

The `refstyle` package*

Danie Els

Department of Mechanical and Mechatronics Engineering

University of Stellenbosch, South Africa.

e-mail: dnjels@sun.ac.za

2010/11/02

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}{\langle key\_lst \rangle}
```

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 form, 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 v0.5, last revised 2010/11/02.

Contents

1	Loading the refstyle Package	3
1.1	Document preamble	3
1.2	Package options	4
1.3	Companion packages	4
2	User Interface	5
3	Command Descriptions	5
3.1	The reference key	5
3.2	Reference label	6
3.3	Reference commands	6
3.4	Range reference commands	7
3.5	Page reference command	7
4	Keyval Optional Arguments	9
4.1	Identifier: <i>key</i>	9
4.2	Plural form: <i>s</i>	9
4.3	Extended reference: <i>vref</i>	9
4.4	External interfaces: <i>xr</i>	9
4.5	Language parameters: <i>name, names, Name, Names, rngtxt, lsttwotxt, lsttxt</i>	10
4.6	Reference formatting command: <i>refcmd</i>	11
5	Default configuration files	12
6	Implementation: refstyle.sty	13
6.1	Identification	13
6.2	External packages	13
6.3	Utility commands	13
6.4	First character case changes	15
6.5	Reference building commands	15
6.6	Reference formatting commands	17
6.7	<i>varioref</i> command predefinitions	20
6.8	Support for language option inclusions in config file	20
6.9	Configuration files	21

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 searches 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 processed. 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 `\langle type \rangle label` are generated by `refstyle` for every `\langle type \rangle` definition. It defaults to

$$\langle type \rangle label\{abc\} \quad \rightarrow \quad \label\{\langle type \rangle : abc\}$$

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

$$\langle type \rangle label\{abc\} \quad \rightarrow \quad \label\{abc\}$$

This is useful for old documents with existing labeling commands or where the user prefer not to use the `\langle type \rangle 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:

<code>\newref{<type>}{<key_lst>}</code>	
— <code>\<type>key</code>	Reference argument prefix
— <code>\<type>label</code>	Custom label
— <code>\<type>ref</code>	In text reference
— <code>\<Type>ref</code>	Start of sentence reference
— <code>\<type>rangeref</code>	In text range reference
— <code>\<Type>rangeref</code>	Start of sentence range reference
— <code>\<type>pageref</code>	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 L^AT_EX 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



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 L^AT_EX `\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 $\langle type \rangle key$ is empty.

Examples: equations with $\backslash newref{eq}\langle key_lst \rangle$:

$\backslash eqkey$	eq:
in eq. $\backslash ref{\backslash eqkey e1}$...	in eq. 1 ...
on page $\backslash pageref{\backslash eqkey e1}$...	on page 6 ...
$\backslash vpageref*\backslash eqkey e1$...	on this page ...

3.2 Reference label

$\backslash \langle type \rangle label \langle lbl \rangle$

The command $\backslash \langle type \rangle label$ prefix the reference string in the $\backslash label$ with the string $\langle type : \rangle$, or its redefinition with the *key* option.

$\backslash \langle type \rangle label \{ abc \}$ \rightarrow $\backslash label \{ \langle type : \rangle abc \}$

Examples:

Let $z = x + iy$ and $\alpha = \beta + i\gamma$, with $i^2 = -1$, then

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

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

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

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

$$i^i = e^{-\pi/2} \tag{4} \backslash eqlabel{e4}$$

3.3 Reference commands

$\backslash \langle type \rangle ref \underbrace{\quad}_* \underbrace{\quad}_{[\langle key_lst \rangle]} \{ \overbrace{\langle lbl \rangle} \}$

$\backslash \langle Type \rangle ref \underbrace{\quad}_* \underbrace{\quad}_{[\langle key_lst \rangle]} \{ \overbrace{\langle lbl \rangle} \}$

The *** optional form of the $\backslash \langle type \rangle ref$ and $\backslash \langle Type \rangle ref$ commands eliminates the name prefix. The *[s]* optional key-value argument is for the plural form of the name prefix.

Examples:

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

A list of references can be used:

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

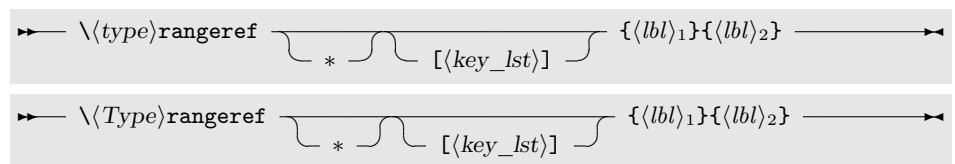
The `\(Type)ref` command is identical to the `\(type)ref` command except that it uses the `Name` and `Names` key-value options.

```
\Eqref{e1} is ...                 Equation (1) is ...
\Eqref{e1,e2} are ...             Equations (1) and (2) are ...
\Eqref[1sttxt={\ 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

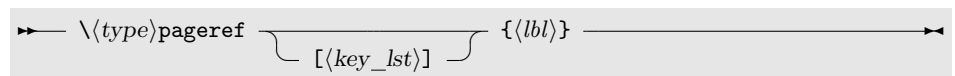


The `\(type)rangeref` and `\(Type)rangeref` commands return a range of references and take two arguments. The `*` optional form again eliminates the name prefix.

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.

Parameter	Default	Commands*						
		$\backslash\langle type \rangle key$	$\backslash\langle type \rangle label$	$\backslash\langle type \rangle ref$	$\backslash\langle Type \rangle ref$	$\backslash\langle type \rangle rangeref$	$\backslash\langle Type \rangle rangeref$	$\backslash\langle type \rangle pageref$
*		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
$[\langle key_lst \rangle]$		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<i>key</i>	$=\langle type \rangle$,	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<i>s</i>	$=\{true\}^\dagger$,	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>vref</i>	$=\{true\}^\dagger$,	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<i>xr</i>	$=\{\}$,	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<i>name</i>	$=\{\}$,	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>names</i> [‡]	$=\{\}$,	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>Name</i>	$=\{\}$,	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>Names</i> [‡]	$=\{\}$,	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<i>lsttwotxt</i>	$=\{\space and\}$,	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>lsttxt</i>	$=\{\space and\}$,	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>rngtxt</i>	$=\{\space to\}$,	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<i>refcmd</i>	$=\ref\{#1\}$,	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

– 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 `\langle type \rangle ref[s]\{ \langle lbl \rangle \}` 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

All 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}
\ref{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 $\langle xr_key \rangle = \{A-\}$, 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.

<i>name</i>	— Inside sentence reference prefix (singular)	default={}
<i>names</i>	— Inside sentence reference prefix (plural)	default={}
<i>Name</i>	— First word reference prefix (singular)	default={}
<i>Names</i>	— First word reference prefix (plural)	default={}
<i>rngtxt</i>	— Range of references	default={\ to~}
<i>lsttwotxt</i>	— List of references (two)	default={\ and~}
<i>lsttxt</i>	— 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, Bringham[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}
```

³Only 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,
  :
}
```

L^AT_EX/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 conjunction with the `\ifRSstar`, `\ifRSnameon`, `\ifRSplural` and `\ifRSscapname` 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

```
1 (*pkg)
2 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
3 \ProvidesPackage{refstyle}[\RefstyleFileDate\space
4                               \RefstyleFileVersion\space
5                               Reference formatting (DNJ Els)]
6 \newcommand*\RS@pkgname{refstyle}
```

6.2 External packages

Load all the external packages.

```
7 \RequirePackage{keyval}
```

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

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

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

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

`\@safe@activestru` If `babel` is not loaded, make the following commands inactive.

```
\@safe@activesfalse 13 \providecommand*\@safe@activestru{}
14 \providecommand*\@safe@activesfalse{}
```

6.3 Utility commands

`\RS@namelet` The following is a list of commands that take a variable `{\langle name \rangle}` as argument.

`\RS@nameuse` This enables on-the-fly definitions of user commands.

```
\RS@namedef
\RS@robustnamedef Usage: \RS@namelet{\langle name \rangle} → \let\langle name \rangle
                    \RS@nameuse{\langle name \rangle} → \langle name \rangle
                    \RS@namedef{\langle name \rangle} → \def\langle name \rangle
                    \RS@robustnamedef{\langle name \rangle} → \def\langle name \rangle{\protect\langle name_ \rangle}\def\langle name_ \rangle
```

```
15 \newcommand*\RS@namelet}[1]{\expandafter\let\csname #1\endcsname}
16 \newcommand*\RS@nameuse}[1]{\csname #1\endcsname}
17 \newcommand*\RS@namedef}[1]{\expandafter\def\csname #1\endcsname}
18 \newcommand*\RS@robustnamedef}[1]{%
19   \expandafter\edef\csname #1\endcsname{%
20     \noexpand\protect\RS@nameuse{#1 }}%
21   \RS@namedef{#1 }}
```

`\RS@ifundefined` This is an improved definition⁴ for the L^AT_EX kernel command `\@ifundefined` that do not leave an undefined command defined as `\relax` after the test.

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

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

```

22 \def\RS@ifundefined#1{%
23   \begingroup\expandafter\expandafter\endgroup
24   \expandafter\ifx\csname#1\endcsname\relax
25     \expandafter\@firstoftwo
26   \else
27     \expandafter\@secondoftwo
28   \fi}

```

`\RS@removedef` The command removes the definition of a command, including robust definitions.

```

29 \newcommand*{\RS@removedef}[1]{%
30   \RS@namelet{#1}\@undefined%
31   \RS@ifundefined{#1 }{-}{\RS@namelet{#1 }\@undefined}}

```

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

```

32 \newcommand*{\RS@testednamedef}[1]{%
33   \RS@ifnamedefinable{#1}\RS@namedef{#1}}
34 \newcommand*{\RS@testedrobustnamedef}[1]{%
35   \RS@ifnamedefinable{#1}\RS@robustnamedef{#1}}

```

`\RS@ifnamedefinable` A modified version of the L^AT_EX kernel command (from `ltxdefns.dtx`).

```

36 \long\def\RS@ifnamedefinable #1{%
37   \edef\reserved@a{#1}%
38   \RS@ifundefined\reserved@a
39     {\edef\reserved@b{\expandafter\@carcube \reserved@a xxx\@nil}%
40     \ifx \reserved@b\@qend \RS@notdefinable\else
41       \ifx \reserved@a\@qrelax \RS@notdefinable\else
42         \PackageInfo{\RS@pkgname}{\@backslashchar\reserved@a\space created}%
43       \fi
44     \fi}%
45   \RS@notdefinable}

```

`\RS@notdefinable` The error message when an illegal definition is attempted.

```

46 \gdef\RS@notdefinable{%
47   \PackageError{\RS@pkgname}{%
48     Command \@backslashchar\reserved@a\space
49     already defined.\MessageBreak
50     Or name \@backslashchar\@qend... illegal.\MessageBreak
51     It can not be redefined by the \@backslashchar newref%
52   }{%
53     If \@backslashchar\reserved@a\space is not important\MessageBreak
54     then \protect\let\@backslashchar\reserved@a%
55     =\protect\relax,\MessageBreak
56     else use a different \@backslashchar newref.}%
57   }

```

`\RS@setbool` The command⁵ `\RS@setbool{\langle conditional \rangle}{\langle true/false \rangle}` sets the `\langle conditional \rangle` to true or false.

```

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

```

⁵Taken from the `ifthen` package.

```

58 \newcommand*{\RS@setbool}[2]{%
59   \lowercase{\def\@tempa{#2}}%
60   \ifundefined{\tempwa\@tempa}%
61     {\PackageError{\RS@pkgname}%
62       {You can only set the option to ‘true’ or ‘false’}\@ehc}%
63     {\csname#1\@tempa\endcsname}}

```

6.4 First character case changes

`\RS@firstcap` This macro⁶ change the first character of a string to uppercase and returns the result in `\RS@cap`.

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

```

64 \def\RS@firstcap#1#2\@nil{%
65   \iffalse{\fi
66     \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.

```

67 \newif\ifRS@keyactive
68 \ifpackagewith{\@currname}{nokeyprefix}%
69   {\RS@keyactivefalse}%
70   {\RS@keyactivetrue}
71 \DeclareOption{nokeyprefix}{\@OptionNotUsed}

```

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

```

72 \newif\ifRSstar\RSstarfalse
73 \newif\ifRSnameon\RSnameontrue
74 \newif\ifRSscapname\RSscapnamefalse
75 \newif\ifRSplural\RSpluralfalse
76 \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`.

```

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

```

`\RS@newref` This command configures a new template.

```

80 \newcommand*{\RS@newref}[2]{%
      Clears an existing template before defining a new one.
81   \RS@clearref{#1}%
      Create \ifRS@⟨key⟩vref conditional
82 %   \expandafter\newif\csname ifRS@#1vref\endcsname%

```

⁶Posted by Dan Luecking on c.t.t.

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

```

83 \ifRS@keyactive
84   \define@key{RS@#1}{key}[#1:]{\RS@namedef{RS@#1@key}{##1}}%
85 \else
86   \define@key{RS@#1}{key}[]{\RS@namedef{RS@#1@key}{##1}}%
87 \fi
88 \define@key{RS@#1}{s}[true]{\RS@setbool{RS@plural}{##1}}%
89 \define@key{RS@#1}{name}[]{\RS@namedef{RS@#1@name}{##1}}%
90 \define@key{RS@#1}{names}[]{\RS@namedef{RS@#1@names}{##1}}%
91 \define@key{RS@#1}{Name}[]{\RS@namedef{RS@#1@Name}{##1}}%
92 \define@key{RS@#1}{Names}[]{\RS@namedef{RS@#1@Names}{##1}}%
93 \define@key{RS@#1}{rngtxt}[\space to~]{\RS@namedef{RS@#1@rngtxt}{##1}}%
94 \define@key{RS@#1}{lsttwotxt}[\space and~]{\RS@namedef{RS@#1@lsttwotxt}{##1}}%
95 \define@key{RS@#1}{lsttxt}[\space and~]{\RS@namedef{RS@#1@lsttxt}{##1}}%
96 \define@key{RS@#1}{refcmd}[\ref{###1}]{\RS@namedef{RS@#1@rcmd}###1{##1}}%
97 \define@key{RS@#1}{xr}[]{\RS@namedef{RS@#1@xr}{##1}}%
98 \define@key{RS@#1}{vref}[true]{\RS@namedef{RS@#1@vref}{##1}}%

```

Set default key-value parameters.

```

99 \RS@setkeys{RS@#1}{key,
100     s=false,
101     name,names,Name,Names,
102     rngtxt,lsttwotxt,lsttxt,
103     refcmd,
104     xr,
105     vref=false}%

```

Set key-values according to user definitions.

```

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

```

Build the reference commands.

```

107 \RS@buildref{#1}%
108 }

```

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

```

109 \newcommand*{\RS@clearref}[1]{%
110   \RS@ifundefined{RS@#1@template}
111     {\RS@namedef{RS@#1@template}{##1}%
112      \PackageInfo{\RS@pkgname}%
113        {New reference template \protect\newref{##1}}{}}
114     {\PackageInfo{\RS@pkgname}%
115      {Reference template \protect\newref{##1} redefined}{}}
116   \RS@firstcap#1\@nil
117   \RS@removedef{##1key}%
118   \RS@removedef{##1label}%
119   \RS@removedef{##1ref}%
120   \RS@removedef{\RS@cap ref}%
121   \RS@removedef{##1rangeref}%
122   \RS@removedef{\RS@cap rangeref}%
123   \RS@removedef{##1pageref}%
124   }%
125 }

```


`\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` → `{\RS@namefalse\RS@cmd{ref}{<key>}}`

```

126 \newcommand*{\RS@buildref}[1]{%
127   \RS@firstcap#1\@nil
128   \RS@testednamedef{#1key}{\RS@nameuse{RS@#1@key}}
129   \RS@testednamedef{#1label}##1{\label{\RS@nameuse{RS@#1@key}##1}}
130   \RS@testedrobustnamedef{#1ref}{\RS@namefalse\RS@cmd{ref}{#1}}
131   \RS@testedrobustnamedef{\RS@cap ref}{\RS@namefalse\RS@cmd{ref}{#1}}
132   \RS@testedrobustnamedef{#1rangeref}{\RS@namefalse\RS@cmd{rangeref}{#1}}
133   \RS@testedrobustnamedef{\RS@cap rangeref}{\RS@namefalse\RS@cmd{rangeref}{#1}}
134   \RS@testedrobustnamedef{#1pageref}{\RS@namefalse\RS@cmd{pageref}{#1}}
135 }

```

`\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_1st>]`
`\RS@cmd{rangeref}{<key>}` → `\RS@rangeref{<key>}[<key_1st>]`
`\RS@cmd{pageref}{<key>}` → `\RS@pageref{<key>}[<key_1st>]`

```

136 \newcommand*{\RS@cmd}[2]{%
137   \@ifstar{\RSstartrue\RSnameonfalse\RS@@cmd{#1}{#2}}%
138   {\RSstarfalse\RSnameontrue\RS@@cmd{#1}{#2}}
139 \newcommand*{\RS@@cmd}[2]{%
140   \@ifnextchar[%
141     {\RS@nameuse{RS@#1}{#2}}%
142     {\RS@nameuse{RS@#1}{#2} []}]

```

6.6 Reference formatting commands

`\RS@ref` The command `\RS@ref{<key>}[<key_1st>]{<label_1st>}` typeset the references to the comma-separated reference label list according to the configuration for `<key>`.
`\RS@@ref` First of all, remove all spaces for the reference label list.

```

143 \def\RS@ref#1[#2]#3{%
144   \begingroup
145     \RS@setkeys{RS@#1}{#2}%
146     \@safe@activestruer%
147     \edef\RS@tmpa{\zap@space#3 \@empty}%
148     \@safe@activesfalse%
149     \edef\RS@tmpa{\noexpand\RS@ref{#1} \RS@tmpa,\relax\noexpand\@eolst}%
150     \RS@tmpa%
151   \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

```

152 \def\RS@@ref#1 #2,#3\@eolst{%

```

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

```

153 \ifx\relax#3\relax
154   \RS@makename{#1}%
155   \RS@makeref{#1}{#2}%
156   \RS@makevpageref{#1}{#2}%
157 \else
158   \RS@pluraltrue%
159   \RS@makename{#1}%
160   \RS@makeref{#1}{#2}%
161   \RS@makevpageref{#1}{#2}%
162   \RS@nameonfalse%
163   \RS@lstwotwotrue%
164   \RS@@@ref{#1} #3\@eolst%
165 \fi}

```

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

```

166 \def\RS@@@ref#1 #2,#3\@eolst{%
167   \ifx\relax#3\relax
168     \ifRS@lstwo
169       \RS@nameuse{RS@#1@lstwotxt}%
170     \else
171       \RS@nameuse{RS@#1@lsttxt}%
172     \fi
173     \RS@makeref{#1}{#2}%
174     \RS@makevpageref{#1}{#2}%
175   \else
176     \RS@lstwofalse%
177     \unskip, \space%
178     \RS@makeref{#1}{#2}%
179     \RS@makevpageref{#1}{#2}%
180     \RS@@@ref{#1} #3\@eolst%
181 \fi}

```

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

```

182 \def\RS@rangeref#1[#2]#3#4{%
183   \begingroup
184   \RS@setkeys{RS@#1}{#2}%
185   \RS@pluraltrue%
186   \RS@makename{#1}%
187   \RS@makeref{#1}{#3}%
188   \RS@nameuse{RS@#1@rngtxt}%
189   \RS@nameonfalse%
190   \RS@makeref{#1}{#4}%
191   \RS@makevpagerefrange{#1}{#3}{#4}%
192 \endgroup}

```

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

```

193 \def\RS@pageref#1[#2]#3{%
194   \begingroup%
195   \RS@setkeys{RS@#1}{#2}%
196   \RS@ifvref{#1}%
197     {\mbox{}}\vpageref*{\RS@lbl{#1}{#3}}}%
198     {\reftextfaraway{\RS@lbl{#1}{#3}}}%

```

```

199         {\pageref{\RS@lbl{#1}{#3}}}%
200     \endgroup}

```

```

201 \newcommand*\RS@true}{true}
202 \newcommand*\RS@false}{false}
203 \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.

```

204 \newcommand{\RS@ifvref}[4]{%
205     \edef\RS@tempa{\RS@nameuse{RS@#1vref}}%
206     \ifx\RS@tempa\RS@true\relax
207         #2%
208     \else\ifx\RS@tempa\RS@far\relax
209         #3%
210     \else\ifx\RS@tempa\RS@false\relax
211         #4%
212     \else
213         \PackageError{\RS@pkgname}%
214             {You can only set the vref option to ‘true’, ‘far’ or ‘false’}\@ehc
215     \fi\fi\fi}

```

`\RS@makename` The command `\RS@makename{<key>}` build the prefix to the reference commands.

```

216 \newcommand{\RS@makename}[1]{%
217     \ifRSstar\else\ifRSnameon
218         \ifRSplural
219             \ifRScapname
220                 \RS@nameuse{RS@#1@Names}%
221             \else
222                 \RS@nameuse{RS@#1@names}%
223             \fi
224         \else
225             \ifRScapname
226                 \RS@nameuse{RS@#1@Name}%
227             \else
228                 \RS@nameuse{RS@#1@name}%
229             \fi
230         \fi
231     \fi\fi
232 }

```

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

$$\RS@lbl{<key>}{<label>} \rightarrow \{<xr_key><key><label>\}$$

```

233 \newcommand*\RS@lbl}[2]{%
234     \RS@nameuse{RS@#1@xr}\RS@nameuse{RS@#1@key}#2%
235 }

```

`\RS@makeref` The command `\RS@makeref{<key>}{<label>}` formats the `\ref` output

$$\RS@makeref{<key>}{<label>} \rightarrow \{<rcmd>\ref{<xr_key><key><label>\}$$

```

236 \newcommand{\RS@makeref}[2]{%
237   \RS@nameuse{RS@#1@rcmd}{\RS@lbl{#1}{#2}}%
238   }

```

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

```

239 \newcommand{\RS@makevpageref}[2]{%
240   \RS@ifvref{#1}%
241     {\vpageref[\unskip]{\RS@lbl{#1}{#2}}}%
242     {\reftextfaraway{\RS@lbl{#1}{#2}}}%
243     }%
244   }

```

`\RS@makevpagerefrange` The command `\RS@makevpagerefrange{<key>}{<lbl1>}{<lbl2>}` adds the `varioref` page range reference if the `vref` option is true.

```

245 \newcommand{\RS@makevpagerefrange}[3]{%
246   \RS@ifvref{#1}%
247     {\space\vpagerefrange[\unskip]{\RS@lbl{#1}{#2}}{\RS@lbl{#1}{#3}}}%
248     {\space\vpagerefrange[\unskip]{\RS@lbl{#1}{#2}}{\RS@lbl{#1}{#3}}}%
249     }%
250   }

```

6.7 varioref command predefinitions

```

251 \AtBeginDocument{%
252   \providecommand{\vpageref}{%
253     \PackageError{\RS@pkgname}%
254     {The vref option used, but varioref.sty not loaded.}%
255     {Load varioref.sty}}
256   \providecommand{\reftextfaraway}{%
257     \PackageError{\RS@pkgname}%
258     {The vref=far option used, but varioref.sty not loaded.}%
259     {Load varioref.sty}}
260   \providecommand{\vpagerefrange}{%
261     \PackageError{\RS@pkgname}%
262     {The vref option used, but varioref.sty not loaded.}%
263     {Load varioref.sty}}
264   }

```

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`.

```

265 \def\RSaddto#1#2{%
266   #2%
267   \@temptokena{#2}%
268   \ifx#1\relax
269     \let#1\@empty
270   \fi
271   \ifx#1\undefined
272     \edef#1{\the\@temptokena}%
273   \else
274     \toks@\expandafter{#1}%
275     \edef#1{\the\toks@\the\@temptokena}%

```

```

276 \fi
277 \@temptokena{\toks@\temptokena}

\DeclareLangOpt Command to declare a language option and add language definitions to the
\extras{language} token for babel.

278 \def\DeclareLangOpt#1#2{%
279   \edef\RS@tempa{\expandafter\@gobble\string#2}%
280   \RS@ifundefined{\RS@tempa}%
281     {\PackageError{\RS@pkgname}{%
282       Unknown definitions \@backslashchar\RS@tempa\MessageBreak
283       for language option ‘#1’}{}}%
284     {\DeclareOption{#1}{%
285       \AtBeginDocument{\expandafter\RSaddto\csname extras#1\endcsname #2}}}%
286   }

```

6.9 Configuration files

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

```

287 \newcommand*\RS@cfgfile{refstyle.cfg}
288 \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`.

```

289 \ifpackagewith{\currname}{noconfig}%
290   {\PackageInfo{\RS@pkgname}{No config file loaded}}%
291   {\InputIfFileExists{\RS@reffile}%
292     {\PackageInfo{\RS@pkgname}{Local config file \RS@reffile\space used}}%
293     {\InputIfFileExists{\RS@cfgfile}%
294       {\PackageInfo{\RS@pkgname}{Config file \RS@cfgfile\space used}}%
295       {\PackageInfo{\RS@pkgname}{No config file found}}}}
296 \DeclareOption{noconfig}{\OptionNotUsed}%

```

Process the options, including options in config file.

```

297 \ProcessOptions*\relax
    The end of this package.
298 \end{package}

```

Change History

v0.0		of <code>\(type)label</code> 17
General: Initial version 1	<code>\RSaddto</code> : Rename <code>\RS@addto</code> to	
v0.1	<code>\RSaddto</code> 20	
General: First stable version 1	v0.3	
v0.2	General: Documentation update . . . 1	
General: First updated version . . . 1	<code>\RS@setkeys</code> : original <code>\setkeys</code>	
Remove redundant <code>\RS@label</code> 19	<code>\RS@setkeys</code> : copy 13	
<code>\RS@buildref</code> : Remove robust def		

v0.4		\efstyle.def	21
General: Add nokeyprefix option	1	\RS@newref: Ad lsttwotxt option	15
Documentation update	1	v0.5	
\ifRS@keyactive: Add nokeyprefix		\@safe@activesfalse: Add	
to turn of the key prefix	15	\@safe@activestruel	13
\ifRSlsttwo: Add \ifRSlsttwo		\@safe@activestruel: Add	
test for list containing only two		\@safe@activesfalse	13
parameter	15	General: Documentation update . .	1
\RS@@@ref: Add \ifRSlsttwo test		\DeclareLangOpt: Add \extras{language}	
for list containing only two pa-		to beginning of document.	21
rameter	17	\RS@ref: Add \@safe@activestruel	
\RS@cfgfile,\RS@reffile: Ad		to avoid active chars is ref list	17
alternative definition file			

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.

Symbols	\KV@prefix	9	\RS@far	203, 208
\@currname	68, 289		\RS@firstcap <u>64</u> , 116, 127	
\@eolst	149,	L	\RS@ifnamedefinable	
152, 164, 166, 180		\label	33, 35, <u>36</u>
\@gobble	279		\RS@ifundefined	
\@ifpackagewith 68, 289		N	<u>22</u> , 31, 38, 110, 280	
\@safe@activesfalse		\newref	<u>77</u> , 113, 115	
.	<u>13</u> , 148		O	
\@safe@activestruel	13, 146	\OptionNotUsed	71, 296	
\@tempc	10		P	
\@temptokena		\pageref	199, <u>233</u> , 237,	
.	267, 272, 275, 277		241, 242, 247, 248	
A			\RS@makenamel	
\AtBeginDocument	251, 285	R	154, 159, 186, <u>216</u>
D		\ref	96	
\DeclareLangOpt	<u>278</u>	\RefstyleFileDate	3	\RS@makeref
\define@key	84, 86, 88–98	\RefstyleFileVersion 4	178, 187, 190, <u>236</u>
I		\reftextfaraway	198, 242, 256	\RS@makevpageref 156,
\ifRS@keyactive	67, 83	\RS@@@ref	<u>143</u> , 164, 166, 180	161, 174, 179, <u>239</u>
\ifRS@capname <u>72</u> , 219, 225		<u>143</u> , 164, 166, 180	\RS@makevpagerefrange
\ifRSlsttwo	<u>72</u> , 168	\RS@@cmd	<u>136</u>
\ifRSnameon	<u>72</u> , 217	\RS@@ref	<u>143</u>
\ifRSplural	<u>72</u> , 218	\RS@buildref	107, <u>126</u>	\RS@namedef
\ifRSstar	<u>72</u> , 217	\RS@cap	66,	84, 86, 89–98, 111
K		120, 122, 131, 133		\RS@namelet
\KV@do	11	\RS@cfgfile	293, 294	<u>15</u> , 30, 31
		\RS@cfgfile,\RS@reffile	\RS@nameusel
		<u>287</u>	<u>15</u> , 128,
		\RS@clearref	81, <u>109</u>	129, 141, 142,
		\RS@cmd	130–134, <u>136</u>	169, 171, 188,
		\RS@false	202, 210	205, 220, 222,
				226, 228, 234, 237
				\RS@newref
				79, <u>80</u>
				\RS@notdefinable
			
				40, 41, 45, <u>46</u>

<code>\RS@pageref</code>	<u>193</u>	<code>\RS@testedrobustnamedef</code>	<u>32</u> , 130–134	<code>\RSstarfalse</code> . . .	72, 138
<code>\RS@rangeref</code>	<u>182</u>	<code>\RS@true</code>	201, 206		<code>\RSstartrue</code>	137
<code>\RS@ref</code>	<u>143</u>	<code>\RSaddto</code>	<u>265</u> , 285			
<code>\RS@reffile</code> . . .	291, 292	<code>\RScapnamefalse</code>		T	
<code>\RS@removedef</code>		<code>\RScapnametrue</code>	74, 130, 132, 134		<code>\the</code>	272, 275
.	<u>29</u> , 117–123	<code>\RSlstwofalse</code>	76, 176		<code>\toks@</code>	274, 275, 277
<code>\RS@robustnamedef</code>	<u>15</u> , 35	<code>\RSlstwotrue</code>	163			
<code>\RS@setbool</code>	<u>58</u> , 88	<code>\RSnameonfalse</code>		U	
<code>\RS@setkeys</code>	<u>8</u>	137, 162, 189		<code>\undefined</code>	271
<code>\RS@tempa</code> . . .	78, 79,	<code>\RSnameontrue</code>	73, 138			
	205, 206, 208,	<code>\RSpluralfalse</code>	75		V	
	210, 279, 280, 282	<code>\RSpluraltrue</code>	158, 185		<code>\vpageref</code>	197, 241, 252
<code>\RS@testednamedef</code> .					<code>\vpagerefrange</code>
.	<u>32</u> , 128, 129				247, 248, 260
					<code>\vref@space</code>	12