The refstyle package[*]

Danie Els

e-mail: dnjels@gmail.com

2024/02/01

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 6 ...
`\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 6 ...

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.

See also refconfig.pdf for setup and examples.

---

[*] This file has version number v0.6b, last revised 2024/02/01.

Contents

1 Loading the **refstyle** Package

1.1 Document preamble

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

(a) With a config file with babel language support

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

When the package is loaded, it first searchers for a local user defined configuration file `refstyle.def`. If it is not available it looks for the global default config file `refstyle.cfg` provided with this package. These configuration files are loaded before the package options are precess. It can therefore contain option commands using globally defined language options and can interface to babel for language changes. The macro

```
\DeclareLangOpt{⟨language⟩}{⟨definitions⟩}
```

is provided to declare the package option ⟨*language*⟩ and add ⟨*definitions*⟩ to the `\extras`⟨*language*⟩ token for babel. It can only be used inside `refstyle.cfg` or `refstyle.def`.

(b) Or with an existing configuration with language support and addition of your own extensions

```
\documentclass[norwegian]{article}
\usepackage{babel}
\usepackage{refstyle}
\RSaddto{\RSnorwegian}{%
   \def\RSthmtxt{teorem~}%
   \def\RSthmstxt{teorem~}%
   \def\RSThmtxt{Teorem~}%
   \def\RSThmstxt{Teorem~}}
\newref{thm}{
   name      = \RSthmtxt,
   names     = \RSthmstxt,
   Name      = \RSThmtxt,
   Names     = \RSThmstxt,
   rngtxt    = \RSrngtxt,
   lsttxt    = \RSlsttxt,
   lsttwotxt = \RSlsttwotxt}
```

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

```
\usepackage[noconfig]{refstyle}
    \newref{⟨reftype1⟩}{⟨key_lst⟩}
    \newref{⟨reftype2⟩}{⟨key_lst⟩}
      ⋮
```

(d) Or with your own configuration file (without babel language support) for a specific project:

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

1.2 Package options

`noconfig` Neither of the user supplied `refstyle.def` config file or `refstyle.cfg` the config file supplied with this package are loaded. The user must declare his or her own definitions and templates. Note that these definitions are not passed on to babel's language changing mechanisms.

`nokeyprefix` Custom labeling commands `\`⟨*type*⟩`label` are generated by refstyle for every ⟨*type*⟩ definition. It defaults to

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

With the *`nokeyprefix`* option, the prefix is turned of and the labeling commands defaults to

`\`⟨*type*⟩`label{abc}` → `\label{abc}`

This is useful for old documents with existing labeling commands or where the user prefer not to use the `\`⟨*type*⟩`label` commands.

1.3 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, xr-hyper: 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:

`\newref{`⟨*type*⟩`}{`⟨*key_lst*⟩`}`

\⟨*type*⟩`key`	Reference argument prefix
\⟨*type*⟩`label`	Custom label
\⟨*type*⟩`ref`	In text reference
\⟨*Type*⟩`ref`	Start of sentence reference
\⟨*type*⟩`rangeref`	In text range reference
\⟨*Type*⟩`rangeref`	Start of sentence range reference
\⟨*type*⟩`pageref`	Page reference

All the \⟨*type*⟩... commands, excluding \⟨*type*⟩`key` and \⟨*type*⟩`label` are robust. All the options for the referencing commands are set with a key-value list. Table 1 on page 8 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 `refstyle.cfg` with the `\newref{eq}{`⟨*key_lst*⟩`}`. See also section §4 for explanations of the key-values.

3.1 The reference key

►►— \⟨*type*⟩`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:

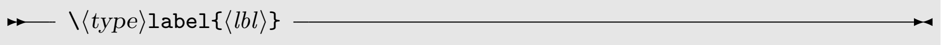
```
\label{\⟨type⟩key abc}          → \label{⟨type:⟩abc}
\ref{\⟨type⟩key abc}            → \ref{⟨type:⟩abc}
```

Note if the the ***`nokeyprefix`*** option is active then the `\`⟨*type*⟩`key` is empty.

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

`\eqkey`	eq:
`in eq.~\ref{\eqkey e1} ...`	in eq. 1 ...
`on page \pageref{\eqkey e1} ...`	on page 6 ...
`\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}`

Examples:

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

$$\mathrm{e}^z = \mathrm{e}^x(\cos y + \mathrm{i}\sin y) \tag{1}$$ `\eqlabel{e1}`

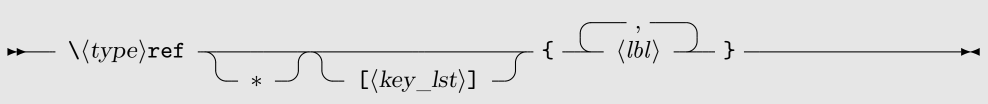
$$z^\alpha = \mathrm{e}^{\alpha \ln z} \tag{2}$$ `\eqlabel{e2}`

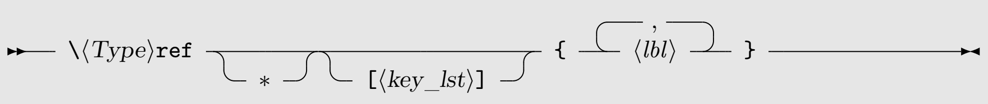
Equations (1) and (2) lead to the following interesting results:

$$\mathrm{e}^{\mathrm{i}\pi} + 1 = 0 \tag{3}$$ `\eqlabel{e3}`

$$\mathrm{i}^\mathrm{i} = \mathrm{e}^{-\pi/2} \tag{4}$$ `\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) ...

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) ...
`in \eqref[lsttxt={, and~}]{e1,e2,e3} ...`	in equations (1), (2), and (3) ...

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 previous page ...
`in \eqref[vref=far]{e1} ...`	in equation (1) on page 6 ...

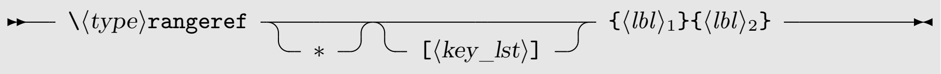
The \⟨*Type*⟩`ref` command is identical to the \⟨*type*⟩`ref` command except that it uses the *`Name`* and *`Names`* key-value options.

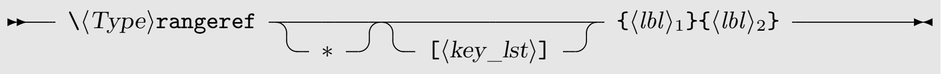
`\Eqref{e1} is ...`	Equation (1) is ...
`\Eqref{e1,e2} are ...`	Equations (1) and (2) are ...
`\Eqref[lsttxt={\ or~}]{e1,e2,e3}`	Equations (1), (2) or (3)

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



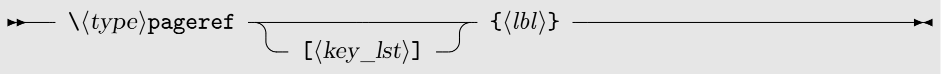


The \⟨*type*⟩`rangeref` and \⟨*Type*⟩`rangeref` commands return a range of references and take two arguments. The `*` optional form again eliminates the name prefix.

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 6 ...
`it is \eqpageref[vref]{e1} ...`	it is on the previous page ...

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

		Commands*						
Parameter	Default	\⟨*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`‡	`={},`	□	□	□	■	□	■	□
`lsttwotxt`	`={\space and~},`	□	□	■	■	□	□	□
`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 `\ref` and `\pageref` commands. The default is *key*=⟨*type*:⟩. With the default `refstyle.cfg`:

```
\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 use the *nokeyprefix* option which defines *key*=`{}`. The normal `\label` command can then be used and the reference commands defaults to

```
          \label{tab:xx}
          \tabref{tab:xx}        → table~\ref{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 or xr-hyper packages:

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

If, for example, an external document defined with ⟨*xr_key*⟩=`{A-}`, uses an identical setup (e.g. 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*, *lsttwotxt*, *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~}`
lsttwotxt	— List of references (two)	default=`{\ and~}`
lsttxt	— List of references (more than two)	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~},    lsttwotxt={\ 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}
```

---

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

`\RSaddto` or manually with the `\RSaddto` command

```
\RSaddto{\extrasenglish}{\RSenglish}
```

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

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

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 $\mathcal{A}\!\mathcal{M}\!\mathcal{S}$ `\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 8, was obtained in this way with the reference `\fnref*{b}`. See `refstyle.cfg` 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 7

5 Default configuration files

The refstyle package first searches for the `refstyle.def` configuration file, and if it not found, then it uses the default configuration file `refstyle.cfg` that is supplied with the package.

The default configuration file, `refstyle.cfg`, makes a number of default reference declarations and provides language definitions for the language parameters. See the file `refconfig.pdf` for the documentation.

Any user is welcome to customize the local copy of the `refstyle.cfg` file or copy the relevant contents to your own `refstyle.def` configuration file.

References

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

6 Implementation: **refstyle.sty**

6.1 Identification

```
⟨*pkg⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{refstyle}[\RefstyleFileDate\space
                           \RefstyleFileVersion\space
                           Reference formatting (DNJ Els)]
\newcommand*{\RS@pkgname}{refstyle}
```

6.2 External packages

Load all the external packages.

```
\RequirePackage{keyval}
```

`\RS@setkeys`

Note if xkeyval is loaded, it redefines keyval's macros. To fix this bug, we need the original `\setkeys` command.

```
\def\RS@setkeys#1#2{%
   \def\KV@prefix{KV@#1@}%
   \let\@tempc\relax
   \KV@do#2,\relax,}
```

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

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

`\@safe@activestrue`
`\@safe@activesfalse`

If babel is not loaded, make the following commands inactive.

```
\providecommand*{\@safe@activestrue}{}
\providecommand*{\@safe@activesfalse}{}
```

6.3 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.

---

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

```
\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}}
```

`\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`

---

[5] Taken from the ifthen package.

```
\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}}
```

6.4 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`.

> 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}}%
```

6.5 Reference building commands

`\ifRS@keyactive` We need to peek into the options list for the ***nokeyprefix*** option before the options are processed to find out if the key prefix must be included in the `\`⟨*key*⟩`label` command. Make ***nokeyprefix*** not used afterwards.

```
\newif\ifRS@keyactive
\@ifpackagewith{\@currname}{nokeyprefix}%
    {\RS@keyactivefalse}%
    {\RS@keyactivetrue}
\DeclareOption{nokeyprefix}{\OptionNotUsed}
```

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

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
\newif\ifRSlsttwo\RSlsttwofalse
```

`\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%
```

---

[6] Posted by Dan Luecking on c.t.t.

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

```
    \ifRS@keyactive
        \define@key{RS@#1}{key}[#1:]{\RS@namedef{RS@#1@key}{##1}}%
    \else
        \define@key{RS@#1}{key}[]{\RS@namedef{RS@#1@key}{##1}}%
    \fi
    \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}{lsttwotxt}[\space and~]{\RS@namedef{RS@#1@lsttwotxt}{##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.

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

Set key-values according to user definitions.

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

Build the reference commands.

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

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

```
\newcommand*{\RS@clearref}[1]{%
   \RS@ifundefined{RS@#1@template}
      {\RS@namedef{RS@#1@template}{#1}%
       \PackageInfo{\RS@pkgname}%
                   {New reference template \protect\newref{#1}}{}}
      {\PackageInfo{\RS@pkgname}%
                   {Reference template \protect\newref{#1} redefined}{}
       \RS@firstcap#1\@nil
       \RS@removedef{#1key}%
       \RS@removedef{#1label}%
       \RS@removedef{#1ref}%
       \RS@removedef{\RS@cap ref}%
       \RS@removedef{#1rangeref}%
       \RS@removedef{\RS@cap rangeref}%
       \RS@removedef{#1pageref}%
      }%
   }
```

`\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*⟩`}}`

```
\newcommand*{\RS@buildref}[1]{%
    \RS@firstcap#1\@nil
    \RS@testednamedef{#1key}{\RS@nameuse{RS@#1@key}}
    \RS@testednamedef{#1label}##1{\label{\RS@nameuse{RS@#1@key}##1}}
    \RS@testedrobustnamedef{#1ref}{\RScapnamefalse\RS@cmd{ref}{#1}}
    \RS@testedrobustnamedef{\RS@cap ref}{\RScapnametrue\RS@cmd{ref}{#1}}
    \RS@testedrobustnamedef{#1rangeref}{\RScapnamefalse\RS@cmd{rangeref}{#1}}
    \RS@testedrobustnamedef{\RS@cap rangeref}{\RScapnametrue\RS@cmd{rangeref}{#1}}
    \RS@testedrobustnamedef{#1pageref}{\RScapnamefalse\RS@cmd{pageref}{#1}}
    }
```

`\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{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*⟩`]`

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

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

6.6 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
        \RS@setkeys{RS@#1}{#2}%
        \@safe@activestrue%
        \edef\RS@tmpa{\zap@space#3 \@empty}%
        \@safe@activesfalse%
        \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{%
```

---

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

```
   \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%
      \RSlsttwotrue%
      \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
        \ifRSlsttwo
            \RS@nameuse{RS@#1@lsttwotxt}%
        \else
            \RS@nameuse{RS@#1@lsttxt}%
        \fi
        \RS@makeref{#1}{#2}%
        \RS@makevpageref{#1}{#2}%
    \else
        \RSlsttwofalse%
        \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
      \RS@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%
      \RS@setkeys{RS@#1}{#2}%
      \RS@ifvref{#1}%
         {\mbox{}\vpageref*{\RS@lbl{#1}{#3}}}%
         {\reftextfaraway{\RS@lbl{#1}{#3}}}%
         {\pageref{\RS@lbl{#1}{#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
         \ifRScapname
            \RS@nameuse{RS@#1@Names}%
         \else
            \RS@nameuse{RS@#1@names}%
         \fi
      \else
         \ifRScapname
            \RS@nameuse{RS@#1@Name}%
         \else
            \RS@nameuse{RS@#1@name}%
         \fi
      \fi
   \fi\fi
   }
```

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

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

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

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

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

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

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

```
\newcommand{\RS@makevpageref}[2]{%
   \RS@ifvref{#1}%
             {\vpageref[\unskip]{\RS@lbl{#1}{#2}}}%
             { \reftextfaraway{\RS@lbl{#1}{#2}}}%
             {}%
   }
```

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

```
\newcommand{\RS@makevpagerefrange}[3]{%
   \RS@ifvref{#1}%
             {\space\vpagerefrange[\unskip]{\RS@lbl{#1}{#2}}{\RS@lbl{#1}{#3}}}%
             {\space\vpagerefrange[\unskip]{\RS@lbl{#1}{#2}}{\RS@lbl{#1}{#3}}}%
             {}%
   }
```

6.7 varioref command predefinitions

```
\AtBeginDocument{%
  \providecommand{\vpageref}{%
     \PackageError{\RS@pkgname}%
        {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}}
  }
```

6.8 Support for language option inclusions in config file

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

```
\def\RSaddto#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}{%
            \AtBeginDocument{\expandafter\RSaddto\csname extras#1\endcsname #2}}}%
  }
```

6.9 Configuration files

`\RS@cfgfile` `\RS@reffile` Define the config file and alternate definition file names.

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

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.

We first test for a local config file `refstyle.def` and then for the global config file `refstyle.cfg`.

```
\@ifpackagewith{\@currname}{noconfig}%
    {\PackageInfo{\RS@pkgname}{No config file loaded}}%
    {\InputIfFileExists{\RS@reffile}%
        {\PackageInfo{\RS@pkgname}{Local config file \RS@reffile\space used}}
        {\InputIfFileExists{\RS@cfgfile}%
            {\PackageInfo{\RS@pkgname}{Config file \RS@cfgfile\space used}}%
            {\PackageInfo{\RS@pkgname}{No config file found}}}}
```

```
\DeclareOption{noconfig}{\OptionNotUsed}%
```

Process the options, including options in config file.

```
\ProcessOptions*\relax
```

The end of this package.

```
⟨/pkg⟩
```

Change History

v0.0

General: Initial version 1

v0.1

General: First stable version 1

v0.2

General: First updated version . . . 1

Remove redundant `\RS@label` 19

`\RS@buildref`: Remove robust def of `\`⟨*type*⟩`label` 17

`\RSaddto`: Rename `\RS@addto` to `\RSaddto` 20

v0.3

General: Documentation update . . 1

`\RS@setkeys`: original `\setkeys` copy 13

v0.4

- General: Add `nokeyprefix` option 1
  - Documentation update 1
- `\ifRS@keyactive`: Add *`nokeyprefix`* to turn of the key prefix 15
- `\ifRSlsttwo`: Add `\ifRSlsttwo` test for list containing only two parameter 15
- `\RS@@@ref`: Add `\ifRSlsttwo` test for list containing only two parameter 17
- `\RS@newref`: Ad `lsttwotxt` option 15
- `\RS@reffile`: Ad alternative definition file `\efstyle.def` 21

v0.5

- General: Documentation update 1
- `\@safe@activesfalse`: Add `\@safe@activestrue` 13
- `\@safe@activestrue`: Add `\@safe@activesfalse` 13
- `\DeclareLangOpt`: Add `\extras`⟨*language*⟩ to beginning of document. 21
- `\RS@ref`: Add `\@safe@activestrue` to avoid active chars is ref list 17

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.