The refstyle package*

Danie Els†

Department of Mechanical Engineering
University of Stellenbosch, South Africa.
e-mail: dnjels@sun.ac.za

2003/04/23

Overview of the refstyle package

When writing complex documents, often a large number of commands for different type of references are defined, for example:

```
\newcommand*{\eqref}[1]{eqn~(\ref{#1})}
\newcommand*{\Eqref}[1]{Equation~(\ref{#1})}
```

The refstyle package was developed to automate this process. The package provides a user interface to define sets of reference and label commands for each referable object such as an equation or a table, etc. When you declare, for example, a set of reference commands to an equation:

`\newref{eq}{`⟨*key_lst*⟩`}`

a series of commands of the format `\eq...` and `\Eq...` are produced. The configuration options are set with a list of key-values. Prefixes, inserts and other options for all the different perturbations such as capitalized first letters, singular and plural from, etc. can be defined. The configuration can be changed temporarily with an optional list of key-values when the commands are used. A direct interface to the varioref package is also provided. This enables compact reference formats:

`\eqref{e1} ...`	→	equation (1) ...
`\Eqref[vref]{e1} ...`	→	Equation (1) on page 5 ...
`\eqref[s]{e1} and ...`	→	equations (1) and ...
`\eqref[name=eq.~]{e1} ...`	→	eq. (1) ...

A range or a list of references can also be referred to in a consistent way.

`\eqref{e1,e2,e3} ...`	→	equations (1), (2) and (3) ...
`\eqrangeref[vref]{e1}{e3} ...`	→	equations (1) to (3) on page 5 ...

Templates for the different reference types and different languages can be loaded with a configuring file.

The package is aimed at large projects, enabling a consistent way of producing references throughout a project. Enough flexibility is provided to make local changes to a single reference. For large projects such as a series of books or a multi volume thesis, written as freestanding documents, a facility is provided to interface to the xr package for external document references.

*This file has version number v0.1, last revised 2003/04/23.
†Any help with proof reading of this document will be appreciated.

Contents

1 Loading the refstyle Package

The refstyle package is loaded in the preamble of the document:

(a) With a default configuration file `refstyle.cfg`.

```
\usepackage{varioref}[2001/09/04]% ← To use the vref option
\usepackage{refstyle}
```

The default config file provided with this package contains interfaces to babel for language changes. A sample configuration file, `reftmpl.cfg`, is provided in section §6. It can be generated by running `refstyle.dtx` through docstrip with the `<tmpl>` option. Modify `reftmpl.cfg` according to your requirements and place it in a directory where TEX can find it.

(b) Or with your own configuration file for a specific project:

```
\usepackage[noconfig]{refstyle}
\input{thisproject.ref}
```

(c) Or without any configuration file, but by declaring your own reference commands in the preamble.

```
\usepackage[noconfig]{refstyle}
    \newref{⟨reftype 1⟩}{⟨key_lst⟩}
    \newref{⟨reftype 2⟩}{⟨key_lst⟩}
      ⋮
```

1.1 Companion packages

The refstyle packages is intended for large projects. It is therefore important that it works together with, or has direct interfaces to the following packages:

varioref:[1] Produce sophisticated page and page range references.

hyperref:[2] To establish hyper links between the references and the labels.

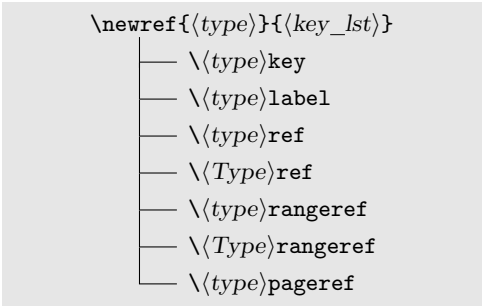
xr: To establish references to external documents.

showkeys: To show all the labels and references. This is very useful to find labels in large documents.

[1] varioref v1.3c, 2001/09/04 or later, because the starred versions of the commands are used.
[2] hyperref v6.72r, 2002/05/27 or later, where a bug for interference with varioref was fixed.

2 User Interface

`\newref` The refstyle package has one configuring command, `\newref`, that internally creates a series of label and reference commands:



All the `\⟨type⟩...` commands, excluding `\⟨type⟩key`, are robust. All the options for the referencing commands are set with a key-value list. Table 1 on page 7 gives a full list of all the key-values and defaults.

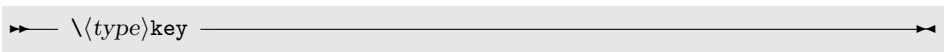
The refstyle package do not redefine any internal LaTeX commands and depends only on the `\label`, `\ref`, `\pageref` and the varioref commands. The internally defined commands do not overwrite any existing command with the same name, and an error results if a command already exists. The exception is commands declared with a previous `\newref` call, can be redefined by calling `\newref` again with a new set of parameter. If the amsmath package is loaded, and you define `\newref{eq}` for references to equations, you need to undefine `\eqref` before issuing `\newref` by

```
\let\eqref=\relax
```

3 Command Descriptions

The structure of the label and reference commands is given by the syntax diagrams that follows. Examples are included for references to equations, defined according to the template in section §6.2 with the `\newref{eq}{`⟨*key_lst*⟩`}`. See also section §4 for explanations of the key-values.

3.1 The reference key



The `\⟨type⟩key` command returns the prefix added to the argument of the label and the reference commands, for example:

`\newref{⟨type⟩}{}`	`\⟨type⟩key` → ⟨*type:*⟩
`\newref{⟨type⟩}{key=xxx-}`	`\⟨type⟩key` → `xxx-`

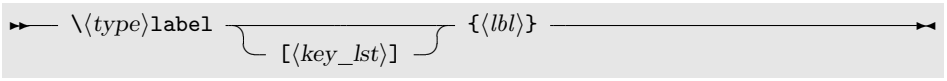
The `\⟨type⟩key` command is not a general command, but was provided only as a link to the standard LaTeX `\ref` and `\pageref` commands:

`\label{\⟨type⟩key abc}`	→ `\label{`⟨*type:*⟩`abc}`
`\ref{\⟨type⟩key abc}`	→ `\ref{`⟨*type:*⟩`abc}`

Examples: equations with `\newref{eq}{`⟨*key_lst*⟩`}`:

`\eqkey`	eq:
`in eq.~\ref{\eqkey e1} ...`	in eq. 1 ...
`on page \pageref{\eqkey e1} ...`	on page 5 ...
`\vpageref*{\eqkey e1} ...`	on this page ...

3.2 Reference label



The command `\`⟨*type*⟩`label` prefix the reference string in the `\label` with the string ⟨*type:*⟩, or its redefinition with the ***`key`*** option.

`\`⟨*type*⟩`label{abc}`	→ `\label{`⟨*type:*⟩`abc}`
`\`⟨*type*⟩`label[`*`key`*`=xxx:]{abc}`	→ `\label{xxx:abc}`

Examples:

Let $z = x + \mathrm{i}y$ and $\alpha = \beta + \mathrm{i}\gamma$, with $\mathrm{i}^2 = -1$, then

$$\mathrm{e}^{z} = \mathrm{e}^{x}\,(\cos y + \mathrm{i}\sin y) \qquad (1)\ \texttt{\textbackslash eqlabel\{e1\}}$$

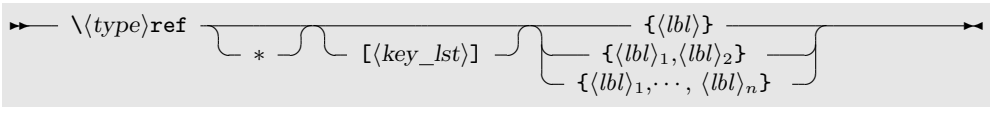
$$z^{\alpha} = \mathrm{e}^{\alpha \ln z} \qquad (2)\ \texttt{\textbackslash eqlabel\{e2\}}$$

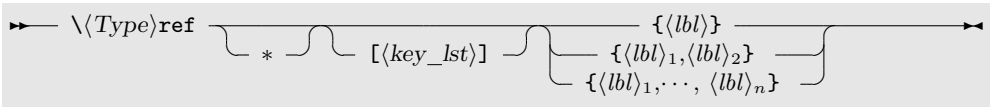
Equations (1) and (2) lead to the following interesting results:

$$\mathrm{e}^{\mathrm{i}\pi} + 1 = 0 \qquad (3)\ \texttt{\textbackslash eqlabel\{e3\}}$$

$$\mathrm{i}^{\mathrm{i}} = \mathrm{e}^{-\pi/2} \qquad (4)\ \texttt{\textbackslash eqlabel\{e4\}}$$

3.3 Reference commands





The `*` optional form of the `\`⟨*type*⟩`ref` and `\`⟨*Type*⟩`ref` commands eliminates the name prefix. The `[s]` optional key-value argument is for the plural form of the name prefix.

Examples:

`in \eqref{e1} ...`	in equation (1) ...
`in \eqref[s]{e1}--\eqref*{e4} ...`	in equations (1)–(4) ...
`in \eqref[name=eq.~]{e2} ...`	in eq. (2) ...

The reference to the page can be included with the *vref* or *vref=far* options that activates the varioref reference.

`in \eqref[vref]{e1} ...`	in equation (1) on the page before ...
`in \eqref[vref=far]{e1} ...`	in equation (1) on page 5 ...

A list of references can be used:

`in \eqref{e1,e2} ...`	in equations (1) and (2) ...
`in \eqref{e1,e2,e3} ...`	in equations (1), (2) and (3) ...

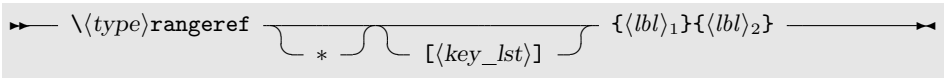
The \⟨*Type*⟩`ref` command is identical to the \⟨*type*⟩`ref` command except that it uses the *Name* and *Names* key-value options.

`\Eqref{e1} is ...`	Equation (1) is ...
`\Eqref{e1,e2} are ...`	Equations (1) and (2) are ...
`\Eqref[lsttxt={\ or~}]{e1,e2,e3}`	Equations (1), (2) or (3)

References to external documents can be added with the *xr* option. Please read the documentation of the xr package.

`\eqref[xr=A-]{xyz}` → `equation~\ref{A-eq:xyz}`

3.4 Range reference commands

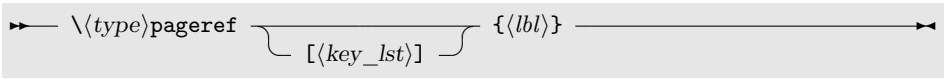


The \⟨*type*⟩`rangeref` and \⟨*Type*⟩`rangeref` commands return a range of references and take two arguments. The `*` optional form again eliminates the name prefix.

Examples:

`in \eqrangeref{e1}{e4} ...`	in equations (1) to (4) ...
`... and \eqrangeref*{e1}{e4} ...`	... and (1) to (4) ...
`\Eqrangeref{e1}{e4} are ...`	Equations (1) to (4) are ...
`\Eqrangeref[vref,rngtxt=--]{e1}{e4}`	Equations (1)–(4) on the preceding page

3.5 Page reference command



The \⟨*type*⟩`pageref` commands returns the page number of a reference.

Examples:

`it is on page \eqpageref{e1} ...`	it is on page 5 ...
`it is \eqpageref[vref]{e1} ...`	it is on the page before ...

Table 1: The available options and key-value parameters for the label and reference commands of the refstyle package.

Parameter	Default	Commands* \⟨*type*⟩key	\⟨*type*⟩label	\⟨*type*⟩ref	\⟨*Type*⟩ref	\⟨*type*⟩rangeref	\⟨*Type*⟩rangeref	\⟨*type*⟩pageref
*		□	□	■	■	■	■	□
[⟨*key_lst*⟩]		□	■	■	■	■	■	■
key	={⟨*type:*⟩},	□	■	■	■	■	■	■
s	={true}†,	□	□	■	■	□	□	□
vref	={true}†,	□	□	■	■	■	■	■
xr	={},	□	□	■	■	■	■	■
name	={},	□	□	■	□	□	□	□
names‡	={},	□	□	■	□	■	□	□
Name	={},	□	□	□	■	□	□	□
Names‡	={},	□	□	□	■	□	■	□
lsttxt	={\space and~},	□	□	■	■	□	□	□
rngtxt	={\space to~},	□	□	□	□	■	■	□
refcmd	=\ref{#1},	□	□	■	■	■	■	■

■ – Available

□ – Not available

*The commands are obtained by calling the configuring command **\newref** and setting the default key-values. The active key-values can be changed temporarily inside the commands.

†Options defaults to *true* but is initialized as *false*. The \⟨*type*⟩**ref[s]**{⟨*lbl*⟩} command results in the plural: *names*. The same principle is also valid for the *vref* option.

‡Option depends on the selection of the **s=true/false** optional key-value for singular or plural.

4 Keyval Optional Arguments

Al the options for the referencing commands are set with a key-value list. Table 1 on the preceding page gives a full list of all the key-values and defaults. The options can also be changed locally with the `[`⟨*key_lst*⟩`]` optional arguments.

4.1 Identifier: *key*

The *key* key-value is the prefix to the reference label of the `\label`, `\ref`, and `\pageref` commands. The default is *key*=⟨*type:*⟩. For reference to tables defined with the template in section §6.3:

```
\newref{tab}{...}
      \tablabel{abc}        → \label{tab:abc}
      \tabref{abc}          → table~\ref{tab:abc}
\newref{tab}{key=xxx-,...}
      \tablabel{abc}        → \label{xxx-abc}
      \tabref{abc}          → table~\ref{xxx-abc}
```

For existing documents containing labels such as `\label{tab:xx}`, you can define *key*`={}` and use the existing labels with refstyle, e.g.: `\tabref{tab:xx}`.

4.2 Plural form: *s*

The *s* conditional option (`true/false`) switches the singular/plural form of the reference on and off. The default is *s*`=true`, but it is initialized to `false`.

```
\tabref{abc}                → table~\ref{tab:abc}
\tabref[s]{abc}             → tables~\ref{tab:abc}
```

4.3 Extended reference: *vref*

The *vref* conditional option (`true/far/false`) switches the varioref page referencing on and off. The default is *vref*`=true`, but it is initialized to `false`.

```
\tabref{abc}                → table~\ref{tab:abc}
\tabref[vref]{abc}          → table~\ref{tab:abc} \vpageref[\unskip]{tab:abc}
\tabref[vref=far]{abc}      → table~\ref{tab:abc} \reftextfaraway{tab:abc}
```

4.4 External interfaces: *xr*

The *xr* option is for references to external documents. It inserts a prefix in the reference label, compatible with the xr package. The default is *xr*`={}`.

External document can be defined in the preamble with the xr package:

```
\usepackage{xr}
\externaldocument[⟨xr_key⟩]{⟨filename⟩}
```

If, for example, an external document defined with ⟨*xr_key*⟩`={A-}`, uses an identical setup (the same `refstyle.cfg`), then it can be referenced with

```
\tabref[xr=A-]{abc}         → table~\ref{A-tab:abc}
```

or otherwise

```
\tabref[xr=A-,key=]{abc}    → table~\ref{A-abc}
```

4.5 Language parameters: *name*, *names*, *Name*, *Names*, *rngtxt*, *lsttxt*

This key-values contain the text prefixes and insertions. Every house style or user has his or her own preference for naming the reference types, therefore are there no defaults provided.

name	— Inside sentence reference prefix (singular)	default={}
names	— Inside sentence reference prefix (plural)	default={}
Name	— First word reference prefix (singular)	default={}
Names	— First word reference prefix (plural)	default={}
rngtxt	— Range of references	default={\ to~}
lsttxt	— List of references	default={\ and~}

Good typographic style manuals recommend the minimum use of capital letters and punctuation that breaks the flow of a sentence or paragraph. For abbreviations, Bringhurst[1] recommends the Oxford house style: Use a period only when the word stops prematurely. The period is omitted if the abbreviation begins with the first letter and end with the last. As an example for equations, use eq. (1) or eqn (1). A good guideline is not to abbreviate any reference type names. If a sentence starts with a reference then the type name must always be written in full. A typical example for references to a table is:

```
name  ={table~},   names ={tables~},
Name  ={Table~},   Names ={Tables~},
rngtxt={\ to~},    lsttxt={\ and~},
```

Note the hardspace after the text. It is needed to keep the text and the reference together on the same line.

The refstyle configuration file can be setup to interface with babel for different languages or for automatic language changes inside a document. The language specific key-values can be added to the babel hook `\extras`⟨*language*⟩. The command `\DeclareLangOpt`[3] is provided to supply a ⟨*language*⟩ option to the package and to add the option contents to `\extras`⟨*language*⟩. The default config file contains the following lines for equations:

`\DeclareLangOpt`

```
\newcommand\RSenglish{%
   \def\RSeqtxt{equation~}%
   \def\RSeqstxt{equations~}%
   \def\RSEqtxt{Equation~}%
   \def\RSEqstxt{Equations~}%
   :
   }
\DeclareLangOpt{english}{\RSenglish}
```

The key-value options for language specific options are then set as:

```
\newref{eq}{%
   name  = \RSeqtxt,
   names = \RSeqstxt,
   Name  = \RSEqtxt,
   Names = \RSEqstxt,
   :
   }
```

[3] Only for use in `refstyle.cfg` the default config file

LaTeX/babel provides some language specific names that can be utilized.

```
\chaptername        \appendixname
\figurename         \tablename
\partname           \pagename
```

To setup a multilingual document with babel, *always* make the language options global so that other language compliant packages can detect it. A typical setup for an Afrikaans/English document would be:

```
\documentclass[UKenglish,afrikaans,⟨options⟩]{⟨LaTeX_class⟩}
\usepackage[T1]{fontenc}%..hyphenation of words with accents
\usepackage{babel}%........language def's
\usepackage{varioref}%.....for vref option
\usepackage{refstyle}
```

4.6 Reference formatting command: *refcmd*

The *refcmd* key-value holds the contents of the internal command that formats the reference. The #1 parameter passed to the command is the full reference label. For example:

refcmd=(\ref{#1}) → (\ref{⟨*label*⟩})

External commands can be employed. For example, to make references to equations identical to the $\mathcal{AMS}$ \eqref command:

refcmd={\textup{\tagform@{\ref{#1}}}}% It needs amsmath.sty

The *refcmd* can be used in conjuntion with the \ifRSstar, \ifRSnameon, \ifRSplural and \ifRScapname internal conditional variables to format the reference. As an example for a reference to a footnote, where a duplicate footnote mark is needed, can the *refcmd* be configured so that the starred form of the reference command produce a superscripted duplicate mark:

```
\newcommand{\RSfnmark}[1]{%
   \begingroup
      \unrestored@protected@xdef\@thefnmark{#1}%
   \endgroup
   \@footnotemark}
```

refcmd={\ifRSstar\RSfnmark{\ref{#1}}\else(\ref{#1})\fi}

The second footnote mark, [†], in table 1 on page 7, was obtained in this way with the reference \fnref*{b}. See section §6.1.2 for another example for references to chapters and appendices.

The nameref package can easily be incorporated if you need elaborate references which include the section or chapter name:

```
\Secref[vref, refcmd={\S\ref{#1}, `\nameref{#1}'}]{PRefCmds}
```

gives

Section §3.5, 'Page reference command' on page 6

5 Default configuration file

The default configuration file, `refstyle.cfg`, makes the following declarations:

`\newref{part}{...}`	→ Parts
`\newref{chap}{...}`	→ Chapters and Appendices
`\newref{sec}{...}`	→ Sections
`\newref{eq}{...}`	→ Equations
`\newref{fig}{...}`	→ Figures
`\newref{tab}{...}`	→ Tables
`\newref{fn}{...}`	→ Footnotes

If there is a need for more reference types, this standard list can be expanded.

At this stage the packages also contains language definitions for `afrikaans` and `english`. Other language definitions will be included if sent to me.

6 Example Templates for the refstyle Package

The user can also define his or her own templates and put it in a configuration file to ensure uniformity of reference formats in your documents.

In this section a number of reference templates are given as examples. A sample configuration file can be generated by running `refstyle.dtx` through docstrip with the `<tmpl>` option. Modify `reftmpl.cfg` according to your requirements and place it in a directory where TEX can find it.

6.1 Document divisions

6.1.1 Parts

References to parts are usually straight forward, except that the document division "Part" must be distinguished from the normal usage of the word "part" in a sentence. A personal preference is to use small caps.

```
⟨*tmpl⟩
%%--- TEMPLATE FOR PARTS -------------------------
  \newref{part}{%
     name    = {Part~},
     names   = {Parts~},
     Name    = {Part~},
     Names   = {Parts~},
     rngtxt  = {\space to~},
     lsttxt  = {\space and~}}
```

6.1.2 Chapters and Appendices

A major problem with a reference to a chapter is that at the time when the reference label, `\label{`⟨*chap-label*⟩`}`, is created, it is unknown whether it would eventually ends up in the main matter or the appendix part of a book or report.

A simple solution is to use the prefix to the ⟨*chapter*⟩ counter, `\p@chapter`, to write the definition of `\@chapapp` to the auxiliary file (`.aux`) together with information of the label. The command `\@chapapp` expands to either `\chaptername`

or \appendixname depending on whether the reference label is defined in the main matter or after the \appendix was called. (Note that \@chapapp is not defined in the $\mathcal{AMS}$ book class.)

```
\renewcommand*{\p@chapter}{\string\chname{\@chapapp}}
\newcommand*{\chname}[1]{}%<-make it harmless
\newcommand*{\chapref}[1]{{\renewcommand*{\chname}[1]{##1~}\ref{#1}}}
```

The reference to a chapter, \chapref{⟨*chap-label*⟩}, will then be prefixed with \chaptername or \appendixname and it does not matter where it was defined or where it was called.

To utilize the full functionality of the refstyle package for references to chapters and appendices, a complex form of \newref needs to be implemented. The following is an example of a template for a book or report classes.

```
%%--- TEMPLATE FOR CHAPTERS & APPENDIXES --------
```

Add \@chapapp to \p@chapter for writeout to auxiliary file. If an AMS book class is loaded, then \chaptername must be used.

```
  \makeatletter
  \providecommand*{\p@chapter}{}
  \renewcommand*{\p@chapter}{\string\chpname{\@chapapp}}
  \makeatother
  \newcommand*{\chpname}[1]{}
```

Define the \RSchpname to typeset all the different perturbations of chapter and appendix names. Use the conditionals to switch between the different options.

```
\newcommand*{\RSchpname}[1]{%
  \ifRSnameon
     \edef\RStmpa{#1}%
     \edef\RStmpb{\appendixname}%
     \ifx\RStmpa\RStmpb\relax%
        \ifRSplural \ifRScapname Appendices~\else appendices~\fi
        \else       \ifRScapname Appendix~\else   appendix~\fi
        \fi
     \else
        \ifRSplural \ifRScapname Chapters~\else chapters~\fi
        \else       \ifRScapname Chapter~\else  chapter~\fi
        \fi
     \fi
  \fi}
  \newref{chap}{%
     refcmd  = {\let\chpname=\RSchpname\ref{#1}},
     rngtxt  = {\space to~},
     lsttxt  = {\space and~}}
```

6.1.3 Sections and paragraphs

A template for references to sections is given below. References to paragraphs are similar and is left as an exercise to the users.

```
%%--- TEMPLATE FOR SECTIONS ---------------------
   \newref{sec}{%
      name    = {section~},
      names   = {sections~},
      Name    = {Section~},
      Names   = {Sections~},
      refcmd  = {\S\ref{#1}},
      rngtxt  = {\space to~},
      lsttxt  = {\space and~}}
```

6.2 Equations

The equation number in references to equations are traditionally written in an upright text, irrespective of the surrounding text.

```
%%--- TEMPLATE FOR EQUATIONS --------------------
```

Undefine `\eqref` if the amsmath.sty is loaded. If it is loaded afterwards, you need additional definitions.

```
   \let\eqref\relax
   \newref{eq}{%
      name      = {equation~},
      names     = {equations~},
      Name      = {Equation~},
      Names     = {Equations~},
      refcmd    = \textup{(\ref{#1})},
      rngtxt    = {\space to~},
      lsttxt    = {\space and~}}%
```

6.3 Figures and Tables

```
%%--- TEMPLATE FOR FIGURES ----------------------
   \newref{fig}{%
      name    = {figure~},
      names   = {figures~},
      Name    = {Figure~},
      Names   = {Figures~},
      rngtxt  = {\space to~},
      lsttxt  = {\space and~}}
```

```
%%--- TEMPLATE FOR TABLES -----------------------
   \newref{tab}{%
      name    = {table~},
      names   = {tables~},
      Name    = {Table~},
      Names   = {Tables~},
```

```
      rngtxt  = {\space to~},
      lsttxt  = {\space and~}}
```

6.4 Footnotes

A reference to a footnote differs from other references in that it is only defined inside the footnote definition itself:

```
\footnote{This is a footnote ...  \label{fn:xx}}
```

It can then be referred to with `\ref{fn:xx}`. A useful application of the refstyle package is for references to footnotes, were you need a duplicate footnote mark that refers to a previously defined footnote. You can use the stared form of the reference command to format the reference as a footnote mark, while the reference commands without a star behave as normal. The second footnote mark, [†], in table 1 on page 7, was obtained with the reference: `\fnref*{b}` while `\fnref[vref]{b}` gives: footnote (†) on page 7.

```
%%--- TEMPLATE FOR FOOTNOTES --------------------
```

Define the `\RSfnmark` to reproduce the footnote mark. Use the `\ifRSstar` conditional to switch between a superscripted and a normal reference.

```
\makeatletter
\newcommand{\RSfnmark}[1]{%
   \begingroup
     \unrestored@protected@xdef\@thefnmark{#1}%
   \endgroup
   \@footnotemark}
\makeatother

\newref{fn}{%
   name    = {footnote~},
   names   = {footnotes~},
   Name    = {Footnote~},
   Names   = {Footnotes~},
   refcmd  = {\ifRSstar\RSfnmark{\ref{#1}}\else(\ref{#1})\fi},
   rngtxt  = {\space to~},
   lsttxt  = {\space and~}}
```

6.5 Enumerated lists

A reference to an item in an enumerated list can be obtained by placing the labelling command after the `\item` command inside the list. The second level numbering of the `\ref` label in the standard LaTeX is: 2a, 2b, etc.

If you want to change the reference labels to 2(a), 2(b), etc., without effecting the display in the `enumerate` environment, you can make the following redefinition:

```
\makeatletter
\renewcommand{\p@enumii}{\expandafter\p@@enumii}
\newcommand{\p@@enumii}[1]{\theenumi(#1)}
\makeatother
```

The contents of the `\@currentlable` is then

`{\expandafter\p@@enumii\theenumii}` → `{\theenumi(\theenumii)}`

Writing a template for enumerated lists is left to the user.

6.6 Theorems, lemmas, etc.

There exists many perturbations to the theorem environment such as, Theorem, Lemma, Exercise, etc., and it is left to the user as an exercise to construct his or her own templates.

85 ⟨/tmpl⟩

References

[1] Bringhurst, R. (1996), *The elements of typographic style*, Hartley & Marks Publishers, Point Roberts, WA, USA and Vancouver, BC, Canada, second edn.

7 Implementation: refstyle.sty

7.1 Identification

```
⟨*package⟩
\NeedsTeXFormat{LaTeX2e}[1999/12/01]
\ProvidesPackage{refstyle}
   [2003/04/23 v0.1 Reference formatting (DNJ Els)]
\newcommand*{\RS@pkgname}{refstyle}
```

7.2 External packages

Load all the external packages.

```
\RequirePackage{keyval}
```

A small bug-fix for showkeys. Will be removed after release of new version.

```
\@ifundefined{vref@space}{\let\vref@space\space}{}
```

7.3 Utility commands

\RS@namelet
\RS@nameuse
\RS@namedef
\RS@robustnamedef

The following is a list of commands that take a variable {⟨*name*⟩} as argument. This enables on-the-fly definitions of user commands.

Usage:
- `\RS@namelet{`⟨*name*⟩`}` → `\let\`⟨*name*⟩
- `\RS@nameuse{`⟨*name*⟩`}` → `\`⟨*name*⟩
- `\RS@namedef{`⟨*name*⟩`}` → `\def\`⟨*name*⟩
- `\RS@robustnamedef{`⟨*name*⟩`}` → `\def\`⟨*name*⟩`{\protect\`⟨*name␣*⟩`}\def\`⟨*name␣*⟩

```
\newcommand*{\RS@namelet}[1]{\expandafter\let\csname #1\endcsname}
\newcommand*{\RS@nameuse}[1]{\csname #1\endcsname}
\newcommand*{\RS@namedef}[1]{\expandafter\def\csname #1\endcsname}
\newcommand*{\RS@robustnamedef}[1]{%
   \expandafter\edef\csname #1\endcsname{%
      \noexpand\protect\RS@nameuse{#1 }}%
   \RS@namedef{#1 }}
```

\RS@ifundefined

This is an improved definition[4] for the LaTeX kernel command `\@ifundefined` that do not leave an undefined command defined as `\relax` after the test.

The usage is: `\RS@ifundefined{`⟨*name*⟩`}{`⟨*true*⟩`}{`⟨*false*⟩`}` executes the contents of ⟨*true*⟩ if \⟨*name*⟩ is not defined and ⟨*false*⟩ if defined.

```
\def\RS@ifundefined#1{%
   \begingroup\expandafter\expandafter\expandafter\endgroup
   \expandafter\ifx\csname#1\endcsname\relax
       \expandafter\@firstoftwo
   \else
       \expandafter\@secondoftwo
   \fi}
```

\RS@removedef

The command removes the definition of a command, including robust definitions.

```
\newcommand*{\RS@removedef}[1]{%
   \RS@namelet{#1}\@undefined%
   \RS@ifundefined{#1 }{}{\RS@namelet{#1 }\@undefined}}
```

[4] Posted by Markus Kohm on c.t.t. 2002/11/11

\RS@testednamedef
\RS@testedrobustnamedef

These command are identical to \RS@namedef and \RS@robustnamedef, but only define the \⟨*name*⟩ command if it is legal. Otherwise an error message is written to the log file and the program is terminated.

```
\newcommand*{\RS@testednamedef}[1]{%
   \RS@ifnamedefinable{#1}\RS@namedef{#1}}
\newcommand*{\RS@testedrobustnamedef}[1]{%
   \RS@ifnamedefinable{#1}\RS@robustnamedef{#1}}
```

\RS@ifnamedefinable

A modified version of the LaTeX kernel command (from ltdefns.dtx).

```
\long\def\RS@ifnamedefinable #1{%
    \edef\reserved@a{#1}%
    \RS@ifundefined\reserved@a
       {\edef\reserved@b{\expandafter\@carcube \reserved@a xxx\@nil}%
       \ifx \reserved@b\@qend \RS@notdefinable\else
          \ifx \reserved@a\@qrelax \RS@notdefinable\else
            \PackageInfo{\RS@pkgname}{\@backslashchar\reserved@a\space created}%
          \fi
       \fi}%
    \RS@notdefinable}
```

\RS@notdefinable

The error message when an illegal definition is attempted.

```
\gdef\RS@notdefinable{%
   \PackageError{\RS@pkgname}{%
      Command \@backslashchar\reserved@a\space
      already defined.\MessageBreak
      Or name \@backslashchar\@qend... illegal.\MessageBreak
      It can not be redefined by the \@backslashchar newref%
      }{%
      If \@backslashchar\reserved@a\space is not important\MessageBreak
      then \protect\let\@backslashchar\reserved@a%
      =\protect\relax,\MessageBreak
      else use a different \@backslashchar newref.}%
      }
```

\RS@setbool

The command[5] \RS@setbool{⟨*conditional*⟩}{⟨*true/false*⟩} sets the ⟨*conditional*⟩ to `true` or `false`.

Usage: \RS@setbool{RSplural}{false} → \RSpluralfalse
\RS@setbool{RSplural}{true} → \RSpluraltrue

```
\newcommand*{\RS@setbool}[2]{%
  \lowercase{\def\@tempa{#2}}%
  \@ifundefined{@tempswa\@tempa}%
    {\PackageError{\RS@pkgname}%
       {You can only set the option to ‘true’ or ‘false’}\@ehc}%
    {\csname#1\@tempa\endcsname}}
```

7.4 First character case changes

\RS@firstcap

This macro[6] change the first character of a string to uppercase and returns the result in \RS@cap.

[5] Taken from the ifthen package.
[6] Posted by Dan Luecking on c.t.t.

Usage: `\RS@fistcap xxxx\@nil` then `\RS@cap` → `Xxxx`

```
\def\RS@firstcap#1#2\@nil{%
  \iffalse{\fi
    \uppercase{\edef\RS@cap{\iffalse}\fi#1}#2}}%
```

7.5 Reference building commands

`\ifRSstar` `\ifRSnameon` `\ifRScapname` `\ifRSplural`

The `\if` conditional values that are set by the reference commands. These values can be accessed by user defined key-values.

```
\newif\ifRSstar\RSstarfalse
\newif\ifRSnameon\RSnameontrue
\newif\ifRScapname\RScapnamefalse
\newif\ifRSplural\RSpluralfalse
```

`\newref`

The main user interface for template setup. It take the `#1` or ⟨*key*⟩ parameter and make it lowercase before passing it on to `\RS@newref`.

```
\newcommand*{\newref}[1]{%
   \lowercase{\def\RS@tempa{#1}}%
   \expandafter\RS@newref\expandafter{\RS@tempa}}
```

`\RS@newref`

This command configures a new template.

```
\newcommand*{\RS@newref}[2]{%
```

Clears an existing template before defining a new one.

```
   \RS@clearref{#1}%
```

Create `\ifRS@`⟨*key*⟩`vref` conditional

```
%   \expandafter\newif\csname ifRS@#1vref\endcsname%
```

Creates a series of key-values for every template that stores the setup for the specific template.

```
   \define@key{RS@#1}{key}[#1:]{\RS@namedef{RS@#1@key}{##1}}%
   \define@key{RS@#1}{s}[true]{\RS@setbool{RSplural}{##1}}%
   \define@key{RS@#1}{name}[]{\RS@namedef{RS@#1@name}{##1}}%
   \define@key{RS@#1}{names}[]{\RS@namedef{RS@#1@names}{##1}}%
   \define@key{RS@#1}{Name}[]{\RS@namedef{RS@#1@Name}{##1}}%
   \define@key{RS@#1}{Names}[]{\RS@namedef{RS@#1@Names}{##1}}%
   \define@key{RS@#1}{rngtxt}[\space to~]{\RS@namedef{RS@#1@rngtxt}{##1}}%
   \define@key{RS@#1}{lsttxt}[\space and~]{\RS@namedef{RS@#1@lsttxt}{##1}}%
   \define@key{RS@#1}{refcmd}[\ref{####1}]{\RS@namedef{RS@#1@rcmd}####1{##1}}%
   \define@key{RS@#1}{xr}[]{\RS@namedef{RS@#1@xr}{##1}}%
   \define@key{RS@#1}{vref}[true]{\RS@namedef{RS@#1vref}{##1}}%
```

Set default key-value parameters.

```
   \setkeys{RS@#1}{key,
                   s=false,
                   name,names,Name,Names,
                   rngtxt,lsttxt,
                   refcmd,
                   xr,
                   vref=false}%
```

Set key-values according to user definitions.

```
173   \setkeys{RS@#1}{#2}%
```

Build the reference commands.

```
174   \RS@buildref{#1}%
175   }
```

`\RS@clearref` Clear a reference template for redefinition. It check if the template already exists and clear it if it does.

```
176 \newcommand*{\RS@clearref}[1]{%
177   \RS@ifundefined{RS@#1@template}
178      {\RS@namedef{RS@#1@template}{#1}%
179       \PackageInfo{\RS@pkgname}%
180                   {New reference template \protect\newref{#1}}{}}
181      {\PackageInfo{\RS@pkgname}%
182                   {Reference template \protect\newref{#1} redefined}{}
183       \RS@firstcap#1\@nil
184       \RS@removedef{#1key}%
185       \RS@removedef{#1label}%
186       \RS@removedef{#1ref}%
187       \RS@removedef{\RS@cap ref}%
188       \RS@removedef{#1rangeref}%
189       \RS@removedef{\RS@cap rangeref}%
190       \RS@removedef{#1pageref}%
191      }%
192   }
```

`\RS@buildref` Build the reference commands. See table 1 for the list of commands. The `\RS@buildref{`⟨*key*⟩`}` build commands to call `\RS@cmd{`⟨*cmd*⟩`}{`⟨*key*⟩`}`, for example:

`\`⟨*key*⟩`ref` → `{\RScapnamefalse\RS@cmd{ref}{`⟨*key*⟩`}}`

```
193 \newcommand*{\RS@buildref}[1]{%
194    \RS@firstcap#1\@nil
195    \RS@testednamedef{#1key}{\RS@nameuse{RS@#1@key}}
196    \RS@testedrobustnamedef{#1label}{\RS@cmd{label}{#1}}
197    \RS@testedrobustnamedef{#1ref}{\RScapnamefalse\RS@cmd{ref}{#1}}
198    \RS@testedrobustnamedef{\RS@cap ref}{\RScapnametrue\RS@cmd{ref}{#1}}
199    \RS@testedrobustnamedef{#1rangeref}{\RScapnamefalse\RS@cmd{rangeref}{#1}}
200    \RS@testedrobustnamedef{\RS@cap rangeref}{\RScapnametrue\RS@cmd{rangeref}{#1}}
201    \RS@testedrobustnamedef{#1pageref}{\RScapnamefalse\RS@cmd{pageref}{#1}}
202    }
```

`\RS@cmd` `\RS@@cmd` The command `\RS@cmd{`⟨*cmd*⟩`}{`⟨*key*⟩`}` calls the final reference formatting commands. It checks for the starred form and set the conditionals `\ifRSstar` and `\ifRSnameon` accordingly. It also extracts the optional key-value list.

`\RS@cmd{label}{`⟨*key*⟩`}` → `\RS@label{`⟨*key*⟩`}[`⟨*key_lst*⟩`]`
`\RS@cmd{ref}{`⟨*key*⟩`}` → `\RS@ref{`⟨*key*⟩`}[`⟨*key_lst*⟩`]`
`\RS@cmd{rangeref}{`⟨*key*⟩`}` → `\RS@rangeref{`⟨*key*⟩`}[`⟨*key_lst*⟩`]`
`\RS@cmd{pageref}{`⟨*key*⟩`}` → `\RS@pageref{`⟨*key*⟩`}[`⟨*key_lst*⟩`]`

```
203 \newcommand*{\RS@cmd}[2]{%
204    \@ifstar{\RSstartrue\RSnameonfalse\RS@@cmd{#1}{#2}}%
205            {\RSstarfalse\RSnameontrue\RS@@cmd{#1}{#2}}}
```

```
\newcommand*{\RS@@cmd}[2]{%
   \@ifnextchar[%
      {\RS@nameuse{RS@#1}{#2}}%
      {\RS@nameuse{RS@#1}{#2}[]}}
```

7.6 Reference formatting commands

\RS@ref
\RS@@ref
\RS@@@ref

The command `\RS@ref{`⟨*key*⟩`}[`⟨*key_lst*⟩`]{`⟨*label_lst*⟩`}` typeset the references to the comma-separated reference label list according to the configuration for ⟨*key*⟩.

First of all, remove all spaces for the reference label list.

```
\def\RS@ref#1[#2]#3{%
   \begingroup
      \setkeys{RS@#1}{#2}%
      \edef\RS@tmpa{\zap@space#3 \@empty}%
      \edef\RS@tmpa{\noexpand\RS@@ref{#1} \RS@tmpa,\relax\noexpand\@eolst}%
      \RS@tmpa%
   \endgroup}
```

Check if there is a single or multiple references in the reference label list. If a single reference label then use the form set by the `s` key-value. If multiple reference labels the use the plural form of the name prefix.[7]

```
\def\RS@@ref#1 #2,#3\@eolst{%
   \ifx\relax#3\relax
      \RS@makename{#1}%
      \RS@makeref{#1}{#2}%
      \RS@makevpageref{#1}{#2}%
   \else
      \RSpluraltrue%
      \RS@makename{#1}%
      \RS@makeref{#1}{#2}%
      \RS@makevpageref{#1}{#2}%
      \RSnameonfalse%
      \RS@@@ref{#1} #3\@eolst%
   \fi}
```

For more than one reference in the reference list, typeset the rest of the references.

```
\def\RS@@@ref#1 #2,#3\@eolst{%
   \ifx\relax#3\relax
      \RS@nameuse{RS@#1@lsttxt}%
      \RS@makeref{#1}{#2}%
      \RS@makevpageref{#1}{#2}%
   \else
      \unskip,\space%
      \RS@makeref{#1}{#2}%
      \RS@makevpageref{#1}{#2}%
      \RS@@@ref{#1} #3\@eolst%
   \fi}
```

\RS@rangeref

The command `\RS@rangeref{`⟨*key*⟩`}[`⟨*key_lst*⟩`]{`⟨*lbl1*⟩`}{`⟨*lbl2*⟩`}` typeset the references as a range.

```
\def\RS@rangeref#1[#2]#3#4{%
   \begingroup
```

[7] The list of reference commands came from the typedref package.

```
      \setkeys{RS@#1}{#2}%
      \RSpluraltrue%
      \RS@makename{#1}%
      \RS@makeref{#1}{#3}%
      \RS@nameuse{RS@#1@rngtxt}%
      \RSnameonfalse%
      \RS@makeref{#1}{#4}%
      \RS@makevpagerefrange{#1}{#3}{#4}%
   \endgroup}
```

`\RS@pageref` The command `\RS@pageref{`⟨*key*⟩`}[`⟨*key_lst*⟩`]{`⟨*lbl*⟩`}` type the page where `{`⟨*lbl*⟩`}` was defined.

```
\def\RS@pageref#1[#2]#3{%
   \begingroup%
      \setkeys{RS@#1}{#2}%
      \RS@ifvref{#1}%
         {\mbox{}\vpageref*{\RS@lbl{#1}{#3}}}%
         {\reftextfaraway{\RS@lbl{#1}{#3}}}%
         {\pageref{\RS@lbl{#1}{#3}}}%
   \endgroup}
```

`\RS@label` The command `\RS@label{`⟨*key*⟩`}[`⟨*key_lst*⟩`]{`⟨*lbl*⟩`}` set a label according to the current configuration of the ⟨*key*⟩ reference type.

```
\def\RS@label#1[#2]#3{%
   \begingroup%
      \setkeys{RS@#1}{#2}%
      \label{\RS@nameuse{RS@#1@key}#3}%
   \endgroup}

\newcommand*{\RS@true}{true}
\newcommand*{\RS@false}{false}
\newcommand*{\RS@far}{far}
```

`\RS@ifvref` The command `\RS@ifvref{`⟨*key*⟩`}{`⟨*true*⟩`}{`⟨*faraway*⟩`}{`⟨*false*⟩`}` executes the contents of ⟨*true*⟩ if the *vref* option for the ⟨*key*⟩ reference type is true and ⟨*false*⟩ otherwise.

```
\newcommand{\RS@ifvref}[4]{%
   \edef\RS@tempa{\RS@nameuse{RS@#1vref}}%
   \ifx\RS@tempa\RS@true\relax
      #2%
   \else\ifx\RS@tempa\RS@far\relax
      #3%
   \else\ifx\RS@tempa\RS@false\relax
      #4%
   \else
      \PackageError{\RS@pkgname}%
         {You can only set the vref option to `true', `far' or `false'}\@ehc
   \fi\fi\fi}
```

`\RS@makename` The command `\RS@makename{`⟨*key*⟩`}` build the prefix to the reference commands.

```
\newcommand{\RS@makename}[1]{%
   \ifRSstar\else\ifRSnameon
      \ifRSplural
```

```
283          \ifRScapname
284             \RS@nameuse{RS@#1@Names}%
285          \else
286             \RS@nameuse{RS@#1@names}%
287          \fi
288       \else
289          \ifRScapname
290             \RS@nameuse{RS@#1@Name}%
291          \else
292             \RS@nameuse{RS@#1@name}%
293          \fi
294       \fi
295    \fi\fi
296    }
```

`\RS@lbl` This command builds the full label string for the `\ref` command.

`\RS@lbl{`⟨*key*⟩`}{`⟨*label*⟩`}` → `{`⟨*xr_key*⟩⟨*key*⟩⟨*label*⟩`}`

```
297 \newcommand*{\RS@lbl}[2]{%
298    \RS@nameuse{RS@#1@xr}\RS@nameuse{RS@#1@key}#2%
299    }
```

`\RS@makeref` The command `\RS@makeref{`⟨*key*⟩`}{`⟨*label*⟩`}` formats the `\ref` output

`\RS@makeref{`⟨*key*⟩`}{`⟨*label*⟩`}` → `\`⟨*rcmd*⟩`{\ref{`⟨*xr_key*⟩⟨*key*⟩⟨*label*⟩`}}`

```
300 \newcommand{\RS@makeref}[2]{%
301    \RS@nameuse{RS@#1@rcmd}{\RS@lbl{#1}{#2}}%
302    }
```

`\RS@makevpageref` The command `\RS@makevpageref{`⟨*key*⟩`}{`⟨*label*⟩`}` adds the varioref page reference if the *`vref`* option is true.

```
303 \newcommand{\RS@makevpageref}[2]{%
304    \RS@ifvref{#1}%
305               {\vpageref[\unskip]{\RS@lbl{#1}{#2}}}%
306               { \reftextfaraway{\RS@lbl{#1}{#2}}}%
307               {}%
308    }
```

`\RS@makevpagerefrange` The command `\RS@makevpagerefrange{`⟨*key*⟩`}{`⟨*lbl1*⟩`}{`⟨*lbl1*⟩`}` adds the varioref page range reference if the *`vref`* option is true.

```
309 \newcommand{\RS@makevpagerefrange}[3]{%
310    \RS@ifvref{#1}%
311               {\space\vpagerefrange[\unskip]{\RS@lbl{#1}{#2}}{\RS@lbl{#1}{#3}}}%
312               {\space\vpagerefrange[\unskip]{\RS@lbl{#1}{#2}}{\RS@lbl{#1}{#3}}}%
313               {}%
314    }
```

7.7 varioref command predefinitions

```
315 \AtBeginDocument{%
316   \providecommand{\vpageref}{%
317     \PackageError{\RS@pkgname}%
318        {The vref option used, but varioref.sty not loaded.}%
```

```
      {Load varioref.sty}}
\providecommand{\reftextfaraway}{%
   \PackageError{\RS@pkgname}%
      {The vref=far option used, but varioref.sty not loaded.}%
      {Load varioref.sty}}
\providecommand{\vpagerefrange}{%
   \PackageError{\RS@pkgname}%
      {The vref option used, but varioref.sty not loaded.}%
      {Load varioref.sty}}
}
```

7.8 Support for language options inclusions in config file

`\RS@addto` Command from the varioref package is used to add language definitions to the `\extras`⟨*language*⟩ token for babel.

```
\def\RS@addto#1#2{%
  #2%
  \@temptokena{#2}%
  \ifx#1\relax
    \let#1\@empty
  \fi
  \ifx#1\undefined
     \edef#1{\the\@temptokena}%
  \else
     \toks@\expandafter{#1}%
     \edef#1{\the\toks@\the\@temptokena}%
  \fi
  \@temptokena{}\toks@\@temptokena}
```

`\DeclareLangOpt` Command to declare a language option and add language definitions to the `\extras`⟨*language*⟩ token for babel.

```
\def\DeclareLangOpt#1#2{%
   \edef\RS@tempa{\expandafter\@gobble\string#2}%
   \RS@ifundefined{\RS@tempa}%
       {\PackageError{\RS@pkgname}%
          {Unknown definitions \@backslashchar\RS@tempa\MessageBreak
           for language option `#1'}{}}%
       {\DeclareOption{#1}{\expandafter\RS@addto\csname extras#1\endcsname #2}}%
}
```

7.9 Package Options

`\RS@cfgfile` Define the config file name.

```
\newcommand*{\RS@cfgfile}{refstyle.cfg}
```

We need to peek into the options list before the options are processed to find out if the config file is to be loaded or not. The config file can contain options and must be loaded before `\ProcessOptions`. Make *`noconfig`* not used afterwards.

```
\@ifpackagewith{\@currname}{noconfig}
   {\PackageInfo{\RS@pkgname}{No config file loaded}}%
   {\InputIfFileExists{\RS@cfgfile}%
      {\PackageInfo{\RS@pkgname}{Config file \RS@cfgfile\space used}}%
      {\PackageInfo{\RS@pkgname}{Config file \RS@cfgfile\space not found}}%
   }
```

357 `\DeclareOption{noconfig}{\OptionNotUsed}`

Process the options, including options in config file.

358 `\ProcessOptions*\relax`

The end of this package.

359 ⟨/package⟩

Change History

v0.0

General: Initial version 1

v0.1

General: First stable version 1

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.

T

U

V