

# Package `regulatory`\*

Erik Nijenhuis  
erik@xerdi.com

2024-01-16

This file is maintained by **Xerdi**.  
Bug reports can be opened at  
<https://github.com/Xerdi/regulatory>.

## Abstract

The `regulatory` package is well-suited for legal professionals in a broad sense. This package encompasses common structures, such as articles, sections, parts, and definitions.

Referring within the legal domain can be a significant challenge; therefore, this package provides an elegant way of referencing, as one would expect with standard  $\text{\LaTeX}$  macros, such as for a chapter using `\section`, namely through labeling. For this purpose, the `regulatory` package introduces its own `\rrref`, `\nref`, and `\aref` macro families that provide default support for both Dutch and English.

Managing definitions uniformly across all documents can be achieved using `BibTeX`. When it comes to referencing definitions, the `regulatory` package aligns with the existing `\gls` macro family of the `glossaries` package. Referencing articles, sections, parts, and definitions is not limited to within a single document but can also be done from other documents written using the `regulatory` package. This makes cross-referencing between documents straightforward. Consider, for example, Terms and Conditions and a Maintenance Agreement that refer to each other's articles or use each other's terms. It is even possible to attach the Terms and Conditions as an appendix to the PDF file for completeness.

---

\*This document corresponds to `regulatory` 0.0.2, written on 2024-01-16.

# Contents

<b>1</b>	<b>Usage</b>	<b>3</b>
<b>2</b>	<b>Structure</b>	<b>4</b>
2.1	Markdown . . . . .	4
<b>3</b>	<b>Definitions</b>	<b>5</b>
<b>4</b>	<b>References</b>	<b>6</b>
4.1	Conjunction . . . . .	7
<b>5</b>	<b>Interdocument References</b>	<b>8</b>
<b>6</b>	<b>Language Support</b>	<b>10</b>

# 1 Usage

The **regulatory** package is explicitly designed for generating PDF documents with L<sup>A</sup>T<sub>E</sub>X. Therefore, use either **pdflatex** or **lualatex**.

```
\documentclass[dutch]{article}
\usepackage{regulatory}
\begin{document}
  \article{...}
\end{document}
```

Listing 1: main.tex

The example does not use any options. This means that **bib2gls** is active. To switch back to TeX code for definition lists, the option [*nobibdefs*] is available. Additionally, there are options such as [*md,alldefs,hidelinks,nameinlink*]. Where **md** activates Markdown support, **alldefs** lists all definitions instead of only the used definitions within the same document (useful for Terms and Conditions where not all definitions necessarily appear), **hidelinks** hides all colored borders of hyperlinks, and **nameinlink** places the hyperlink around the label.

The example of listing 1 can be generated to PDF as follows:

```
pdflatex main
# Or
lualatex main
# Or keep generating
latexmk -pvc -lualatex -interaction=nonstopmode main
```

Listing 2: Commandline examples

Suppose definition lists are used; in that case, additional steps are added to the generation process, e.g:

```
lualatex main
bib2gls main
lualatex main
lualatex main
# Or for bibtex
lualatex main
makeglossaries main
lualatex main
lualatex main
```

Listing 3: Commandline with definitions

In case **latexmk** is used, the command **bib2gls** or **makeglossaries** can be executed in a separate terminal. LatexMK automatically detects file changes and regenerates the document accordingly.

## 2 Structure

`\article` This package provides familiar structures without breaking the existing functionalities of L<sup>A</sup>T<sub>E</sub>X. For example, consider `\article{<title>}` and `\para{<title>}`. These are defined as separate macros and formatted using `titlesec`.

`\paras` For the `\paras`<sup>1</sup> and “subparagraphs”, a new environment has been created using `enumitem`. The labels for the members have been adjusted to `\thearticle`, `\arabic*` for paragraphs and `\abalphnum{<arabic*>}` for subparagraphs. For “subparagraphs”, `\abalphnum` from the `fmtcount` package is used to enumerate multiple subparagraphs. Suppose `\alph` were used; in that case, `\paras` (second level) would be limited to 26 subparagraphs. With `\abalphnum`, for example, with a value of 125, the result is ‘du’.

```
\article{Voorbeeld}
\begin{paras}
  \item \textfill
  \begin{paras}
    \item \textfill
  \end{paras}
  \item \textfill
\end{paras}

\article{Voorbeeld2}
\textfill

\para{Voorbeeld3}
\textfill
```

See listings 5 and 6 for more L<sup>A</sup>T<sub>E</sub>X examples.

### 2.1 Markdown

With the package option `md`, this package ensures that all these structures are handled. However, this means that no chapters or other standard components can be typed. Instead, the writer is specifically limited to the components described in this chapter. Refer to listing 8 for a Markdown example, and check listing 7 to see how such a Markdown source can ultimately be used from within L<sup>A</sup>T<sub>E</sub>X.

---

<sup>1</sup>Originally the naming was taken from Dutch “artikel, lid en onderdeel”, which was hard to introduce in English, due to ambiguity reasons. Therefore, section and part are aliased to `\paras`.

### 3 Definitions

For referencing definitions, `glossaries-extra` is used. This allows referencing with the `\gls{<label>}` macro family.

<code>definitions</code>	To prevent conflicts between terms, abbreviations, and definitions, this package
<code>externals</code>	adds two <code>glossary</code> types. Definitions within the same file are placed under the type <code>definitions</code> , while definitions from other documents are placed under <code>externals</code> (see section 5).
<code>\printdefs</code>	To list definitions, various macros have been added. The most straightforward one, <code>\printdefs{&lt;width of text&gt;}</code> , enumerates the definitions with a customized style. The argument <code>{&lt;width of text&gt;}</code> is provided for the alignment of definition labels and descriptions.
<code>\describe</code>	To achieve the same result with <code>\describe{&lt;label&gt;}</code> , it is first necessary to invoke <code>\glssetwidest</code> , for example, in Markdown (see listing 8). The <code>\describe</code> macro is well-suited for manually placing definition descriptions. This macro adds an anchor point, required for functional hyperlinks.
<code>\loadglsdefs</code>	To load definitions, two macros are defined. The <code>\loadglsdefs{&lt;src&gt;}</code> macro adds BibTeX libraries under the type <code>definitions</code> and has the category <code>definitions</code> . Definitions used in these libraries will be listed when <code>\printdefs</code> is called. If the package is given the <code>alldefs</code> option, then all definitions in those libraries will be listed. The listing is sorted according to the Dutch dictionary. Words not found in it are listed first.
<code>\loadextdefs</code>	For <code>\loadextdefs[&lt;category&gt;]{&lt;src&gt;}</code> , it can be useful to provide a category so that definitions from different sources can be distinguished. However, it is unwise to call this macro directly since <code>\masterdocument</code> is already smartly handling it.

## 4 References

For referencing articles, members, and parts, `zref` is used behind the scenes. All components mentioned in section 2 are configured for this purpose. However, `zref` does not provide as many formatting adjustments as `cleveref`.

`\rref` Due to this limitation, it was decided to develop entirely new variants, including `\Rref` `\rref{<label>}`. With `\rref`, one can refer to articles just as is customary for `\section`, `\subsection`, and so on. The `\rref` family consists of a total of four different macros:

<code>\rref</code>	Starting with a lowercase letter and with a hyperlink.		
	<b>Article</b>	<b>Paragraph</b>	<b>Subparagraph</b>
	2	(1)	first
<code>\rref*</code>	Starting with a lowercase letter and without a hyperlink.		
	<b>Article</b>	<b>Paragraph</b>	<b>Subparagraph</b>
	2	(1)	first
<code>\Rref</code>	Starting with an uppercase letter and with a hyperlink.		
	<b>Article</b>	<b>Paragraph</b>	<b>Subparagraph</b>
	2	(1)	First
<code>\Rref*</code>	Starting with an uppercase letter and without a hyperlink.		
	<b>Article</b>	<b>Paragraph</b>	<b>Subparagraph</b>
	2	(1)	First

In the examples above, a notable difference is already apparent in the alternative `\zref`, namely the presentation of the reference number/letter/word and a distinct type in the title. For instance, for `ex1-lid:lorem`, the title is '2.1,' and it is referenced as '(1)'.

`\nref` To reference with the corresponding designation, the macro family `\nref{<label>}` `\Nref` has been developed. This family, like `\rref`, has four variants. In the following example, for simplicity, we'll only consider `\Nref`.

	<b>Article</b>	<b>Paragraph</b>	<b>Subparagraph</b>
EN	Article 2	(1)	First subparagraph
NL	Artikel 2	Eerste lid	Onderdeel a

A notable difference with the alternative `\zcref` is that `\nref` can take into account the position of the designation. For example, consider the outcome for the subparagraph (*written ordinal* *designation*).

With the macro families `\rref` and `\nref`, a lot is already possible; however, there are still many other factors to consider when it comes to referencing. For example, the `\nref` macro already includes the correct designation, but when referring to a member, the corresponding article is not mentioned. For complete and automatic references, the `\aref{<labels...>}` has been developed. This macro family records

`\aref`  
`\Aref`

all components of the reference. Additionally, `\aref` accepts multiple labels and connects them in the correct way. This can result in a list enumeration, such as [article 2(2),] first, third and fourth subparagraph, of Example One<sup>2</sup>, or a range, like [article 2(2),] first to fourth subparagraph, of Example One. However, there is one limitation: the `nameinlink` option cannot be applied when multiple labels are provided. This limitation does not apply when only one label is given. Another additional feature is that these references are easily translatable into Dutch:

```
\selectlanguage{dutch} / \selectlanguage{english}
Zie / See \aref{ex1-sub:lorem,ex1-sub:lorem3,ex1-sub:lorem
4}
en / and \aref{ex1-sub:lorem,ex1-sub:lorem2,ex1-sub:lorem3,
ex1-sub:lorem4}.
```

Zie artikel 2, tweede lid, onderdeel a, c en d, of Example One en  
artikel 2, tweede lid, onderdeel a tot d, of Example One.

The only limitation that arises is that the reference name for the document isn't translated.

## 4.1 Conjunction

<pre>\setmiddleconjunction   {\format} \setlastconjunction \setrangeconjunction   \setconjunction     {\middle}     {\last}     {\range}</pre>	<p>The linking of labels is done through <code>zref</code>. However, thanks to the reimplementa- tion, other macros have been written in a similar way as <code>cleveref</code> does, i.e:</p> <pre>\setmiddleconjunction}{, } \setlastconjunction}{ \GetTranslation{and} } \setrangeconjunction}{ \GetTranslation{to} } \setconjunction{, }{ \GetTranslation{and} }{   \GetTranslation{to} }</pre>
--	---

<pre>\setjuncto \unsetjuncto</pre>	<p>There are also macros to easily switch to legacy notation. With <code>\setjuncto</code>, one can switch throughout the document to the conjunction word ‘<code>jo.\</code>’. Using <code>\unsetjuncto</code> restores <code>\lastconjunction</code> back to ‘<code>and</code>’. If manual changes have been made to the conjunctions earlier, those changes will be overridden by one of these macros. In that case, use <code>\setlastconjunction{&lt;value&gt;}</code> instead of <code>\unsetjuncto</code>.</p>
------------------------------------	---

---

<sup>2</sup>Example One

## 5 Interdocument References

`\refdocument`  
  `[\langle prefix \rangle]`  
  `{\langle name \rangle}`  
  `{\langle opts ... \rangle}`

Referencing Other Documents Generated with the `regulatory` Package is Quite Simple. After specifying `\refdocument` in the preamble, it is possible to refer to articles, paragraphs, and subparagraphs. The macro family `\aref` continues to work seamlessly, thanks to `\zexternaldocument` from `zref-xr`. To avoid ambiguity, a special `prefix` can be added. If a `prefix` is omitted, a default `prefix` is still applied, namely `ext-`. For instance, label `lid:lorem` becomes `ext-lid:lorem`. Note that the `prefix` does not apply when referring to definitions with `\gls`.

`\masterdocument`  
  `[\langle prefix \rangle]`  
  `{\langle name \rangle}`  
  `{\langle opts ... \rangle}`

To create a complete link with other `regulatory` documents, the `\masterdocument` macro is used. A complete link involves:

1. referring with the `\aref` family;
2. referencing definitions with the `\gls` family;
3. referring to the associated document;
4. a footnote with the document attached as an appendix to the first occurrence of a reference or definition.

It's even possible for a document to have multiple 'master' documents, as is the case with this document (in Dutch):

```
\newcommand\definitionlabel[1]{~(zie #1)}
\masterdocument[ex1-]{example1}{
  defs=example1,
  author=E. Nijenhuis,
  subject= Voorbeeld Één,
  description = Het éérste voorbeeld document,
  ref label=van Voorbeeld Één,
  def label=\definitionlabel
}

\masterdocument[ex2-]{example2}{
  defs=example2,
  author=E. Nijenhuis,
  subject= Voorbeeld Twee,
  description = Het tweede voorbeeld document,
  ref label=van Voorbeeld Twee,
  def label=\definitionlabel
}
```

Both macros `\refdocument` and `\masterdocument` have, as a third argument, a choice among the following options:

`name`                    by default, the same as the first argument of the macros.



<code>filename</code>	by default, the first argument concatenated with <code>.pdf</code> . This option can be overridden if the name does not correspond to the filename.
<code>ref label</code>	by default, this macro has no value, and in that case, the following default value is used in <code>\documentlabel: \GetTranslation {of the} \artifactssubject{#1}</code> . This macro receives the name of the document as an argument.
<code>def label</code>	by default, this macro has the following value: <code>~(\GetTranslation {see} #1)</code> . The argument contains the <b>subject</b> with a possible footnote (depending on <b>referred</b> ).
<code>footnote</code>	by default, the value is <b>true</b> , so that footnotes are added at the first occurrence. This can be set to <b>false</b> to prevent this.
<code>footnote label</code>	This macro receives the attached document with <b>subject</b> as its representation in the text. By default, this macro only prints the first argument.
<code>url</code>	this option is yet to be implemented. The purpose of this option is to indicate the source of the document in the footnote.
<code>referred</code>	This option is for internal use. <code>\documentfootnote</code> sets this value to <b>true</b> .
<code>defs</code>	This option is used to load external definition lists under this document. This way, the correct source can be mentioned for first occurrences of definitions.
<code>author</code>	This option is used in <code>\documentattachment</code> for metadata purposes in some PDF viewing applications.
<code>subject</code>	This option is used, like <b>author</b> , in the appendix.
<code>description</code>	This option is used, like <b>author</b> , in the appendix.

`\documentlabel`    For both references to definitions and articles, etc., the source is mentioned, and at the first appearance, a footnote is placed with an appendix of the document. Behind the scenes, `\documentlabel`<sup>3</sup>, `\documentfootnote`, and `\documentattachment` are invoked for articles, etc., and definitions.

`\documentfootnote`    `{<label>}`

`\documentattachment`    `[<link text>]`

`{<label>}`

These macros can be called manually. For example, `\documentfootnote{example 2}`, which results in: <sup>4</sup>.

---

<sup>3</sup>This macro uses the label of references and not definitions.

<sup>4</sup>Example Two

## 6 Language Support

Initially, this package only provided support for Dutch. When English was implemented, certain macros were added to facilitate easy switching between languages. The notation of references in English and Dutch differs to an extent that setting the language can be quite complex, but this may make it adjustable for other languages as well.

<code>\rref@setup</code> <code>{\lang}</code> <code>{\article opts...}</code> <code>{\para opts...}</code> <code>{\sub opts...}</code>	<p>The <code>\rref@setup</code> macro takes the <math>\langle language \rangle</math> as first argument. The other three arguments accept multiple options, namely:</p>
<p><b>name</b></p>	<p>the designation in lower case. For example ‘article’, ‘art.’, ‘par’, et cetera. Default for article <code>\GetTranslation{article}</code>, paragraph <code>\GetTranslation{paragraph}</code> and subparagraph <code>\GetTranslation{subparagraph}</code>.</p>
<p><b>Name</b></p>	<p>the designation starting with a capital letter. For example, ‘Article’. Like <b>name</b>, this option defaults to using <code>\GetTranslation</code>, but with an initial capital letter.</p>
<p><b>ref format</b></p>	<p>a macro with one argument, namely the current number. For example, for a member (or subparagraph in English), <code>\ordinalstringnum</code> is used for this. In such cases, it is important to consider case sensitivity using <code>\@ifrrrefcap</code>. Example: <code>\@ifrrrefcap{\Ordinalstringnum{...}}{\ordinalstringnum{...}}</code>.</p>
<p><b>label format</b></p>	<p>the order of the reference and the name. This macro takes two arguments, namely the <b>name</b> and the result of <b>ref format</b>. For example, for an article in Dutch, the order is <code>{#1 #2}</code>, for a paragraph in Dutch <code>{#2 #1}</code>, and for a paragraph/lid in English <code>\@gobble{#1}#2</code><sup>5</sup>. Note that you need to specify a macro. In the first example, you would pass <code>\mylabelformat</code> as an option and define it as: <code>\newcommand\mylabelformat[2]{#1 #2}</code>.</p>
<p><b>group conjunction</b></p>	<p>This value indicates how it should be linked to the parent part. For example (Dutch), in “artikel_1_eerste_lid,” the <b>group conjunction</b> is set to <code>,~</code> for the article.</p>
<p><b>group format</b></p>	<p>This option accepts a macro with one argument. The first argument contains all subparts. For example, in English, for a subparagraph/onderdeel, the article and paragraph/lid need to be surrounded by square brackets. The value would then be for the subpart <code>[#1]</code>.</p>

---

<sup>5</sup>`\@gobble` processes the argument but does not print anything.

These options can vary for each article, paragraph, and subparagraph.

When defining a new language, it's important to know that the default values are based on the English configuration. For example, consider the Dutch configuration, including all the used helper macros:

```

1 \newcommand\rref@reformat@noop[1]{#1}
2 \newcommand\rref@reformat@parenthesis[1]{(#1)}
3 \newcommand\rref@reformat@ordinal[1]{%
4     \@ifrrrefcap{%
5         \Ordinalstringnum{#1}%
6     }{%
7         \ordinalstringnum{#1}%
8     }%
9 }
10 \newcommand\rref@reformat@alpha[1]{%
11     \@ifrrrefcap{%
12         \ABAlphnum{#1}%
13     }{%
14         \abalphnum{#1}%
15     }%
16 }
17 \newcommand\rref@label@prefix[2]{#1 #2}
18 \newcommand\rref@label@postfix[2]{#2 #1}
19 \newcommand\rref@label@refonly[2]{\@gobble{#1}#2}
20 \newcommand\rref@group@braced[1]{\{[]#1{~}\}}
21
22 \rref@setup{dutch}{
23     group conjunction={,~}
24 }{
25     ref format=\rref@reformat@ordinal,
26     label format=\rref@label@postfix,
27     group conjunction={,~},
28     group format=\rref@reformat@noop
29 }{
30     ref format=\rref@reformat@alpha,
31     label format=\rref@label@prefix,
32     group conjunction={,~},
33     group format=\rref@reformat@noop
34 }

```

Listing 4: Dutch configuration

## Examples

```
1 \begin{center}
2   \Huge \translation{Example One}{Voorbeeld Één}
3 \end{center}
4 \article{Definitiones}\label{art:defs}
5 \printdefs{Nam dui}
6 \article{Quisque ullamcorper}\label{art:lorem}
7 \begin{paras}
8   \item \label{lid:lorem} \textfill
9   \item \label{lid:lorem2} \textfill
10  \begin{paras}
11    \item \label{sub:lorem} \textfill~\Aref{lid:lorem}.
12    \item \label{sub:lorem2} \textfill~\Aref{ex2-lid:
13      lorem}.
14    \item \label{sub:lorem3} \textfill~\Aref{lid:lorem,
15      lid:lorem2}.
16    \item \label{sub:lorem4} \textfill~\Aref{ex2-lid:
17      lorem,ex2-lid:lorem2,ex2-lid:lorem3}.
18    \item \label{sub:lorem5} \textfill~\Aref{ex2-lid:
19      lorem,ex2-lid:lorem2,ex2-lid:lorem3,ex2-lid:
20      lorem5}.
21    \item \label{sub:lorem6} \textfill~\Gls{def1} ipsum
22      dolor sit amed.
23    \item \label{sub:lorem7} \textfill~\Aref{ex2-sub:
24      lorem,ex2-sub:lorem2,ex2-sub:lorem4}.
25    \item \label{sub:lorem8} \textfill
26  \end{paras}
27 \end{paras}
28 \article{Etiam ac leo}\label{art:lorem2}
29 \para{A risus tristique nonummy}
30 \textfill
31 \article{Nulla in ipsum}\label{art:lorem3}
32 \textfill
```

Listing 5: example1.tex

```
1 \begin{center}
2   \Huge \translation{Example Two}{Voorbeeld Twee}
3 \end{center}
4 \article{Pactum}\label{art:agreement}
5 \begin{paras}
6   \item \label{lid:lorem} \textfill~\Gls{def1}.
7   \item \label{lid:lorem2} \textfill~\Gls{def2}.
8   \item \label{lid:lorem3} \textfill
```

```

9      \item \label{lid:lorem4} \textfill
10     \item \label{lid:lorem5} \textfill~\Gls{def4} ipsum
        dolor sit amed.\\
11     \printdefs{Suspendisse}
12     \begin{paras}
13         \item \label{sub:lorem} \textfill
14         \item \label{sub:lorem2} \textfill
15         \item \label{sub:lorem3} \textfill
16         \item \label{sub:lorem4} \textfill
17     \end{paras}
18 \end{paras}
19 \article{Suspendisse}
20 \para{Aliquam}
21 \textfill

```

Listing 6: example2.tex

```

1 \documentclass[12pt,dutch]{article}
2 \usepackage{babel}
3 \usepackage[md,alldefs]{regulatory}
4 \usepackage{lipsum}
5
6 \newcounter{lip}
7 \setcounter{lip}{1}
8 \newcommand\textfill{\lipsum[\arabic{lip}]\stepcounter{lip}
   }}
9
10 \refdocument[ex2-]{example2-nl}{
11     defs=example2
12     ref label=van Voorbeeld Twee
13 }
14
15 \loadglsdefs{example1}
16
17 \begin{document}
18     \begin{center}
19         \Huge Voorbeeld Markdown
20     \end{center}
21
22     \markdownInput{example.md}
23
24 \end{document}

```

Listing 7: md-example.tex

```

1 # Definitiones
2 \label{art:defs}
3 \glsssetwidest{Nam dui}
4
5 Lorem
6 : \describe{def1}
7 Nam dui
8 : \describe{def2}
9 Nulla
10 : \describe{def3}
11
12 # Quisque ullamcorper
13
14 1. \label{lid:lorem}\textfill
15 2. \label{lid:lorem2}\textfill
16     1. \label{sub:lorem}\textfill~\Aref{lid:lorem}.
17     2. \label{sub:lorem2}\textfill~\Aref{ex2-lid:lorem}.
18     3. \label{sub:lorem3}\textfill~\Aref{lid:lorem,lid:
19         lorem2}.
20     4. \label{sub:lorem4}\textfill~\Aref{ex2-lid:lorem,ex2-
21         lid:lorem2,ex2-lid:lorem3}.
22     5. \label{sub:lorem5}\textfill~\Aref{ex2-lid:lorem,ex2-
23         lid:lorem2,ex2-lid:lorem3,ex2-lid:lorem5}.
24     6. \label{sub:lorem6}\textfill~\Gls{def1} ipsum dolor
25         sit amed.
26     7. \label{sub:lorem7}\textfill~\Aref{ex2-sub:lorem,ex2-
27         sub:lorem2,ex2-sub:lorem4}.
28     8. \label{sub:lorem8}\textfill

```

Listing 8: example.md

```

1
2 @entry{def1,
3     name = {Lorem},
4     description = {\textfill}
5 }
6
7 @entry{def2,
8     name = {Nam dui},
9     description = {\textfill}
10 }
11
12 @entry{def3,
13     name = {Nulla},
14     description = {\textfill}
15 }

```

Listing 9: example1.bib

```

1
2 @entry{def4,
3     name = {Suspendisse},
4     description = {\textfill}
5 }

```

Listing 10: example2.bib