

# Het `regulatory` pakket\*

Erik Nijenhuis  
erik@xerdi.com

2024-01-27

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

## Samenvatting

Het `regulatory` pakket leent zich uitstekend voor juristen in brede zin. Dit pakket brengt veel voorkomende structuren, zoals artikelen, leden, onderdelen en definities.

Verwijzen binnen het juridisch domein kan een grote uitdaging zijn, daarom biedt dit pakket een elegante manier van verwijzen, zoals men mag verwachten bij standaard `LATEX` macro's, zoals voor een hoofdstuk `\section`, namelijk door middel van labelen. Hiervoor introduceert pakket `regulatory` zijn eigen `\rref`, `\nref` en `\aref` macrofamilies die standaard ondersteuning bieden voor Nederlands en Engels.

Het beheren van definities kan eenduidig voor alle documenten met behulp van `BibTeX`. Voor het verwijzen naar definities speelt het `regulatory` pakket in op de al bestaande `\gls` macrofamilie van `glossaries`. Het verwijzen naar artikelen, leden, onderdelen en definities beperkt zich niet alleen tot één document, maar zijn ook aan te halen vanuit andere documenten geschreven met het `regulatory` pakket. Op deze manier blijft het eenvoudig verwijzen voor documenten onderling. Denk bijvoorbeeld aan Algemene Voorwaarden en een Onderhoudsovereenkomst die naar elkaars artikelen verwijzen of elkaars begrippen gebruiken. Het is zelfs mogelijk om bij documenten de Algemene Voorwaarden als bijlage in het PDF-bestand te voegen voor de volledigheid.

---

\*Dit document correspondeert aan `regulatory` 0.0.3, geschreven op 2024-01-27.

# Inhoudsopgave

<b>1</b>	<b>Gebruik</b>	<b>3</b>
<b>2</b>	<b>Structuur</b>	<b>4</b>
2.1	Markdown . . . . .	4
<b>3</b>	<b>Definities</b>	<b>5</b>
<b>4</b>	<b>Verwijzingen</b>	<b>6</b>
4.1	Conjunctie . . . . .	7
<b>5</b>	<b>Interdocumentaire Verwijzingen</b>	<b>8</b>
<b>6</b>	<b>Taalondersteuning</b>	<b>10</b>

# 1 Gebruik

Het pakket `regulatory` is uitdrukkelijk bedoeld voor het genereren van PDF-documenten met  $\text{\LaTeX}$ . Gebruik daarom `pdflatex` of `lualatex`.

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

Codelijst 1: `main.tex`

Het voorbeeld gebruikt geen opties. Dit houdt in dat `bib2gls` actief is. Om terug te schakelen naar TeX code voor definitielijsten is er de optie [*nobibdefs*]. Verder zijn er nog de opties [*md,alldefs,hidelinks,nameinlink*]. Waar `md` Markdown support activeert, `alldefs` alle definities opsomt in plaats van alleen de gebruikte definities binnen hetzelfde document (handig voor Algemene Voorwaarden waarin niet alle definities per se voorkomen), `hidelinks` alle gekleurde kaders van hyperlinks verbergt en `nameinlink` de hyperlink om het label heen plaatst.

Het voorbeeld van codelijst 1 kan als volgt gegenereerd worden naar PDF:

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

Codelijst 2: Commandline voorbeelden

Stel er worden definitie lijsten gebruikt, dan komen er nog extra stappen bij in het generatieproces, namelijk:

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

Codelijst 3: Commandline met definities

In het geval gebruik gemaakt wordt van `latexmk`, dan kan er in een aparte terminal het commando `bib2gls` of `makeglossaries` gebruikt worden. LatexMK ziet vanzelf de bestanden wijzigen en genereert dan het document opnieuw.

## 2 Structuur

Dit pakket levert bekende structuren, zonder de bestaande functionaliteiten van `\article` L<sup>A</sup>T<sub>E</sub>X te breken. Neem bijvoorbeeld `\article{<title>}` en `\para{<title>}`. Deze `\para` zijn als aparte macro's gedefinieerd en opgemaakt met behulp van `titlesec`.

`paras` (*env.*) Voor de `paras` en “onderdelen” is een nieuwe environment aangemaakt met behulp van `enumitem`. De labels van de leden zijn aangepast naar `\thearticle.\arabic*`, voor leden en, `\abalphnum{\arabic*}`, voor onderdelen. Voor onderdelen wordt er gebruik gemaakt van `\abalphnum` uit pakket `fmtcount`<sup>1</sup> om meerdere onderdelen te kunnen opsommen. Stel er zou gebruik gemaakt worden van `\alph`, dan is `paras` beperkt tot 26 onderdelen. Bij `\abalphnum` met bijvoorbeeld een waarde van 125 is ‘du’ de uitkomst.

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

\article{Voorbeeld2}
\textfill

\para{Voorbeeld3}
\textfill
```

Zie codelijsten 5 en 6 voor meer L<sup>A</sup>T<sub>E</sub>X voorbeelden.

### 2.1 Markdown

Met de pakketoptie `md` zorgt dit pakket ervoor dat al deze structuren gehanteerd worden. Dit betekent echter wel dat er geen hoofdstukken of andere standaard onderdelen meer getypt kunnen worden. In plaats daarvan is de schrijver juist beperkt tot de onderdelen omschreven in dit hoofdstuk. Kijk naar codelijst 8 voor een markdown voorbeeld en naar codelijst 7 hoe zo'n Markdown bron uiteindelijk kan gebruikt worden vanuit L<sup>A</sup>T<sub>E</sub>X.

---

<sup>1</sup>De Nederlandse taal definitie is momenteel nog in behandeling bij pakket `fmtcount`. In de tussentijd levert dit pakket daarvoor de juist configuratie mee.

### 3 Definities

Voor het verwijzen naar definities wordt gebruik gemaakt van `glossaries-extra`. Dit zorgt ervoor dat er met de `\gls{<label>}` macrofamilie kan worden verwezen.

`definitions` (*env.*) Om conflicten tussen begrippen, afkortingen en definities te voorkomen voegt dit pakket twee `glossary` types toe. Voor definities binnen hetzelfde bestand worden ze ingedeeld onder het type `definitions`, terwijl definities van andere documenten worden ingedeeld onder `externals` (zie paragraaf 5).

`\printdefs` Om definities in te laden zijn er twee macro's gedefinieerd. De `\loadglsdefs{<src>}` macro voegt BibTeX bibliotheken toe onder het type `definitions` en heeft als categorie `definitions`. Gebruikte definities in deze bibliotheken zullen opgesomd worden wanneer `\printdefs` wordt aangeroepen. Indien de optie `alldefs` bij het pakket is meegegeven, dan zullen alle definities in die bibliotheken opgesomd worden. De opsomming wordt gesorteerd aan de hand van het Nederlandse woordenboek. Woorden die niet voorkomen daarin zullen als eerste opgesomd worden.

`\describe` Om hetzelfde resultaat te behalen met `\describe{<label>}` is het eerst nodig om `\glssetwidest` aan te roepen in bijvoorbeeld Markdown (zie codelijst 8). De macro `\describe` leent zich uitstekend om handmatig definitie beschrijvingen te plaatsen. Deze macro voegt namelijk een ankerpunt toe, vereist voor werkende hyperlinks.

`\loadglsdefs` Om definities in te laden zijn er twee macro's gedefinieerd. De `\loadglsdefs{<src>}` macro voegt BibTeX bibliotheken toe onder het type `definitions` en heeft als categorie `definitions`. Gebruikte definities in deze bibliotheken zullen opgesomd worden wanneer `\printdefs` wordt aangeroepen. Als bij het pakket de optie `alldefs` is meegegeven, dan zullen alle definities in die bibliotheken opgesomd worden. De opsomming wordt gesorteerd aan de hand van het Nederlandse woordenboek. Woorden die niet voorkomen daarin worden als eerste opgesomd.

`\loadextdefs` Voor `\loadextdefs[<category>]{<src>}` kan het handig zijn om een categorie mee te geven, zodat definities van verschillende bronnen uit elkaar gehouden kunnen worden. Het is echter niet aan te raden deze direct aan te roepen, aangezien `\masterdocument` hier al slim op in speelt.

## 4 Verwijzingen

Voor het verwijzen naar artikelen, leden en onderdelen wordt onder water gebruik gemaakt van `\zref`. Alle onderdelen genoemd in paragraaf 2 zijn hiervoor ingesteld, echter biedt `\zref` niet zoveel formaat aanpassingen als `\cleveref`.

`\rref` Door deze beperking is er voor gekozen om geheel nieuwe varianten te ontwikkelen, waaronder `\rref{<label>}`. Er kan dus met `\rref` verwezen worden naar artikelen net als gebruikelijk is voor `\section`, `\subsection`, en dergelijke. De `\rref` familie kent in totaal vier verschillende macro's:

<code>\rref</code>	Beginnend met een kleine letter en met hyperlink.	
<b>Artikel</b>	<b>Lid</b>	<b>Onderdeel</b>
2	eerste	a
<code>\rref*</code>	Beginnend met een kleine letter en zonder hyperlink.	
<b>Artikel</b>	<b>Lid</b>	<b>Onderdeel</b>
2	eerste	a
<code>\Rref</code>	Beginnend met een hoofdletter en met hyperlink.	
<b>Artikel</b>	<b>Lid</b>	<b>Onderdeel</b>
2	Eerste	A
<code>\Rref*</code>	Beginnend met een hoofdletter en zonder hyperlink.	
<b>Artikel</b>	<b>Lid</b>	<b>Onderdeel</b>
2	Eerste	A

In de voorbeelden hierboven is al een opmerkelijk verschil te zien tussen alternatief `\zref`, namelijk de presentatie van de/het verwijsnummer/letter/woord en een afwijkend type in de titel. Bijvoorbeeld voor `ex1-lid:lorem` is de titel '2.1' en wordt aangehaald met 'eerste'.

`\nref` Om te verwijzen met de bijbehorende benaming is er de macrofamilie `\nref{<label>}` ontwikkeld. Deze familie heeft net als `\rref` vier varianten. In het volgende voorbeeld gaan we voor het gemak alleen uit van `\Nref`.

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

Een opmerkelijk verschil tussen het alternatief `\zcref` is dat `\nref` rekening kan houden met de positie van de benaming. Kijk bijvoorbeeld naar de uitkomst van het lid (*<geschreven ordinaal> <benoeming>*).

Met de macrofamilies `\rref` en `\nref` is er dus al veel mogelijk, echter zijn er nog veel andere factoren die meespelen als het gaat om verwijzen. Macro `\nref` doet bijvoorbeeld wel al de juiste benaming erbij, maar bij het verwijzen naar een lid wordt er geen bijbehorend artikel genoemd. Voor volledige en automatische verwijzingen is de `\aref{<labels...>}` ontwikkeld. Deze macrofamilie noteert dus

alle onderdelen van de verwijzing. Daarnaast accepteert `\aref` meerdere labels en koppelt het de labels op de juiste manier. Dit kan een lijst opsomming geven, zoals artikel 2, tweede lid, onderdeel a, c en d, van Voorbeeld Één<sup>2</sup>, of een reeks, zoals artikel 2, tweede lid, onderdeel a tot d, van Voorbeeld Één. Hierop is alleen één tekortkoming, namelijk, de optie `nameinlink` kan niet toegepast worden wanneer er meerdere labels meegegeven worden. Deze tekortkoming geldt niet wanneer er één label wordt meegegeven. Een andere bijkomstigheid is dat deze verwijzingen gemakkelijk te vertalen zijn naar het Engels:

```
\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}.
```

See [article 2(2),] first, third and fourth subparagraph, van  
Voorbeeld Één and [article 2(2),] first to fourth subparagraph, van  
Voorbeeld Één.

De enige beperking die optreedt is dat de verwijsnaam voor het document niet vertaald is.

## 4.1 Conjunctie

```
\setmiddleconjunction Het koppelen van de labels gebeurt via zref. Er zijn echter dankzij de herimple-
  {<format>} mentatie andere macro's geschreven op een vergelijkbare manier als cleveref dat
  \setlastconjunction doet, namelijk:
\setrangeconjunction
  \setconjunction \setmiddleconjunction}{, }
  {<middle>} \setlastconjunction}{ \GetTranslation{and} }
  {<last>} \setrangeconjunction}{ \GetTranslation{to} }
  {<range>} \setconjunction{, }{ \GetTranslation{and} }{
  \GetTranslation{to} }
```

`\setjuncto` Er zijn ook macro's om gemakkelijk naar verouderde notatie te schakelen. Met  
`\unsetjuncto` `\setjuncto` kan er willekeurig in het document omgeschakeld worden naar het  
koppelwoord ' jo.\ '. Door `\unsetjuncto` te gebruiken wordt `\lastconjunction`  
weer terug gezet naar ' en '. Als eerder handmatig de koppelingen zijn veranderd,  
dan zijn die wijzigingen door één van deze macro's overschreven. Gebruik dan in  
plaats van `\unsetjuncto` `\setlastconjunction{<waarde>}`.

---

<sup>2</sup>Voorbeeld Één

## 5 Interdocumentaire Verwijzingen

Verwijzen naar andere documenten gegenereerd met het `regulatory` pakket kan vrij eenvoudig. Na het opgeven van `\refdocument` in de preamble kan er naar artikelen, leden en onderdelen verwezen worden. De macro familie `\aref` blijft dankzij `\zexternaldocument` van `zref-xr` nog steeds werken. Om ambiguïteit te voorkomen kan er een speciale `prefix` toegevoegd worden. Bij het weglaten van een `prefix` is er alsnog een `prefix`, namelijk `ext-`. Label `lid:lorem` wordt dan `ext-lid:lorem`. Let wel dat de `prefix` niet geldt voor het verwijzen naar definities met `\gls`.

`\masterdocument` Om een volledige koppeling te maken met andere `regulatory` documenten wordt de `\masterdocument` macro gebruikt. Een volledige koppeling houdt in:

1. verwijzen met de `\aref` familie;
2. verwijzen naar definities met de `\gls` familie;
3. verwijzen naar het bijbehorende document;
4. een voetnoot met het document als bijlage bij de eerste verschijning van een verwijzing of definitie.

Het kan zelfs dat een document meerdere ‘master’ documenten heeft, zoals dit document:

```
\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
}
```

Beide macro's `\refdocument` en `\masterdocument` hebben als derde argument keuze uit de volgende opties:



<code>name</code>	standaard hetzelfde als het eerste argument van de macro's.
<code>filename</code>	standaard het eerste argument met <code>.pdf</code> geconcateneerd. Deze optie kan overschreven worden wanneer de naam niet correspondeert met de bestandsnaam.
<code>ref label</code>	standaard heeft deze macro geen waarde en in dat geval wordt er in <code>\documentlabel</code> de volgende standaard waarde gebruikt: <code>\GetTranslation{of the} \artifactssubject{#1}</code> . Deze macro krijgt als argument de <code>name</code> van het document.
<code>def label</code>	standaard heeft deze macro de volgende waarde: <code>~(\GetTranslation{see} #1)</code> . Het argument bevat de <code>subject</code> met eventueel de <code>footnote</code> (afhankelijk van <code>referred</code> ).
<code>footnote</code>	standaard is de waarde <code>true</code> , zodat er voetnoten geplaatst worden bij de eerste verschijning. Deze kan naar <code>false</code> gezet worden om dit te voorkomen.
<code>footnote label</code>	deze macro krijgt als argument het bijgevoegde document met <code>subject</code> als weergave in de tekst. Standaard print deze macro enkel het eerste argument.
<code>url</code>	deze optie wordt is nog niet geïmplementeerd. De bedoeling van deze optie is om in de voetnoot de vindplaats van het document te duiden.
<code>referred</code>	deze optie is voor intern gebruik. <code>\documentfootnote</code> zet deze waarde naar <code>true</code> .
<code>defs</code>	deze optie wordt gebruikt om externe definitielijsten in te laden onder dit document. Op deze manier kan bij eerste verschijningen van definities de juiste bron vermeldt worden.
<code>author</code>	deze optie wordt in <code>\documentattachment</code> gebruikt voor metadata t.b.v. sommige PDF-weergave applicaties.
<code>subject</code>	deze optie wordt net als <code>author</code> gebruikt in de bijlage.
<code>description</code>	deze optie wordt net als <code>author</code> gebruikt in de bijlage.

`\documentlabel` `{\label}` Voor zowel verwijzingen naar definities als artikelen e.d. wordt de bron vermeldt en wordt er bij de eerste verschijning een voetnoot geplaatst met een bijlage van het exemplaar. Onder water worden `\documentlabel`<sup>3</sup> `\documentfootnote` en `\documentattachment` aangeroepen voor artikelen e.d. en definities.

`\documentfootnote` `{\label}`

`\documentattachment` `{\label}` Deze macro's kunnen ook handmatig uitgevoerd worden. Neem bijvoorbeeld `\documentfootnote{example2}` wat zou leiden tot: <sup>4</sup>.

`{\link text}`

<sup>3</sup>Deze macro gebruikt het label van verwijzingen en dus niet van definities.

<sup>4</sup>Voorbeeld Twee

## 6 Taalondersteuning

In beginsel bood dit pakket alleen ondersteuning voor Nederlands. Toen Engels geïmplementeerd werd zijn er bepaalde macro's bij gekomen, zodat er makkelijk geschakeld kan worden tussen talen. De notatie van verwijzingen in het Engels en Nederlands verschilt dusdanig dat de instelling van een taal enigszins complex kan zijn. Echter, dit maakt het mogelijk aanpasbaar voor andere talen.

<code>\rref@setup</code>	De <code>\rref@setup</code> macro heeft als eerste argument <i>&lt;de taal&gt;</i> die moet worden ingesteld. De andere drie argumenten accepteren meerdere opties, namelijk:
<code>{&lt;lang&gt;}</code>	
<code>{&lt;article opts...&gt;}</code>	
<code>{&lt;para opts...&gt;}</code>	<code>name</code>
<code>{&lt;sub opts...&gt;}</code>	de benaming in kleine letters, bijvoorbeeld 'artikel', 'art.', 'onderdeel', enzovoort. Standaard voor artikel <code>\GetTranslation{article}</code> , lid <code>\GetTranslation{paragraph}</code> en onderdeel <code>\GetTranslation{subparagraph}</code> .
	<code>Name</code>
	de benaming beginnend met een hoofdletter, bijvoorbeeld 'Artikel'. Net als <code>name</code> gebruikt deze optie standaard <code>\GetTranslation</code> , maar dan met een hoofdletter.
	<code>ref format</code>
	een macro met één argument, namelijk het huidige nummer. Voor bijvoorbeeld een lid (of subparagraph in het Engels) is hiervoor <code>\ordinalstringnum</code> gebruikt. In zo'n geval is het belangrijk om rekening te houden met hoofdletter gevoeligheid d.m.v. <code>\@ifrrrefcap</code> . Voorbeeld: <code>\@ifrrrefcap{\Ordinalstringnum{...}}{\ordinalstringnum{...}}</code> .
	<code>label format</code>
	de volgorde van de verwijzing en de naam. Deze macro krijgt twee argumenten mee, namelijk de <code>name</code> en het resultaat van <code>ref format</code> . Voor bijvoorbeeld een artikel is de volgorde <code>{#1 #2}</code> , voor een lid <code>{#2 #1}</code> , en voor een paragraph/lid in het engels <code>\@gobble{#1}#2<sup>5</sup></code> . Let wel dat je een macro dient op te geven. Bij het eerste voorbeeld zou je <code>\mylabelformat</code> als optie meegeven en zou gedefinieerd zijn als: <code>\newcommand\mylabelformat[2]{#1 #2}</code> .
	<code>group conjunction</code>
	deze waarde geeft aan hoe het dient gekoppeld te worden aan het bovenliggende onderdeel. Bijvoorbeeld, in "artikel <sub>1</sub> , eerste_lid" is de <code>group conjunction</code> bij de instelling van het artikel gelijk aan <code>,~</code> .
	<code>group format</code>
	deze optie accepteert een macro met één argument. Het eerste argument bevat alle onderliggende onderdelen. Bijvoorbeeld in het Engels dient bij een subparagraph/onderdeel het artikel en paragraph/lid omringd te worden met blokhaken. De waarde is dan bij het onderdeel <code>[#1]</code> .

---

<sup>5</sup>`\@gobble` verwerkt het argument maar print niks uit.

Deze opties kunnen verschillen per artikel, lid en onderdeel.

Bij het definiëren van een nieuwe taal is het belangrijk om te weten dat de standaard waarden gebaseerd zijn op de Engelse configuratie. Neem bijvoorbeeld de Nederlandse configuratie, inclusief alle gebruikte hulp macro's:

```
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 }
```

Codelijst 4: Nederlandse configuratie

## Voorbeelden

```
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
```

Codelijst 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
```

Codelijst 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-n1}{
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}

```

Codelijst 7: md-example.tex

```

1 # Definitiones
2 \label{art:defs}
3 \glssetwidest{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

```

Codelijst 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 }
```

Codelijst 9: example1.bib

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

Codelijst 10: example2.bib