsetup[<“default” or type>]{ <key1> = <value1>, <...> }
```

*type* is a single or a double type (e.g. `page` or `pagenote`). The spaces around the equal signs and the commas are optional and ignored; if a value contains leading or trailing spaces, it must be put inside a group (e.g. `delimiter = {, }`).

Under the hood, `\crfnmsetup` (re)defines the low-level configuration macros, which you can also manipulate directly.

When successive calls to `\crfnmsetup` contradict one another, the last one prevails, except that type-specific settings always have precedence over the default ones.

In the following subsections, I will generally present the low-level macros corresponding to the default settings, which have `Default` in their name. If you want to redefine a macro for a specific type, simply replace `Default` with the (capitalized) name of the type (e.g. `\crfnmPageEnumDelim` instead of `\crfnmDefaultEnumDelim`).

Setting a value in the key-value interface or a macro to `\relax` will cause `\crossrefenum` to fall back to the default value. To set a configuration option to an absence of operation, use an empty group (e.g. `\def\crfnmPageEnumDelim{}` if you really don't want any delimiter between page numbers while the default delimiter is a comma followed by a space).

Unless specified otherwise, the examples in the following subsections correspond to the built-in configuration.

## 4.2 Prefixes, delimiters and separators

You can define the singular and plural prefixes printed before the value of the reference like this:

```

\crfnmsetup[page]{sg={p.~}, pl={pp.~}}
\crfnmsetup[note]{sg={n.~}, pl={nn.~}}
\crfnmsetup[edpage]{sg={p.~}, pl={pp.~}}
\crfnmsetup[edline]{sg={l.~}, pl={ll.~}}

```

which almost amounts to the built-in configuration:

```

\def\crfnmPage{p.~}
\def\crfnmPages{pp.~}
\def\crfnmNote{n.~}
\def\crfnmNotes{nn.~}
\let\crfnmEdpage\crfnmPage
\let\crfnmEdpages\crfnmPages
\def\crfnmEdline{l.~}
\def\crfnmEdlines{ll.~}

```

(it would have been more accurate to write `\crfnmsetup[edpage]{sg=\crfnmPage, pl=\crfnmPages}`).

The delimiters printed respectively between the successive references in an enumeration and before the last one are set so:

```

\crfnmsetup[default]{
  delimiter = {, },
  before last reference = { and }
}

```

which amounts to:

```
\def\crfnmDefaultEnumDelim{, }  
\def\crfnmDefaultBeforeLastInEnum{ and }
```

The separator in a range is set like this:

```
\crfnmsetup[default]{range separator = -}
```

which amounts to:

```
\def\crfnmDefaultRangeSep{-}
```

## 4.3 Collapsible and non-collapsible types

The configuration option `collapsible?` and the macro `\crfnmDefaultCollapsible` define if ranges are allowed. The built-in configuration corresponds to:

```
\crfnmsetup[default]{collapsible?=yes}  
\crfnmsetup[note]{collapsible?=no}
```

which amounts to:

```
\def\crfnmDefaultCollapsible{yes}  
\def\crfnmNoteCollapsible{no}
```

Thus, unless you change that, a reference to consecutive notes is formatted like “nn. 3, 4 and 5”, not like “nn. 3–5”.

Ranges are not accepted in the argument of `\crossrefenum` for non-collapsible types. This extends to double types that include a non-collapsible type (such as `pagenote` in the built-in configuration).

## 4.4 Double types

You can set like this the separator between the two values in a double reference (e.g. the page and the note numbers in a `pagenote` reference):

```
\crfnmsetup[default]{subtypes separator={, }}
```

which amounts to:

```
\def\crfnmDefaultSubtypesSep{, }
```

When more than one reference is cited in an enumeration, you may not want the first prefix to be repeated every time (e.g. you may prefer “pp. 5, n. 2; 7, n. 4” to “p. 5, n. 2; p. 7, n. 4”). In this case, set `print prefix of first subtype` or `\crfnmDefaultPrintFirstPrefix` to once. Otherwise you will get:



```
\crfnmsetup[default]{  
  print prefix of first subtype = always  
}
```

which amounts to:

```
\def\crfnmDefaultPrintFirstPrefix{always}
```

If you want to format the second subtype in a special way (e.g. in superscript), set the key formatting when second subtype either to {} (no formatting) or to a macro which will take the reference number and all its affixes as its only argument (e.g. \textsuperscript). Alternatively, you can define \crfnmDefaultFormatInSecond with one argument. What \crossrefenum comes with is:

```
\crfnmsetup[default]{formatting when second subtype = {}}
```

which amounts to:

```
\def\crfnmDefaultFormatInSecond#1{#1}
```

If you don't want any prefix to be printed in the second term of a double reference, set print prefix when second subtype? or \crfnmDefaultPrintPrefixInSecond to no (built-in: yes).

Here is how you can print the line number in superscript when it comes after the corresponding page number:

```
\crfnmsetup[edline]{
  formatting when second subtype = \textsuperscript,
  print prefix when second subtype? = no
}
```

which amounts to:

```
\def\crfnmEdlineFormatInSecond#1{\textsuperscript{#1}}
\def\crfnmEdlinePrintPrefixInSecond{no}
```

After that, `\crossrefenum[edpageline]{mylabel}` may return “p. 5<sup>10</sup>”, whereas `\crossrefenum[edline]{mylabel}` would return “l. 10”.

You can specify a specific delimiter for the second part of double references and a specific string to be printed before the last reference of the second subtype in a double reference (e.g. the last reference to a line in “p. 5, l. 10, 13, 16”, which is “16”). For instance, you may want to use the word “and” before the last note number if the reference type is a simple one (note) and a comma if it comes in second in a double reference (e.g. in `pagenote`). To achieve this, you can set `delimiter when second subtype` (`= \crfnmDefaultEnumDelimInSecond`) and `before last reference when second subtype` (`= \crfnmDefaultBeforeLastInSecond`). In the built-in configuration, there is no difference between a simple type used alone and the same simple type taken as the second subtype of a double type, but we could imagine the following:

```

\crfnmsetup[note]{
  delimiter = {; },
  before last reference = { and },
  delimiter when second subtype = {, },
  before last reference when second subtype = {, }
}
\crossrefenum[note]{lbl1, lbl2, lbl3}
= \crossrefenum[pagenote]{lbl1, lbl2, lbl3}

```

which may yield: “n. 1; 2 and 5 = p. 8, n. 1, 2, 5”.

When citing a range, the two parts of the reference can be either split (e.g. “p. 5, l. 3 – p. 7, l. 44”) or grouped (“p. 5–7, l. 3–44”). This is controlled via `group subtypes?` (= `\crfnmDefaultGroupSubtypes`), which can be set to yes or no. This works only with [collapsible types](#):

```
\crfnmsetup[default]{group subtypes? = no}
```

which amounts to:

```
\def\crfnmDefaultGroupSubtypes{no}
```

To know if a reference to “p. 6, l. 34” should be merged with “p. 7, l. 35”, *crossrefenum* needs to know if the lineation is continuous (in this case, these lines are consecutive) or per page (they are not, so they should not be merged). You can set accordingly `continuous numbering?` (= `\crfnmDefaultNumbering`

ContinuousAcrossDocument)<sup>1</sup> to yes (built-in) or no. Note that *crossrefenum* cannot merge a reference to the last line of a page and the first line of the following page if the lineation is not continuous.

In the built-in configuration, the order of the subtypes in the name of a subtype (e.g. “page” and “note” in “pagenote”) determines by default the order in which they are printed (e.g. “p. 6, n. 2” instead of “n. 2, p. 6”). If you want to change this, set order (= \crfnmDefaultOrder) to inverted (built-in: normal).

---

1. In this case, you could set more specifically \crfnmLineNumberingContinuousAcrossDocument or \crfnmEdlineNumberingContinuousAcrossDocument or use \crfnmsetup with [line] and [edline].

## 5 How to extend crossrefenum with other types and formats

Adding support for new types consists in defining the related macros in your preamble. Here is a commented example that would add support for references to lines in ConT<sub>E</sub>Xt if this feature were not already included in *crossrefenum*. I suppose that the labels are inserted in the document using the standard ConT<sub>E</sub>Xt macros, i.e. `\someLine` for line references and `\pagereference` for page references.

```
% Register the types. Take care about capitalization!

%% Set the names of the types and their corresponding macros:
%% the control sequence should always be lowercase
%% and the name of the type capitalized.
\def\crfnm@line{Line}
\def\crfnm@pageline{Pageline}

%% Add them to the lists of known types.
\crfnm@declareType[simple][\crfnm@line]
\crfnm@declareType[double][\crfnm@pageline]
```

```

%% For the double types, set the primary and the secondary type.
%% The primary type corresponds to the widest typographic unit
%% ("page" for "pagenote", "section" for "sectionpage"...).
\let\crfnm@PagelinePrimary\crfnm@page
\let\crfnm@PagelineSecondary\crfnm@line

% Define the macro used to typeset the reference number.
\def\crfnm@LineRef#1{\in[lr:b:#1]}

% Define the macro used by \crossrefenum internally
% to retrieve the raw reference number without typesetting it.
% This macro must be purely expandable.
% If you want to support multiple formats,
% see how the macro \crfnm@case is used in crossrefenum.tex.
% Note: the Lua function get_raw_ref_number is defined
% in crossrefenum.tex for ConTeXt.
\def\crfnm@getLineNumber#1{\directlua{get_raw_ref_number('lr:b:#1', 'linenumber')}}

% Define all specific configuration options in the regular way.
% Instead of the following, you can use \crfnmsetup.

```

```

%% Required
\def\crfnmLine{l.~}
\def\crfnmLines{ll.~}

%% If it differs from the defaults.
\def\crfnmLineCollapsible{yes}
\def\crfnmLineBeforeLastInSecond{, }
\def\crfnmPagelineSubtypesSep{}
\def\crfnmPagelinePrintFirstPrefix{once}
\def\crfnmLineFormatInSecond#1{\crfnmSuperscript{#1}}
\def\crfnmLinePrintPrefixInSecond{no}

```

Adding support for additional formats is far less trivial, not least because most of them do not provide a straightforward way to get reference numbers via purely expandable macros, which is required for *crossrefenum* to perform its calculations. If you have wishes or hints about this, please contact me.

## 6 Compatibility issues

With L<sup>A</sup>T<sub>E</sub>X, the *nameref* package (required by *hyperref*) must be loaded explicitly before *crossrefenum*. However, if the links in the output of `\crossrefenum` (not the numbers) point to a wrong location, you may wish to patch the macro to disable *hyperref* locally:

```
\makeatletter
\let\oldcrfnm@enum\crfnm@enum
\def\crfnm@enum[#1][#2]#3{%
  \begin{NoHyper}\oldcrfnm@enum[#1][#2]{#3}\end{NoHyper}%
}
\makeatother
```



# 7 Changes

## 7.1 2022/11/11: Version 1.0

## 7.2 2023/02/20: Version 1.0.1

`\crossrefenum` is now protected against early expansion.

## 7.3 2023/02/20: Version 1.0.2

Fixed some minor issues in version numbers and in the manual.

## 7.4 2024/04/13: Version 1.1

Various fixes in the documentation and new convenience macro `\crfnm@declareType`.

The test suite now uses *lualatex*.

## 7.5 2025/08/07: Version 1.2

The last argument of `\crossrefenum` can now be a comma-delimited list instead of a list of groups. Lists of groups are still supported and will remain so.

[Table of contents](#)

[Summary](#)

The new macro `\crfnmsetup` now provides a key-value configuration interface, thanks to Jonathan P. Spratte's [expkv](#) package. The dependency to `expkv` can be removed by deactivating the key-value interface.

Various improvements in the manual.