The varioref package

Frank Mittelbach

2019/11/05

Abstract

This package defines the commands \vref, \vpageref, \vrefrange, and \vpagerefrange for \LaTeX\2\epsilon. \vref is similar to \ref but adds an additional page reference, like ‘on the facing page’ or ‘on page 27’ whenever the corresponding \label is not on the same page. The command \vpageref is a variation to \pageref with a similar functionality. The \v...range commands take two labels as arguments and produce strings which depend on whether or not these labels fall onto a single page or on different pages. Generated strings are customizable so that these commands are usable with various languages.

Contents

1 Introduction 1
2 Loading the package 2
3 The user interface 2
4 Multi-lingual usage 5
5 Customization 6
   5.1 ... if Babel is used 6
6 Options 7
7 A few warnings 7
8 The documentation driver file 8
9 The implementation 8
   9.1 Options ............ 9
   9.2 Defining the main macros 23
   9.2.1 Supporting ranges 29
   9.2.2 Compatibility with older \LaTeX\ formats ....... 32

1 Introduction

In many cases it is helpful when referring to a figure or table to put both a \ref and a \pageref command into the document especially when there are one or more pages between the reference and the object. Therefore some people use a command like

\newcommand{\fullref}[1]{\ref{#1} on page\pageref{#1}}

*This file has version number v1.6c, last revised 2019/11/05.
which reduces the number of key strokes, necessary to make such a complete reference. But since one never knows where the referenced object finally falls, using such a device may result in a page reference to the current page which is disturbing and therefore should be avoided.

2 Loading the package

The suggested way of loading the package these days is by always using the package option \texttt{nospace}, i.e.,

\texttt{\usepackage[nospace]{varioref}}

See below for an explanation of why this is needed. There are some other options that you may want to use additionally; they are explained on page 7.

3 The user interface

\texttt{\vref} The implementation of \texttt{\vref} below produces only a \texttt{\ref} when reference and \texttt{\label} are on the same page. It will additionally produce one of the strings ‘on the facing page’, ‘on the preceding page’, or ‘on the following page’, if label and reference differ by one and it will produce both \texttt{\ref} and \texttt{\pageref} when the difference is larger. The word ‘facing’ is used when label and reference both fall onto a double spread. However, if a special page numbering scheme is used instead of the usual arabic numbering (e.g., \texttt{\pagenumbering{roman}}) then there will be no distinction between one or many pages off.

\texttt{\vpageref} Sometimes one wants to refer only to page number and again such a reference should normally be suppressed if we are referring to the current page. For this purpose the package defines the \texttt{\vpageref} command. It will produce the same strings as \texttt{\vref} except that it doesn’t start with the \texttt{\ref}. A further difference is that it will produce the string that is saved in \texttt{\reftextcurrent} if label and reference fall onto the same page (and no optional argument is used). By defining \texttt{\reftextcurrent} to produce “on this page” or something similar, we can avoid that

\ldots see the example \texttt{\vpageref{ex:foo}} which shows \ldots

comes out as “\ldots see the example which shows \ldots”, which could be misleading.

A space in front of \texttt{\vpageref} it will be ignored if the command doesn’t produce any text at all.

But in fact \texttt{\vpageref} allows even more control. If has two optional arguments. With the first one, one can specify the text that should be used if label and reference fall on the same page. This is very helpful if both are near to each other, so that they may or may not be separated by a page break. In such a case we usually know (!) whether the reference is before or after the label so that we can say something like

\ldots see the example \texttt{\vpageref[above]{ex:foo}} which shows \ldots

which will then come out as “\ldots see the example above which shows \ldots” if we are still on the same page, but as “\ldots see the example on the page before which shows \ldots” (or something similar depending on the settings of the
\textbf{\texttt{\textbackslash refext..before} commands) when there was a page break in the meantime. One warning however, if you use \texttt{\textbackslash vpageref} with the optional argument to refer to a figure or table, keep in mind that depending on the float placement parameters the float may show up on top of the current page and therefore before the reference even if it came after it in the source file.

But maybe you prefer to say “... see the above example” if example and reference fall onto the same page, i.e., reverse the word order. In fact, in some languages the word order automatically changes in that case. To allow for this variation the second optional argument can be used. It specifies the text preceding the generated reference if object and reference do not fall onto the same page. Thus one would write

\begin{verbatim}
... see the \texttt{\textbackslash vpageref[above example][example]\texttt{\{ex:foo\}} which shows ...
\end{verbatim}

to achieve the desired effect.

\texttt{\textbackslash vrefrange}

This command is similar to \texttt{\textbackslash vref} but it takes two mandatory arguments denoting a range to refer to (e.g., a sequence of figures or a sequence of equations, etc.). So if \texttt{\texttt{fig:a}} is your first figure in the sequence and \texttt{\texttt{fig:c}} your last you can write

\begin{verbatim}
... see figures \texttt{\textbackslash vrefrange\{fig:a\}\{fig:c\}} ...
\end{verbatim}

which would then be formatted as

\begin{verbatim}
... see figures 3.4 to 3.6 on pages 23–24 ...
\end{verbatim}

or, if they happen to all fall onto the next page, as

\begin{verbatim}
... see figures 3.4 to 3.6 on the following page ...
\end{verbatim}

i.e., the command is deciding what to say depending on where the two labels are placed in relation to each other; it is essentially implemented using \texttt{\textbackslash vpagerefrange} described below. The optional argument the command may take is the text to use in case both labels are placed on the current page.

\texttt{\textbackslash vpagerefrange}

This command is similar to \texttt{\textbackslash vpageref} but takes two mandatory arguments which are two labels denoting a range. If both labels fall onto the same page, the command acts exactly like \texttt{\textbackslash vpageref} (with a single label), otherwise it produces something like “on pages 15–18” (see customization possibilities below). The optional argument it may take is the text to use in case both labels are placed on the current page.

\texttt{\textbackslash vrefpagenum}

This macro is provided to allow the user to write their own small commands which implement functions similar to those provided by the two previous commands. It takes two arguments: the second is a label (i.e., as used in \texttt{\textbackslash label} or \texttt{\textbackslash ref}) and the first is an arbitrary command name (make sure you use our own) that receives the page number related to this label. So if you have two (or more) labels you could retrieve their page numbers, compare them and then decide what to print. For example, the following not very serious definition (also using the \texttt{\textbackslash ifthen} package)

\begin{verbatim}
\newcommand\amusingversion[2]{the definition%
 \texttt{\textbackslash vrefpagenum}\texttt{\{firstnum\}\{#1\}%
 \texttt{\textbackslash vrefpagenum}\texttt{\{secondnum\}\{#2\}%
 \texttt{\textbackslash ifthenelse}\{\texttt{equal}\texttt{\{firstnum\}\{secondnum\}}%
\end{verbatim}

3
...\amusingversion{foo}{bar}

will print something like

...the definitions of 3 and 4 on the previous page

in the case both labels are on the same page but something like

...the definition of 3 on the next page and of 4 on page 13

in case they are on different pages.

For this kind of application the package also provides \vpagerefcompare

\newcommand\amusingversion[2]{
  \vpagerefcompare{#1}{#2}
  \{s of \ref{#1} and \ref{#2} \vpageref{#1}\}
  \{ of \ref{#1} \vpageref{#1} and of \ref{#2} \vpageref{#2}\}
}

and you don’t have to load the ifthen package any longer to make it work.

Another command for conditional processing is \vpagerefnearby. It takes

\newcommand\vpagerefnearby[3]{
  \vpagerefcompare{\ref{#1}}{#2}
  \{s of \ref{#1} and \ref{#2} \vpageref{#1}\}
  \{ of \ref{#1} \vpageref{#1} and of \ref{#2} \vpageref{#2}\}
}

If you don’t use the nospace option the user commands \vref, \vpageref,

\vpagerefrange all work by first removing any space on their left and then

inserting some space of their own (\vref, for example, a nonbreakable space).

That seemed like a good idea back then, but it has the disadvantage that you
can’t use these macros in situations where you definitely do not want any space
before the generated text. E.g., in situations like (\vref{foo} ...) you end up

with a space after the open parenthesis.

Since it was too late to change the default in 2001 I added star versions of the
macros which do not add any space before the generated text (they do nevertheless
remove space at the left).

However, that too has problems (one is that hyperref introduced the star
forms as a means to generating references without hyperlinks (which is clearly
the more important application). So these days I recommend to always call the
package with the option nospace which prevents varioref from messing with the
space in front. Of course you are then responsible to always add it when necessary,
but that does make the source more readable so it is a good thing. This also has

\footnote{There is also a \vpagerefcomparenearby but this is only meant for package writers, so it is
only documented in the implementation section.}
the advantage that the star forms are now unnecessary again and can be used in
the way implemented by hyperref if both packages are used together.

In fact it would be much nicer to make that the default but the package has
been used by many people for so many years that changing the default would
break way to many older documents.

A reference via \ref produces by default the data associated with the corre-
spanding \label command (typically a number); any additional formatting has
to be provided by the user. With \labelformat it is possible to declare such
formatting for each “reference” counter, e.g.,

\labelformat{section}{section#1}
\labelformat{equation}{equation(#1)}

A side effect of using \labelformat is that, depending on the defined format-
ting, it becomes impossible to use \ref at the beginning of a sentence (if its re-
placement text starts with a lowercase letter). For that reason varioref provided
the commands \Ref and \Vref. They behave like \ref and \vref except that
they uppercase the first token of the generated string. These days \labelformat
and \Ref are already provided by the \LaTeX{} kernel; only \Vref is still defined
within the varioref package.

To make \Ref or \Vref work properly the very first token in the second
argument of the \labelformat has to be a simple ASCII or UTF-8 letter, oth-
wise the capitalization will fail or worse, you will end up with some er-
ror messages. If you actually need something more complicated in this place
(e.g., an accented letter not written as a UTF-8 character) you have to ex-
plicitly surround it with braces, to identify the part that needs to be cap-
italized. For example, for figure references in the Hungarian language you
might want to write \labelformat{figure}{\'{a}bra~\thefigure} or use
\labelformat{figure}{ab{\`a}ra~\thefigure} which avoids the brace problem.

If you like to have \vref suppress the page number on pages where label and
reference fall onto the same page, but prefer reference to page numbers otherwise
then \thepagerefnum can be used. This macro hold the current page “number”
when \pageref and friends are executed. Thus, by defining, for example

\renewcommand\reftextfaceafter {on page~\thepagerefnum}
\renewcommand\reftextfacebefore {on page~\thepagerefnum}
\renewcommand\reftextafter {on page~\thepagerefnum}
\renewcommand\reftextbefore {on page~\thepagerefnum}

textual references can be suppressed.

4 Multi-lingual usage

The package works well together with the babel system if it is loaded after the
babel package, e.g., a document in English (default), French and German text
could load the packages as follows:

\usepackage[ngerman,french,english]{babel,varioref}

(or in two separate lines if you prefer one \usepackage line per package). Depending
on the current babel language \vref and friends would then generate strings
in different languages. If the language options are given as global options to the
document class then it is enough to use

\usepackage{babel,varioref}

to achieve the same effect.

If more than one language is specified with \texttt{varioref} then it will start using
the last language selected. Switching of languages is managed through Babel;
evertheless all languages of interest need to be specified when \texttt{varioref} is loaded
so that it can integrate their strings into the Babel system.

## 5 Customization

The package supports all options defined by the \texttt{babel} package to translate the
fixed strings into other languages than English. (Some languages need updating,
however—help is appreciated.) It also supports languages currently not in \texttt{babel}
(those can only be used standalone, obviously); check the section on options
later on.

You can also modify some or all of the strings by redefining the fol-
nowing commands after the package has been loaded. Backward references
use \texttt{\reftextbefore} if the label is on the preceding page but invisible and
\texttt{\reftextfacebefore} if it is one the facing page (i.e., if the current page number
is odd). Similarly \texttt{\reftextafter} is used when the label comes on the next page
but one has to turn the page and \texttt{\reftextfaceafter} if it is on the following but
facing page.

In fact, \texttt{\reftextface}... is used only if the user or the document class spec-
ified two-sided printing.

\texttt{\reftextfaraway} \hspace{1em}
Finally we have \texttt{\reftextfaraway} which is used whenever label and reference
differ by more than one or when they aren’t numeric. This macro is a bit different
because it takes one argument, the symbolic reference string so that one cane make
use of \texttt{\pageref} in its replacement text.

\texttt{\reftextvario} \hspace{1em}
To allow a bit random variation in the generated strings one can use the com-
mand \texttt{\reftextvario} inside the string macros. It takes two arguments and selects
one or the other for printing depending on the number of already seens \texttt{\vref}
or \texttt{\vpageref} commands. As an example see the definitions of \texttt{\reftextbefore} etc.
on page 9.

The commands \texttt{\refrange} and \texttt{\vpagerefrange} produce their text using
two macros described below. By redefining them one can modify the results to
accommodate special requirements.

They both take two mandatory arguments denoting the first and the last label
of the range.

\texttt{\reftextpagerange} \hspace{1em}
This macro produces text that describes the page range of the two labels, e.g.,
the default for English is “on pages \texttt{\pageref{#1}--\pageref{#2}}”.

\texttt{\reftextlabelrange} \hspace{1em}
This macro produces text that describes the range of figures, tables, or whatever
the labels refer to, the default for English is “\texttt{\ref{#1} to \ref{#2}}”.

### 5.1 ... if Babel is used

\textit{If babel is being used then the above commands need to be changed using the Babel
mechanism, e.g.,}
Otherwise babel will reintroduce its own defaults whenever the language is changed (which happens in particular during \begin{document} where the main language is selected).

6 Options

As mentioned above the package supports all standard options offered by the Babel system to customize the strings produced, it may also have a few not supported by babel, for a full list look in section 9.1 on page 9. In addition it offers the option draft to turn error messages into warnings during development. The default final produces error message when a generated string falls onto a page boundary (see next section).

Newly introduced in 2019 are the options space (default) and nospace. While the former is the default to allow documents written in the last twenty years to continue to work correctly, I nowadays, suggest that you always use the package with the option nospace. This will stop varioref from meddling with the spaces preceding commands and thus make the star forms \vref* and \vpageref* unnecessary.

This means that the star forms can then be used as defined by hyperref for suppressing a hyperlink when both packages are used together (which is a rather typical scenario these days).

7 A few warnings

Defining commands like the ones described above poses some interesting problems. Suppose, for example, that a generated text like ‘on the next page’ gets broken across pages. If this happens it is very difficult to find an acceptable solution and in fact can even result in a document that will always change from one state to another (i.e., inserting one string, finding that this is wrong, inserting another string on the next run which makes the first string correct again, inserting . . . ). The current implementation of varioref therefore issues an error message whenever the generated text is broken across page boundaries, e.g.,

\vrefwarning\rref\vrefwarning

would would result in an error, which needs to be resolved by the user by replacing the \rref command with an ordinary \ref just before the final run. This is not completely satisfactory but in such case no solution really is. During document preparation, while one is still changing the text, such error messages can be turned into warnings by placing a \vrefwarning command in the preamble. This is equivalent to specifying “draft” as an option to the package. \vrefshowerrors ensures that varioref stops when detecting a possible loop. This is the default and equivalent to specifying “final” as an option.

At the end final a warning: every use of \rref will internally generate two macro names to keep track of the string positions within the document. As a
result you may run out of name space or main memory\textsuperscript{2} if you make heavy use of this macro on a small \TeX{} installation. For this reason the primitive command \fullref{} is also provided. This command can be used whenever you know for sure that label and reference can’t fall onto nearby pages.

8 The documentation driver file

The next bit of code contains the documentation driver file for \TeX{}, i.e., the file that will produce the documentation you are currently reading. It will be extracted from this file by the \texttt{docstrip} program.

\begin{verbatim}
\documentclass{ltxdoc}
\usepackage{multicol,varioref}
\GetFileInfo{varioref.sty}
\setlength\hfuzz{1pt} % ignore slight overfulls
\EnableCrossrefs
%\DisableCrossrefs % Say \DisableCrossrefs if index is ready
\RecordChanges % Gather update information
%\OnlyDescription % comment out for implementation details
\begin{document}
\DocInput{varioref.dtx}
\end{document}
\end{verbatim}

9 The implementation

The main implementation idea is to generate an internal label command for every \vpageref{} and compare the page reference of this label with the page reference of the user-requested label. Since this means one additional macro name for each use of \vpageref{} or \vref{} (which is implemented in terms of \vpageref{}) the memory requirements of this package may be high in terms of main and macro name memory. Since the information held in the internal labels is used only once one could envision a different scheme if suitable primitives for this kind of “one place information” would be provided by \LaTeX{}. This type of data structure is planned for \LaTeX{}X but for the moment we have to live with the memory restrictions.

We start by checking if this file was already loaded. If not we identify the current version. This is actually done at the top of the file, so we comment it out here

\begin{verbatim}
\NeedsTeXFormat{LaTeX2e}[1994/06/01]
\ProvidesPackage{varioref}[iledate space \fileversion space]
\% \LaTeX{}X2e package for extended references (FMi)]
\end{verbatim}

\vref@addto To support the use of babel we want to add the additional strings to the \texttt{\language} commands. Since \texttt{\addto} in the current implementation of

\textsuperscript{2}Not really that likely these days, but one never knows . . . . In any case \TeX{} will process your document a tiny bit faster.
babel has a bug that does not allow to use arguments containing hashmarks we do this by hand.

\def\vref@addto#1#2{% #2% \@temptokena{#2}% \iffalse\undefined \edef#1{\the\@temptokena}% \else \toks0\expandafter{#1}% \edef#1{\the\toks0\the\@temptokena}% \fi \@temptokena{}\toks0\@temptokena %}
\@onlypreamble\vref@addto

\textbf{9.1 Options}

\texttt{\vref@excuse} \textbf{Excuse that we don’t know the strings for a certain language.}

\def\vref@excuse#1{% \PackageWarningNoLine{varioref}{Sorry, language ‘#1’ not supported\MessageBreak English strings used by default. \MessageBreak Please modify \protect\reftext... commands manually.\MessageBreak If you can suggest translations for this language, please mail them to the author of this package}}
\@onlypreamble\vref@excuse

\texttt{\vref@stringwarning} \textbf{And we sometimes just don’t know some strings ...}

\def\vref@stringwarning#1{% \PackageWarningNoLine{varioref}{Sorry, there is no proper translation for the string\MessageBreak English string used instead.\MessageBreak If you can suggest translations for the current\MessageBreak language, please mail them to the author of this\MessageBreak package}}

\texttt{\reftextafter} \textbf{The options do set the macros that generate the textual strings. Note, that they do not start with a space, the space is already added in the main macro below.}

\texttt{\reftextbefore} \texttt{\reftextafter} \texttt{\reftextbefore} \texttt{\reftextcurrent} Afrikaans option contributed by Danie Els (dnjels@sun.ac.za).

\DeclareOption{afrikaans}{%
\vref@addto\extrasafrikaans{%
\def\reftextfaceafter {op die \reftextvario{regterbladsy}%
{bladsy langsanaan}}}%
Arabic option text was contributed by Ihab Samir Mattar (dr_kremlin@hotmail.com). As our standard documentation workflow uses pdfTeX the Arabic letters can’t be displayed easily, so you will see missing glyphs below. To see the real letters you have to look at the source or package file.

The austrian defaults are the same but needed for “Neue Rechtschreibung”.

The austriaen defaults are the same as the german ones.
Text for basque defaults was contributed by Íñaki Larrañaga Murgoitio (aka dooteo) (<dooteo@euskalgnu.org>).

Text for the Bahasa Malaysia defaults was contributed by Bob Margolis 2005-12-02.

Notes: pada muka surat translates as “on the page” though BM has no definite (or indefinite) article; adjectives and other modifiers follow the noun (clause); bertentangan is “opposite” in the sense of “facing”; berikutnya is “next” (in this context - actually next-to-this, literally); berikutnya also translates as “following this” sebelumnya is “preceding”; also “previous” “ini” translates, literally, as “this” but is also used in the sense of current page.

Text for brazil defaults was contributed by Alcino Dall Igna Junior (adij@di.ufpe.br).
Text for Breton defaults was contributed by Christian ROLLAND (Christian.Rolland@univ-brest.fr).

Text for Bulgarian defaults was contributed by Georgi Boshnakov (georgi.boshnakov@gmail.com).
Text for Catalan defaults was contributed by Robert Fuster (rfuster@mat.upv.es).

Text for Croatian defaults was contributed by Šime Ungar (ungar@math.hr).

Defaults for the Czech option provided by Milan Vancura <milan@suse.cz>

Defaults for Danish provided by Torsten Martinsen (tmar91@kom.au.dk). Further defaults by Ole Laursen (olau@hardworking.dk).
Default string for dutch have been contributed by Frank Poppe (POPPE@SWOV.NL). This option currently supports one additional string macro \refpagename so that you can easily change to bladzijde instead of pagina if you prefer this word for “page”. However, I will not guarantee that this will survive future versions of this package, so use it on your own risk (you can always update the full strings to be on the safe side).

Concerning the next defaults I received the following comment by Frederik Fouvy <fouvy@coli.uni-sb.de>: “In the Dutch options (where I can exert a proper language judgement ;-) , ‘t/m’ is not used in formal (scientific) texts. I would either use ‘–’ or ‘tot en met’ (the abbreviation written out). ”

The esperanto strings were suggested by Albert Reiner (areiner@tph.tuwien.ac.at).
The finnish strings were suggested by Matti Rintala (bitti@cs.tut.fi) and Hillevi Gavel (Hillevi.Gavel@mdh.se).

French defaults are provided by Daniel Flippo (Daniel.Flipo@univ-lille1.fr).

Galician defaults are provided by Matthias Moebius (Matthias.Moebius@uni-konstanz.de).
There are no good variants for German (I think and still think but this is a matter of taste :-). The following definitions were recently suggested to me but since the original are in for a long time i don’t want to change them now since that could make a lot of documents change their formatting. If you fancy them, add a redefinition of the corresponding macro(s) to the preamble of your document.

\def\reftextfaceafter {auf der \reftextvario{gegen"uberliegenden}{anderen} Seite} \\
\def\reftextfacebefore {auf der \reftextvario{gegen"uberliegenden}{anderen} Seite} \\
\def\reftextafter {auf der \reftextvario{n"achsten}{folgenden} Seite} \\
\def\reftextbefore {auf der \reftextvario{vorigen}{vorhergehenden} Seite} \\
\def\reftextcurrent {\reftextvario{auf dieser}{diese} Seite} \\
\def\reftextfaraway#1{auf Seite~\pageref{#1}} \\
\def\reftextpagerange#1#2{auf den Seiten~\pageref{#1}--\pageref{#2}} \\
\def\reftextlabelrange#1#2{\ref{#1} bis~\ref{#2}}

The \texttt{ngerman} defaults are the same but needed for “Neue Rechtschreibung” (i.e., a different set of hyphenation patterns).

\DeclareOption{ngerman}{% \\
{\ref@addto\extrasngerman{\% \\
\def\reftextfaceafter {auf der \reftextvario{n"achsten Seite}} \\
\def\reftextfacebefore{auf der vorherigen Seite} \\
\let\reftextafter \reftextfaceafter \\
\let\reftextbefore \reftextfacebefore \\
\def\reftextcurrent {auf dieser Seite} \\
\def\reftextfaraway#1{auf Seite~\pageref{#1}} \\
\def\reftextpagerange#1#2{auf den Seiten~\pageref{#1}--\pageref{#2}} \\
\def\reftextlabelrange#1#2{\ref{#1} bis~\ref{#2}}} 

Defaults for greek suggested by Apostolos Syropoulos (apostolo@obelix.ee.duth.gr).

\DeclareOption{greek}{% \\
{\ref@addto\extrasgreek{\% \\
\def\reftextfaceafter {sthn \reftextvario{paro'usa}{ep'omenh} sel'ida} \\
\def\reftextfacebefore{sthn \reftextvario{paro'usa}{prohgo'umenh} sel'ida} \\
\def\reftextafter {sthn ep'omenh sel'ida} \\
\def\reftextcurrent {\reftextvario{auf dieser}{diese} Seite}}} 

16
Defaults for Italian suggested by Giovanni Pensa (pensa@dsi.unimi.it) with iTEXnici.

Defaults for Hungarian by Jeff Goldberg (jeffrey@goldmark.org). There is a problem with the use of the definite article a or az in Hungarian before expansions of \vref or \ref. The problem is that the word should be az if the number following would begin with a vowel if spelled out, and a if the number would begin with a consonant.

The option assumes that there is a command \aza defined which helps resolving this problem. \footnote{This information is actually quite old but unfortunately never made it into varioref. In case any work has been undertaken to resolve this please inform the author about it.}

Defaults for Hungarian by Jeff Goldberg (jeffrey@goldmark.org). There is a problem with the use of the definite article a or az in Hungarian before expansions of \vref or \ref. The problem is that the word should be az if the number following would begin with a vowel if spelled out, and a if the number would begin with a consonant.

The option assumes that there is a command \aza defined which helps resolving this problem. \footnote{This information is actually quite old but unfortunately never made it into varioref. In case any work has been undertaken to resolve this please inform the author about it.}
The following suggested patch has not been added yet (pending changes in babel system).

\def\reftextfaraway#1{a~\csname @@magyar@az@lowxu\endcsname\pageref{#1}.~ol-
\def\reftextpagerange#1#2{a~\csname @@magyar@az@lowxu\endcsname\pageref{#1}--\pageref{#2}.~ol-
\def\reftextlabelrange#1#2{{}\csname @@magyar@told\endcsname\ref{#1}+tol\ref{#2}-ig}%
\def\reftextfaraway#1{\aza{\pageref{#1}} \pageref{#1}.~ol-
\def\reftextpagerange#1#2{\vref@stringwarning\reftextpagerange\ref{#1}--\ref{#2}}%
\def\reftextlabelrange#1#2{\vref@stringwarning\reflabelpagerange\ref{#1} to\ref{#2}}%
\}\%

The \AtBeginDocument needs to be executed when the option is processed and not inside the babel interface.

\AtBeginDocument\providecommand\aza[1]{a/az%
\PackageError{varioref}{% Proper definition for command \noexpand\aza missing}%\{You need to (re)define the command \noexpand\aza that takes one argument (a number or a \noexpand\pageref command) and produces the string ‘a’ or ‘az’ depending on the argument value. It should also accept an optional argument (default ‘1’) to denote whether or not the string should be lowercase or uppercase.\}}%
}

The Norwegian and ‘nynorsk’ translations have been provided by Dag F Langmyhr <dag@ifi.uio.no> and Karl Ove Hufthammer <huftis@bigfoot.com>.

\DeclareOption{norsk}{
\DeclareOption(nynorsk)}
The polish translations have been provided by Tomasz Michniewski <tomekm@ias.wat.waw.pl>. The use of the command \eob and aob etc. requires appropriate definitions as provided, for example, by the babel system. This should probably be handled differently but it would require modifications in babel’s language support.

Updates to the default text for Romanian have been suggested by Florin Oprina (<florin.oprina@gmail.com>).
The text for Russian was taken from the book in Russian by I. A. Kotelnikov and P. Z. Chebotaev, and adapted to the encoding-independent form by Vladimir Volovich (vvv@vsu.ru).

The text for Slovak defaults was contributed by Miroslav Sedivy (<miro@slovensko.com>).

The text for Slovene defaults was contributed by Roman Maurer (<roman.maurer@amis.net>).
The text for Spanish defaults was contributed by Julio Sanchez (jsanchez@gmv.es) and Fernando Dobladez (<ferd@fuego.com>).

The text for Swedish was contributed by Mats Dahlgren (matsd@sssk.se).

The text for Ukrainian defaults was contributed by Mykola Lyakhovych.
We support francais as an alternative to french since people might be used to it, but we discourage it.

During document preparation errors due to \vref are usually of not much interest. For this reason the draft option turns them into warnings by issuing a \vrefwarning command. The final option (default) stops with an error message when varioref detects a possible loop.

Whether or not varioref uses its own white space and removes the one in the source document is handled by the following two options. For historical reasons the default is to handle the space, but the recommendation these days is to use nospace.

After all options are declared we execute the default option english to set up the defaults and then call \ProcessOptions* to handle any option passed to the package. The star form is used since we want to execute the options in the order they are given (in case this isn’t used together with the babel package).
9.2 Defining the main macros

\vr@cnt To generate new labels we use a counter and construct the internal label names by prefixing the current counter value with the string vr@. We need a \LaTeX counter to get \includes handled correctly.

603 \newcounter{vrcnt}

\vpageref The command \vpageref generates the appropriate string by first parsing the optional arguments (if any) and then compares the internal and the user label. This command should be robust since the user might put it anywhere.

We now start by scanning for a star and then call either \vp@gerefstar or \vp@geref. The space handling logic is a bit twisted these days: if the option space is in force then preceding spaces in the document are removed and the star means that the command should not add its own space. However, with option nospace any space already in the source will be left alone and the command doesn’t add its own space either. Thus in this case the star form doesn’t have any effect and below \vp@gerefstar is let to \vp@geref. This allows a package like hyperref to redefine it and use it for its own tasks (like denoting that no link should be produced).

604 \DeclareRobustCommand\vpageref{\@ifstar
605 {\let\vref@maybe@space\@empty\vp@gerefstar}%
606 {\let\vref@maybe@space\space \vp@geref}}%
607 \def\vp@geref{\@ifnextchar[%
608 \@vpageref{\@vpageref\[\reftextcurrent\]}%
609 }

\vref@space \vref@maybe@space A default for \vref@space. This isn’t really needed except in the case that somebody has hooked into the \texttt{varioref} interfaces at a lower level (which isn’t really supported, but...). So giving a default will prevent an error message in that case (and hopefully produce the same behavior as before).

610 \let\vref@space\space
611 \let\vref@maybe@space\space

\vpageref More parsing...

612 \def\vpageref[#1]{\@ifnextchar[%
613 {\@@vpageref{#1}}{\@@vpageref{#1}\[]}}

\@@vpageref The \@@vpageref macro finally generates the references by comparing the value of an internal label with the value of the user label.

614 \def\@@vpageref#1[#2]#3{%
615 First it switches to horizontal mode if necessary. If there was a preceding space we remember its size in \texttt{@tempskipa} and then remove it for reinsertion later.
616 \leavevmode
617 \@@tempskipa@lastskip \unskip
618 Then it advances the counter \texttt{c@vrcnt} by one which is used to generate internal label names.
619 \global\advance@c@vrcnt\@ne
Then it checks if for the current value of \texttt{\the\c@vrcnt} a label command was issued in the last run. If not it pretends that there was one with the value \texttt{??}. Thereafter it stores the page ref value for this label in the macro \texttt{\@tempa}.

Here we check that the generated text is not going across a page boundary.

In version 1.0l the label string has been turned around so that the number comes first. This was done to allow easy explicit expansion of the number before it is passed to the \texttt{\label} command. In the babel system the argument of \texttt{\label} was not expanded with the result that wrong label strings have been generated. This is a general problem that might need a completely different solution in babel but for now the change below (and in some other places in the code) should solve the problem for this special combination of packages.

These days we actually use \texttt{\vref@label} instead of \texttt{\label} as this avoids problems with Babel or \texttt{amsmath} redefining the latter command.

Since we have removed any leading space above we now insert a space here after the label. It is done here so that it is still visible from following code—the space before the label would be hidden. What kind of space depends on the options to the package. If \texttt{space} was specified, we insert our own space, namely \texttt{\vref@maybe@space}. This will typeset a space unless the star form of the current command was used.

But if \texttt{nospace} is in force we simply use the same space value that was used in the source document (unless that was zero or non-existent.

Inside displays of the \texttt{amsmath} package the \texttt{\label} command is redefined which makes the test for loops incorrect if a \texttt{\vref} is used in, say, \texttt{\intertext}. So we test this condition first and only do the test if \texttt{\label} doesn’t have a special meaning. [This is no longer necessary as \texttt{\label} isn’t used any more]

If both points do not fall onto the same page with either issue an error or a warning message. However, we do no longer issue that warning now but wait until we are sure that both labels aren’t “far away” because in that case it doesn’t matter. So we just save that info in a macro now.
Same game for the user requested label; this time the page reference is saved in `\thevpagerefnum`.

Now after the internal label has served its purpose if would be nice to free the memory it occupies by using something like

```
\global\expandafter\let\csname r@\the\c@vrcnt @vr\endcsname\@empty
```

But this is not possible because it would result in getting ‘Labels may have changed...’ warnings for every run. Now we are ready to produce the textual strings.

```
\ifx\@tempa\thevpagerefnum
If they are the same we insert the first argument into the input stream. Otherwise we insert the second argument. Recall that those are the two optional arguments of \... However, if the user explicitly requested an empty text string using the optional argument we cancel any preceding space since in this case we are otherwise likely to end up with spaces one before and one after `\vpageref`.
```

```
\def\@tempc{#1}\
\ifx\@tempc\@empty
\unskip
\else
#1%
\fi
\else
If \#2 is empty we do nothing otherwise we insert it followed by a space.
```

```
\def\@tempc{#2}\
\ifx\@tempc\@empty \else
#2\vref@space
\fi
```

Now we check if the page number of the referenced object (stored in `\thevpagerefnum`) is a single positive number.

```
\is@pos@number\thevpagerefnum
{%
If so, we check if the current position (stored in `@tempa`) is a positive number. If this is the case we assign this number to the counter register `@tempc@nta` and add one to it.
```

```
\is@pos@number@tempa
{\@tempc@nta@tempa
\advance\@tempc@nta@one
}%
```

If it is not a positive number we assign the largest possible number to `@tempc@nta` and thereby pretending that label and reference are miles away from each other.

```
{\@tempc@nta@maxdimen}%
```
Now we are ready to check if reference and object are on nearby pages. \thevpagerefnum will expand to the page number of the object (and we know that this is a number) and \@tempcnta is either one higher than the reference page or completely out of bounds. So if both represent the same value then the object lies one page after its reference.

\ifnum \thevpagerefnum = \@tempcnta
Thus if the object falls onto an odd page then the reference is on the facing even page (and so we insert \reftextfaceafter), otherwise the object cannot be seen from the reference (and we in insert \reftextafter). Don’t be surprised if we are not using \thevpagerefnum in the check. Since \@tempcnta has the same value it is faster to use the register instead of parsing the macro contents anew.

\ifodd\@tempcnta
In fact we are going to use \reftextfaceafter only if we are doing two-sided printing, otherwise \reftextafter is always used. Since the value of \if@twoside is evaluated before reading in packages we could do better (saving some tokens) by defining the current macro in dependence of this boolean.

\if@twoside
\reftextfaceafter
\else
\reftextafter
\fi
\else
\reftextafter
\fi

If the object is not on the page following the reference we check if it is on the page before the reference. In \LaTeX{} this situation is not too common, for example with floats it normally does not occur, but of course it isn’t impossible if you more than one reference to the same object, or if you have back references to sections, theo-

\advance\@tempcnta - 2

If now \thevpagerefnum and \@tempcnta have the same value then the object lies one page before the reference.

\ifnum \thevpagerefnum = \@tempcnta
Again we have to check for odd or even pages to distinguish between the facing and the non-facing situation.

\ifodd\@tempcnta
\reftextbefore
\else
\reftextbefore
\fi
\else
Again the \reftextfacebefore only if @twoside evaluates to true.

\if@twoside
\reftextfacebefore
\else
\reftextbefore
\fi
\fi

26
If the above test also returns false then we have object and references on pages which are far away or don’t contain simple page numbers. Therefore we generate the \reftextfaraway string and drop any possible “looping warning” in \vref@may@loop. Recall that this is a macro which has the user label as an argument.

\begin{verbatim}
682 \else
683  \let\vref@may@loop\relax % added
684  \reftextfaraway{#3}\
685 \fi
686 \fi
687 \}%

We do the same if our first test (that the page with the object has a positive page number) turns out to be false.

\begin{verbatim}
688 {%
689  \let\vref@may@loop\relax % added
690  \reftextfaraway{#3}}%
691 \fi
\end{verbatim}

Finally we generate the internal label so that it can be checked on the next run. This means that we compare the position after the string with the position of the referenced object. There is one thing to note: to conserve space we locally make \@currentlabel empty since we are only interested in the page number value of this internal label.

These days we actually use \vref@label instead of \label as this avoids problems with Babel or amsmath redefining the latter command.

\begin{verbatim}
692  \vref@label{the\c@vrcnt @vr}%
\end{verbatim}

Finally we run \vref@may@loop which is either \relax or generates a “may loop” warning or error as appropriate.

\begin{verbatim}
693  \vref@may@loop
694 }
\end{verbatim}

\vp@gerefstar The default code for the star form is identical to the previous macro \vp@geref. It only produces a different result because before calling it we set \vref@maybe@space to do nothing.

The hyperref package can then change the meaning of this command to suppress a hyperlink.

It is also a good way to check if we are running with a new varioref by testing

\begin{verbatim}
695 \ifdef\vp@gerefstar
696  ... code for new varioref
697 \else
698  ... code for old varioref
699 \fi
\end{verbatim}

\vref@label This macro is like \label in the kernel but without passing on \@currentlabel, ie useful if all we are interested in is the page number. It also has the effect that it is unaffected by amsmath or Babel redefining the \label command.

\begin{verbatim}
696  \def\vref@label\if\@bsphack
697    \protected@write\@auxout{%}
\end{verbatim}
This macro allows to define \reftextafter and friends to refer to the page number rather than producing a textual reference.
\let\thevpagerefnum@empty

When the \vref command detects a possible crossing over a page boundary it will call \vref@err to generate an error message. During document preparation one can turn such errors into warnings by issuing a \vrefwarning declaration.
\def\vrefwarning{\def\vref@err{\PackageWarning{varioref}}}
\def\vrefshowerrors{\def\vref@err##1{\PackageError{varioref}{##1} %
{Please check the pages in question. You might need to replace \vref or \vref{#1} by a normal \page{#1}}}}

We start by scanning for a star form and then continue in a different macro depending on the result.
\DeclareRobustCommand\vref{\@ifstar{\vref@star}{\vr@f}}
\vref@star The star form is easy (and just a convenient shortcut), just call \ref followed by \vpageref and pass the optional argument to the latter. We don’t support the second optional argument of \vpageref, it that is needed one has to call both commands explicitly
\newcommand\vref@star[2][]{%
\ref{#2}
\vpageref[#1]{#2}}
\vr@f The normal case is a little bit more elaborated ...
\def\vr@f{%
\leavevmode
If option space is used then we remove a preceding space and add our own unbreakable space instead. Otherwise we don’t do any space-handling.
\if@vrefhandlespace
\unskip \nobreakspace
\fi
The use of \nobreakspace or ~ after the \unskip means that this command will always produce a normal space while something like \nobreakspace will react to settings of \nonfrenchspacing. Unfortunately the latter will also act in cases where one really wants a normal space, e.g., in Fig.\vref{..}, resulting in a large extra space after the dot. For this reason the first solution is used.
After having handled (or not handled) the spacing we call \vref@star to scan the optional arguments and do the rest. If that command has been altered by hyperref then the current one needs altering too and does the argument scanning explicitly.
\vref@star % in hyperref this needs to be explicit
The utility macro \is@pos@number takes three arguments: a string that is tested for being a valid non-negative integer and the actions to be taken in case the test comes out true or false.

We start by passing the string to the macro \is@pos@num after prefixing it with a 0 and adding a space after it. To have a well-defined ending point we also add \@nil at the end.

\begin{verbatim}
def\is@pos@number#1{\is@pos@num0#1\space\@nil}
\end{verbatim}

Now we parse the enlarged string into a counter register. To get control back after everything that is regarded by \TeX as a number is put into that register we assign \is@pos@num@ to the \afterassignment token. Since the string started with 0, we can be sure that the register assignment will be carried out without an error message. In case of counter assignments leading zeros are discarded.

\begin{verbatim}
def\is@pos@num{\afterassignment\is@pos@num@ \count@}
\end{verbatim}

Now we have to check whether or not the whole string was parsed into that register or some remainder was left over. Since we have added a \@nil token at the very end we can use that to delimit the argument of \is@pos@num@. Note, that the added space in \is@pos@number above gets parsed away by the counter assignment. If it would be missing, and the full string would consist of a number, \TeX would try to replace \@nil by its definition to see if it would contain additional digits and thus we would be in trouble at this point.

\begin{verbatim}
def\is@pos@num@#1\@nil{\ifx\@nil#1\@nil\else\expandafter\@firstoftwo \else \expandafter\@secondoftwo \fi}
\end{verbatim}

This macro gives a little bit of random variation in the text because the outcome depends on the number of \vref commands seen before.

\begin{verbatim}
def\reftextvario#1#2{\ifodd\c@vrcnt #1\else#2\fi}
\end{verbatim}

And here is the primitive command that always produces a \ref and a \pageref.

\begin{verbatim}
def\fullref#1{\ref{#1} \reftextfaraway{#1}}
\end{verbatim}

### 9.2.1 Supporting ranges

This command takes a label as \#1 extracts the page number associated with it and saves the result in a csname which is passed as \#1. If the label isn’t associated with a page number it essentially associates it with ?? for both the label number and the page, thus defining the label this way. This means that no label warning is generated for this label in case it is undefined.

\begin{verbatim}
def\vref@pagenum#1#2{%
@ifundefined{r@#2}{\@namedef{r@#2}{??}{??}}{}%\end{verbatim}
Extracting and saving the page number turned out to be a source of subtle bugs. Initially it was done through an `\edef` with a bunch of `\expandafter` commands inside. This posed a problem if the page number itself contained code which needed protection (e.g., pr/4080) so this got changed to use `\protected@edef`. However, that in turn failed with Babel (pr/4093) if the label contained active characters, e.g., a “:” in French. So now we use even more `\expandafter` commands to avoid any expansion other than what is absolutely required—I’m sure there is now a simpler way to code this, but . . . in fact we didn’t use enough initially resulting in another bug (pr/4159):

```
\expandafter\expandafter\expandafter\vref@@pagenum
\csname r@#2\endcsname{}\@nil#1\%
\def\vref@@pagenum#1#2#3\@nil#4{\def#4{#2}}
```

\vrefpagenum Same as the above but this time we generate a warning for undefined labels.

```
\def\vrefpagenum#1#2{\@ifundefined{r@#2}\G@refundefinedtrue\@latex@warning{Reference ‘#2’ on page \thepage \space undefined}}{}\vref@pagenum{#1}{#2}
```

\vpagerefrange If both labels refer to the same page then we produce a `\vpageref`. Otherwise we produce the result of `\reftextpagerange`.

\vpagerefrange\@vpagerefrange\@svpagerefrange One could think of making this command even more clever by producing a special string if the difference between the two page number is one (but this is not done).

We need two versions to implement a star form:

```
\newcommand\@svpagerefrange[3]\[\reftextcurrent\]{\vrefpagenum\@tempa{#2}\vrefpagenum\@tempb{#3}\ifx\@tempa\@tempb\vpageref*[{#1}]{#2}\else\reftextpagerange{#2}{#3}\fi}
```

```
\newcommand\@vpagerefrange[3]\[\reftextcurrent\]{\vrefpagenum\@tempa{#2}\vrefpagenum\@tempb{#3}\ifx\@tempa\@tempb\vpageref\[{#1}\]{#2}\else\reftextpagerange{#2}{#3}\fi}
```

And here we put things together.

```
\newcommand\vpagerefrange{\@ifstar\@svpagerefrange\@vpagerefrange}
```
\vrefrange
\newcommand\vrefrange[3][\textcurrent]{\textlabelrange{#2}{#3} \vpagerefrange[\textcurrent]{#1}{#2}{#3}}

\Vref \Vref is the counterpart to \vref with \ref replaced by \Ref in its definition.
\DeclareRobustCommand\Vref{\@ifstar{\Vref@star}{}{\Vref@f}}
\Vref@star
\newcommand\Vref@star[2]{% \Ref{#2} \vpageref[\textcurrent]{#1}{#2}}
\Vref@f
\def\Vref@f{% \if@vrefhandlespace
\unskip \vref@space
\fi
\Vref@star % in hyperref this needs to be explicit
}
\vpagerefcompare
\DeclareRobustCommand\vpagerefcompare[2][\textcurrent]{\vrefpagenum\@tempa{#1}\vrefpagenum\@tempb{#2}\ifx\@tempa\@tempb\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}
\vpagerefnearby
\DeclareRobustCommand\vpagerefnearby[1][\textcurrent]{\leavevmode\@tempskipa\lastskip \unskip\global\advance\c@vrcnt\tw@ % keep oddness to not disturb \vrefvario\vref@label{\the\c@vrcnt @vn}\ifdim\@tempskipa=\z@ \else\hskip\@tempskipa\fi\vref@pagenum\@tempa{\the\c@vrcnt @vn}\is@pos@number\@tempa\ifcase\numexpr\@tempa-\@tempb+1\relax\or\or\else\expandafter\@thirdofthree\fi\@firstoftwo\else\expandafter\@secondoftwo\fi\%}
\vpagerefnearby
\vpagerefcomparenearby \texttt{\textbackslash vpagerefcomparenearby} True if the two pages have arabic page numbers are at most off by one so that \texttt{\textbackslash vpageref} would produce a special string.

This is in some sense a general version of \texttt{\textbackslash vpagerefnearby} for use in more complex situations. I.e., if you try to determine if a larger set of references would be close to the current page and you use each time \texttt{\textbackslash vpagerefnearby} then you generate a lot of internal labels for the current point. Thus by setting the label yourself you can save processing time and space. On the other hand you have to be careful to place the label into the right place which can be a bit complicated if you look at the definition of \texttt{\textbackslash vpagerefnearby} that does all this automatically for you. This is why the current macro is not discussed in the user interface section as it is really just for package developers.

\texttt{\textbackslash DeclareRobustCommand\textbackslash vpagerefcomparenearby[2]}{\% label1, label2, TRUE, FALSE
\texttt{\textbackslash vrefpagenum\textbackslash@tempa{#1}}%
\texttt{\is@pos@number\textbackslash@tempa}{%\%\%\%
\texttt{\textbackslash vrefpagenum\textbackslash@tempb{#2}}%
\texttt{\is@pos@number\textbackslash@tempb}{%\%\%\%
\texttt{\ifcase\numexpr\@tempa-\@tempb+1\relax
\texttt{\or \or \else \expandafter\@thirdofthree \fi\texttt{\@firstoftwo}}%
\texttt{\@secondoftwo}}%
\texttt{\@secondoftwo}}}%

9.2.2 Compatibility with older \LaTeX{} formats

\texttt{\textbackslash labelformat} We don’t need to define \texttt{\textbackslash labelformat} and \texttt{\textbackslash Ref} or change the definition of \texttt{\textbackslash refstepcounter} any longer because this is now done in the \LaTeX{} kernel. However, if this package is used together with an older kernel, this is not true, so for this special case we keep the definitions.

\texttt{\textbackslash ifx\textbackslash labelformat\textbackslash@undefined % assume we are in an older kernel
\texttt{\textbackslash def\textbackslash labelformat#1{\expandafter\textbackslash def\csname p@#1\endcsname\#1}}
\texttt{\DeclareRobustCommand\textbackslash Ref[1]{\protected\textbackslash def\textbackslash@tempa{\textbackslash ref[1]}%\expandafter\textbackslash MakeUppercase\textbackslash@tempa}}
\texttt{\textbackslash def\textbackslash refstepcounter#1{\stepcounter#1%\texttt{\textbackslash protected\textbackslash def\textbackslash@currentlabel\texttt{\{\csname p@#1\expandafter\textbackslash endcsname\csname the#1\endcsname\}}}\texttt{\textbackslash fi}}
\texttt{\textbackslash fi}}
Change History

v0.1e  \vr@cnt: use LaTeX counter ........... 23
v0.1f  \@@@@@pageref: Add another label
to catch cross-overs .............. 24
v0.1g  \vref@err: Macro added ........... 28
        \vrefwarning: Macro added ........ 28
v0.1h  General: Small documentation
        changes .......................... 8
v1.0a  General: Upgrade to LaTeX2e ........ 1
v1.0b  \reftextcurrent: Replace in
        incorrect 'def by 'let .............. 16
        \vref@addto: Avoid using babel
        'addto .......................... 8
        The babel commands are
        starting with extras ............ 8
v1.0f  \reftextcurrent: Inserted
        missing hash mark .............. 21
v1.0g  \@@@@@pageref: Looping pages
        shown in wrong order ............ 24
        \reftextcurrent: Brazil defaults
        added .......................... 11
        \vref@err: Macro added ........... 28
        General: Use
        \DeclareRobustCommand ........ 1
v1.0h  \@@@@@pageref: Change internal
        interface ........................ 23
        \@@@@@pageref: Pass first argument
        in both cases ..................... 23
v1.0i  \@@@@@pageref: Do it correctly ....... 23
v1.0j  \reftextcurrent: Added french
        defaults .......................... 15
v1.0k  \reftextcurrent: Corrected
        misspelled cmd 'extrasspanish 21
v1.0l  \@@@@@pageref: Explicitly expand
        \label argument for babel ....... 27
        Explicitly expand 'label
        argument for babel ............. 24
        \reftextcurrent: Added dutch
        defaults .......................... 14
v1.0m  \reftextcurrent: Added finnish
        strings .......................... 15
v1.0n  \reftextcurrent: Added Danish
        defaults .......................... 13
v1.0o  \reftextcurrent: Added defaults
        for breton ....................... 12
v1.0p  \reftextcurrent: Added swedish
        strings .......................... 21
v1.0q  \reftextcurrent: Added defaults
        for norsk ........................ 18
        Added defaults for nynorsk .... 18
        \vref@excuse: Use
        'PackageWarning ................ 9
v1.1b  \reftextcurrent: Added defaults
        for Italian ...................... 17
v1.1c  \reftextcurrent: Added defaults
        for catalan ...................... 13
v1.1d  \reftextcurrent: Added defaults
        for polish .................... 19
v1.1e  \vr@f: Don’t use \nobreakspace
        (pr/1866) ....................... 28
v1.1f  \reftextcurrent: Added galician
        defaults ........................ 15
v1.1g  \@@@@@pageref: Test if inside AMS
        display (pr2175) ................ 24
v1.1h  \vrefshrovers: Macro added ........ 28
        General: Added draft and final
        options .......................... 22
v1.2a  \@@@@@pagerefrange: Macro added .... 30
        \reftextcurrent: Added
        american defaults for
        \reftextpagerange and
        \reftextlabelrange ............. 9
        Added austrian defaults
        (identical to german) ......... 10
        Added defaults for
        \reftextpagerange and
        \reftextlabelrange ............. 20
Added defaults for greek ... 16
Added defaults for magyar ... 17
Added dutch defaults for \reftextpagerange and \reftextlabelrange ... 14
Added english defaults for \reftextpagerange and \reftextlabelrange ... 14
Added french defaults for \reftextpagerange and \reftextlabelrange ... 15
Added norsk defaults for \reftextpagerange and \reftextlabelrange ... 18
Added nynorsk defaults for \reftextpagerange and \reftextlabelrange ... 18
Added polish defaults for \reftextpagerange and \reftextlabelrange ... 19
Added russian strings ... 20
Added swedish defaults for \reftextpagerange and \reftextlabelrange ... 21

Removed incorrect warning ... 17
Updated defaults for norsk ... 18
Updated defaults for nynorsk ... 18
\vr@f: Add \leavevmode ... 28
Use \nobreakspace (pr/2909) ... 28
\vref@pagenum: Macro added ... 29
\vref@stringwarning: Macro added ... 9
\vrefpagenum: Macro added ... 30
\vrefrange: Macro added ... 31

v1.2b
\reftextcurrent: Added portuguese defaults ... 19
v1.2c
\reftextcurrent: Added more portuguese defaults ... 19
v1.2d
\reftextcurrent: Added austrian defaults (identical to german) ... 10
Added ngerman defaults (identical to german) ... 16
v1.2e
\reftextcurrent: Added czech defaults ... 13
v1.2f
\vrefpageref: Don’t output space if optional argument empty (pr/3230) ... 25

v1.2g
\reftextcurrent: Updated defaults for catalan ... 13
v1.2h
\reftextcurrent: Updates to danish defaults ... 13
v1.2i
\reftextcurrent: Updated nynorsk defaults ... 18
v1.3a
\vpageref: Scan for star form ... 23
\vref: Scan for star form ... 28
v1.3b
\@svpagerefrange: Scan for star form ... 30
v1.3c
\vr@f: Correct typo in name ... 28
v1.4a
\Vr@f: Macro added ... 31
\Vref: Macro added ... 31
\reftextcurrent: Contributed defaults added ... 20
Missing defaults added ... 21
\vref@maybe@space: Default added ... 23
v1.4b
\reftextcurrent: Contributed defaults added ... 20
v1.4c
\@@@@vpageref: Use \thestvpagerefrnum instead of \@tempb ... 24, 27
\vpagerefrnum: Macro added ... 28
v1.4d
\reftextcurrent: Changed default for \reftextbefore 9, 14
Changed default for \reftextlabelrange ... 14
v1.4e
\@@@@vpageref: Use \vref@label instead of \label (pr/3489) ... 24, 27
\vref@label: Macro added (pr/3489) ... 27
v1.4g
\reftextcurrent: Added afrikaans option ... 9
v1.4i
\reftextcurrent: Moved \AtBeginDocument ... 18
v1.4j
\is@pos@number: Use \@nil for testing ... 29
\vpagerefcopareneartby: Macro added .................................. 32
\vpagerefnearby: Macro added ................................ 31
\vrefdef: Adjusted for new spacing regime ................... 28
\vref: Adjusted for new spacing regime ....................... 28
\vref@star: Command added to support hyperref better ...... 28
General: \labelformat and \Ref now available in the kernel so
removed from this package but only in new kernel ........ 1
Provide options space and nospace .......................... 22
v1.6b \reftextcurrent: Option arabic added .................. 10
v1.6c \vref@maybe@space: Support (mis)use of internal interface by fancyref (sx/515106) .... 23