The varioref package∗

Frank Mittelbach

2022/01/09

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
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 26
9.2.1 Supporting ranges 33
9.2.2 Compatibility with older \LaTEX formats . . . . . 36

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 v1.6f, last revised 2022/01/09.
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 no space, i.e.,

\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

\vref The implementation of \vref below produces only a \ref when reference and
\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 \ref and \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., \pagenumbering{roman}) then there will be
no distinction between one or many pages off.

\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 \vpageref command. It will produce the same
strings as \vref except that it doesn’t start with the \ref. A further difference
is that it will produce the string that is saved in \reftextcurrent if label and
reference fall onto the same page (and no optional argument is used). By defining
\reftextcurrent to produce “on this page” or something similar, we can avoid
that

\ldots see the example \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 \vpageref it will be ignored if the command doesn’t produce
any text at all.

But in fact \vpageref allows even more control. It 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 \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 be-
fore which shows \ldots” (or something similar depending on the settings of the
\refreset before commands) when there was a page break in the meantime. One warning however, if you use \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

\[ ... see the \vpageref[above example][example]{ex:foo}\]

which shows ...

to achieve the desired effect.

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

\[ ... see figures \vrefrange{fig:a}{fig:c} ... \]

which would then be formatted as

\[ ... see figures 3.4 to 3.6 on pages 23–24 ... \]

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

\[ ... see figures 3.4 to 3.6 on the following page ... \]

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 \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.

\vpagerefrange This command is similar to \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 \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.

\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 \label or \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 ifthen package)

\newcommand\amusingversion[2]{the definition%
    \vrefpagenum\firstnum[#1]%
    \vrefpagenum\secondnum[#2]%
    \ifthenelse{\equal\firstnum\secondnum}3
{s of \ref{#1} and \ref{#2} \vpageref{#1}}
{ of \ref{#1} \vpageref{#1} and of \ref{#2} \vpageref{#2}}
}

...\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.

\vpagerefcompare

For this kind of application the package also provides \vpagerefcompare as a command that takes four argument: two labels for comparison and a \langle true\rangle and \langle false\rangle argument. One of them is executed depending on whether the two labels are on the same page or on different pages. With its help the above definition could be shortened to

\newcommand\amusingversion[2]{the definition}\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.

\vpagerefnearby

Another command for conditional processing is \vpagerefnearby. It takes three arguments: a label and a \langle true\rangle and \langle false\rangle argument. It compares the page reference to the label with the current page number and if both are Arabic number and only differ by ±1 the \langle true\rangle argument is executed otherwise the \langle false\rangle argument. In other words it tells you if a \vpageref to the label would result in a textual reference to the previous, current or next page.\footnote{There is also a \vpagerefcomparenearby but this is only meant for package writers, so it is only documented in the implementation section.}

If you don’t use the nospace option the user commands \vref, \vpageref, and \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 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 too many older documents.

\begin{verbatim}
\labelformat
A reference via \ref produces by default the data associated with the corre-
sponding \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.,

\begin{verbatim}
\labelformat{section}{section"#1}
\labelformat{equation}{equation"(#1)}
\end{verbatim}
\end{verbatim}
\end{verbatim}

\Ref A side effect of using \labelformat is that, depending on the defined for-
matting, it becomes impossible to use \ref at the beginning of a sentence (if its
replacement 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 \labelformat has to be a simple ASCII or UTF-8 letter, oth-
erwise 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-
tilized. For example, for figure references in the Hungarian language you
might want to write \begin{verbatim} \labelformat{figure}{{\'{a}bra~\thefigure} \end{verbatim} or use
\begin{verbatim} \labelformat{figure}{`abra~\thefigure} \end{verbatim} which avoids the brace problem.

\begin{verbatim}
\thevpagerenum
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 \thevpagerenum can be used. This macro hold the current page “number”
when \vpageref and friends are executed. Thus, by defining, for example
\begin{verbatim}
\renewcommand\reftextfaceafter {on page~\thevpagerenum}
\renewcommand\reftextfacebefore {on page~\thevpagerenum}
\renewcommand\reftextafter {on page~\thevpagerenum}
\renewcommand\reftextbefore {on page~\thevpagerenum}
\end{verbatim}
\end{verbatim}

\end{verbatim}
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:

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

(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 varioref then it will start using the last language selected. Switching of languages is managed through Babel; nevertheless all languages of interest need to be specified when varioref is loaded so that it can integrate their strings into the Babel system.

5 Customization

The package supports all options defined by the 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 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 following commands after the package has been loaded. Backward references use \refbefore if the label is on the preceding page but invisible and \refafter if it is one the facing page (i.e., if the current page number is odd). Similarly \refafter is used when the label comes on the next page but one has to turn the page and \refafter if it is on the following but facing page.

In fact, \refface... is used only if the user or the document class specified two-sided printing.

Finally we have \reffaraway 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 can make use of \pageref in its replacement text.

To allow a bit random variation in the generated strings one can use the command \reffvario inside the string macros. It takes two arguments and selects one or the other for printing depending on the number of already seen \vref or \vpageref commands. As an example see the definitions of \refbefore etc. on page 9.

The commands \reffrange and \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.

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

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

In some languages (currently only for Japanese) the generated text has to typeset in different word order to most other (Western) languages. To accommodate for this \reffformat, etc. has been added to allow to alter this on a per language basis. There should be normally no need to alter the setup. For details see the code section.
5.1 ... if Babel is used

If babel is being used then the above commands need to be changed using the Babel mechanism, e.g.,

\addto\extrasngerman{
  \renewcommand\reftextfaceafter {auf der n"achsten Seite}\
  ...
}

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 \texttt{draft} to turn error messages into warnings during development. The default \texttt{final} produces error message when a generated string falls onto a page boundary (see next section).

Newly introduced in 2019 are the options \texttt{space} (default) and \texttt{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 \texttt{nospace}. This will stop \texttt{varioref} from meddling with the spaces preceding commands and thus make the star forms \texttt{\vref*} and \texttt{\vpageref*} unnecessary.

This means that the star forms can then be used as defined by \texttt{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 \texttt{varioref} therefore issues an error message whenever the generated text is broken across page boundaries, e.g.,

\vrefwarning

\texttt{table 5 on the current (page break) page}

would result in an error, which needs to be resolved by the user by replacing the \texttt{\vref} command with an ordinary \texttt{\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 \texttt{\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 \vref 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\footnote{Not really that likely these days, but one never knows . . . . In any case \TeX will process your document a tiny bit faster.} 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 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 \LaTeXe 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}[^filedate\space\fileversion\space\space}
\PackageRequired{varioref} for extended references (FMi)]
\end{verbatim}
To support the use of babel we want to add the additional strings to the \extras commands. Since \addto in the current implementation of babel has a bug that does not allow to use arguments containing hashmarks we do this by hand.

\begin{verbatim}
def\vref@addto#1#2{% #2% \@temptokena{#2}% \ifx#1\undefined\edef#1{\the\@temptokena}% \else \toks@expandafter{#1}% \edef#1{\the\toks@\the\@temptokena}% \fi \@temptokena{}\toks@\@temptokena}
\end{verbatim}

\section{Options}

\vref@excuse Excuse that we don't know the strings for a certain language.

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

\vref@stringwarning And we sometimes just don't know some strings ...

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

\vref@addto 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.

Afrikaans option contributed by Danie Els (dnjels@sun.ac.za).

\begin{verbatim}
\DeclareOption{afrikaans}{\vref@addto\extrasafrikaans{%
\end{verbatim}
Arabic option text was contributed by Ihab Samir Mattar (dr_kremlin@hotmail.com). As our standard documentation workflow uses pdflatex 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.

Arabic Option (arabic)

\DeclareOption{arabic}{
\let\vrefformat\vrefdefaultformat
\let\Vrefformat\Vrefdefaultformat
\let\fullrefformat\fullrefdefaultformat
\let\vrefrangeformat\vrefrangedefaultformat
}\}

10
The Austrian defaults are the same as the German ones.

The Austrian defaults are the same but needed for "Neue Rechtschreibung".

Text for Basque defaults was contributed by Iñaki 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.

\DeclareOption{bahasa}{
  \vref@addto\extrasbahasa{
    \def\reftextfaceafter {pada muka surat \reftextvario{bertentangan}{berikutnya}}
    \def\reftextfacebefore {pada muka surat \reftextvario{bertentangan}{sebelumnya}}
    \def\reftextafter {pada muka surat berikutnya}
    \def\reftextbefore {pada muka surat sebelumnya}
    \def\reftextcurrent {pada muka surat ini}
    \def\reftextfaraway#1{pada muka surat~\pageref{#1}}
    \def\reftextpagerange#1#2{pada muka surat~\pageref{#1}--\pageref{#2}}
    \let\vrefformat\vrefdefaultformat
    \let\Vrefformat\vrefdefaultformat
    \let\fullrefformat\fullrefdefaultformat
    \let\vrefrangeformat\vrefrangedefaultformat
  }}

Text for Brazil defaults was contributed by Alcino Dall Igna Junior (adij@di.ufpe.br).

\DeclareOption{brazil}{
  \vref@addto\extrabrazil{
    \def\reftextfaceafter {na \reftextvario{página oposta}{próxima página}}
    \def\reftextfacebefore {na página \reftextvario{oposta}{anterior}}
    \def\reftextafter {na \reftextvario{página seguinte}{próxima página}}
    \def\reftextbefore {na página \reftextvario{anterior}{precedente}}
    \def\reftextcurrent {nesta página na página corrente}
    \def\reftextfaraway#1{na \reftextvario{página}{página ~\pageref{#1}}}
    \def\reftextpagerange#1#2{nas \paginas ~\pageref{#1}--\pageref{#2}}
    \let\vrefformat\vrefdefaultformat
    \let\Vrefformat\vrefdefaultformat
    \let\fullrefformat\fullrefdefaultformat
    \let\vrefrangeformat\vrefrangedefaultformat
  }}

Text for Breton defaults was contributed by Christian ROLLAND (Christian.Rolland@univ-brest.fr).

\DeclareOption{breton}{
  \vref@addto\extrabreton{
    \def\reftextfaceafter {war ar bajenn \reftextvario{a-dal}{da heul}}
    \def\reftextfacebefore {war ar bajenn \reftextvario{a-dal}{a-raok}}
    \def\reftextafter {war ar bajenn da heul}
    \def\reftextbefore {war ar bajenn a-raok}
    \def\reftextcurrent {war ar \reftextvario{bajenn-mañ}{bajenn war-wel}}
    \def\reftextfaraway#1{war ar bajenn ~\pageref{#1}}
    \def\reftextpagerange#1#2{\vref@stringwarning\reftextpagerange on pages ~\pageref{#1}--\pageref{#2}}
  }}
Text for croatian defaults was contributed by Šime Ungar (ungar@math.hr).

Defaults for the Czech option provided by Milan Vancura <milan@suse.cz>. Further defaults by Ole Laursen (olau@hardworking.dk).

Defaults for Danish provided by Torsten Martinsen (tmar91@kom.auc.dk). Further defaults by Ole Laursen (olau@hardworking.dk).
Default string for Dutch have been contributed by Frank Poppe (POPE@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 Fouvry <fouvry@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)."

Concerning the next defaults I received the following comment by Frederik Fouvry <fouvry@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).

Can’t combine numbers with the necessary suffix well.

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 gegenüberliegenden Seite}\
\def\reftextfacebefore {auf der gegenüberliegenden Seite}\
\def\reftextafter {auf der nächsten Seite}\
\def\reftextbefore {auf der vorherigen Seite}\
\def\reftextcurrent {auf dieser Seite}\
\def\reftextfaraway#1{auf Seite~\pageref{#1}}\
\let\vrefformat\vrefdefaultformat\
\let\Vrefformat\Vrefdefaultformat\
\let\fullrefformat\fullrefdefaultformat\
\let\vrefrangeformat\vrefrangedefaultformat\}

\DeclareOption{german}{\vref@addto\extrasgerman{\
\def\reftextfaceafter {auf der nächsten 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}}}\}

17
The ngerman defaults are the same but needed for “Neue Rechtschreibung” (i.e., a different set of hyphenation patterns).

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

Icelandic defaults suggested by Thorhallur Sverrisson (thorhs@basis.is).
Defaults for Italian suggested by Giovanni Pensa (pensa@dsi.unimi.it) with iTExnici.

\DeclareOption{italian}{
  \vref@addto\extrasitalian{% 
    \def\reftextbefore {\reftextvario{a fronte} \{nella pagina successiva\}%
    \def\reftextcurrent {\reftextvario{a fronte} \{nella pagina precedente\}%
    \def\reftextafter {\reftextvario{a fronte} \{successiva\}%
    \def\reftextbefore {nella pagina precedente}%
    \def\reftextcurrent {nella pagina successiva}%
    \let\vrefformat\vrefdefaultformat
    \let\Vrefformat\vrefdefaultformat
    \let\fullrefformat\fullrefdefaultformat
    \let\vrefrangeformat\vrefrangedefaultformat
  }}
}

Defaults for Japanese. It needs a special \vrefformat, \Vrefformat, \fullrefformat and \vrefrangeformat for grammatical reasons. As our standard documentation workflow uses pdf\TeX{} the Japanese 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.

\DeclareOption{japanese}{
  \vref@addto\extrasjapanese{% 
    \def\reftextfaceafter {\reftextvario{}}%
    \def\reftextfacebefore {\reftextvario{}}%
    \def\reftextafter {\reftextvario{}}%
    \def\reftextbefore {\reftextvario{}}%
    \def\reftextcurrent {\reftextvario{}}%
    \def\reftextfaraway#1{\pageref{#1}}%
    \def\reftextpagerange#1#2{\pageref{#1}--\pageref{#2}}%
    \def\reftextlabelrange#1#2{\ref{#1} til\ref{#2}}%
    \let\vrefformat\vrefdefaultformat
    \let\Vrefformat\vrefdefaultformat
    \let\fullrefformat\fullrefdefaultformat
    \let\vrefrangeformat\vrefrangedefaultformat
  }}

Note that the parentheses below are not normal ones but full width ones U+FF08 and U+FF09!
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.}

\begin{verbatim}
\DeclareOption{magyar}{
  \let\fullrefformat\fullrefdefaultformat
  \let\vrefrangeformat\vrefrangedefaultformat
}
\AtBeginDocument{
  \providecommand\aza[1]{a/az\if\@cap\capitalize\else\lowercase\fi\expandafter\aza\fi
\PackageError{varioref}{Proper definition for command \aza missing}{You need to (re)define the command \aza that takes one argument (a number or a \pageref command) and produces the string ‘a’ or ‘az’ depending on the argument value. It should also accept an optional argument (default ‘l’) to denote whether or not the string should be lowercase or

\begin{verbatim}
\DeclareOption{magyar}{
  \def\vrefrangeformat#1#2#3\vpagerefrange[#1]{#2}{#3}
}
\def\vpagerefrange[#1]{{}\vpagerefrange[#1]{}}
}}
\end{verbatim}
\end{verbatim}

The following suggested patch has not been added yet (pending changes in babel system).

\begin{verbatim}
\def\reftextfaraway#1{\aza{\pageref{#1}}\pageref{#1}.~ol\da\kon}
\def\reftextpagerange#1#2{\aza{\pageref{#1}}--\pageref{#2}.~ol\da\kon}
\def\reftextlabelrange#1#2{{}\vref@stringwarning\reflabelpagerange\ref{#1} to \ref{#2}}
\end{verbatim}
\end{verbatim}
\end{verbatim}

The \AtBeginDocument needs to be executed when the option is processed and not inside the babel interface.
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}
{\vref@addto\extrasnorsk{
\def\reftextfaceafter {p\aa{} neste side}\%
\def\reftextfacebefore{p\aa{} forrige side}\%
\def\reftextbefore {p\aa{} forrige side}\%
\def\reftextcurrent {p\aa{} denne siden}\%
\def\reftextfaraway#1{p\aa{} side\pageref{#1}}%
\def\reftextpagerange#1#2{p\aa{} side\pageref{#1}--\pageref{#2}}%
\def\reftextlabelrange#1#2{\ref{#1} til\ref{#2}}%
\let\vrefformat\vrefdefaultformat
\let\Vrefformat\Vrefdefaultformat
\let\fullrefformat\fullrefdefaultformat
\let\vrefrangeformat\vrefrangedefaultformat
}}

\DeclareOption{nynorsk}
{\vref@addto\extrasnynorsk{
\def\reftextfaceafter {p\aa{} neste side}\%
\def\reftextfacebefore{p\aa{} forrige side}\%
\def\reftextbefore {p\aa{} forrige side}\%
\def\reftextcurrent {p\aa{} denne siden}\%
\def\reftextfaraway#1{p\aa{} side\pageref{#1}}%
\def\reftextpagerange#1#2{p\aa{} side\pageref{#1}--\pageref{#2}}%
\def\reftextlabelrange#1#2{\ref{#1} til\ref{#2}}%
\let\vrefformat\vrefdefaultformat
\let\Vrefformat\Vrefdefaultformat
\let\fullrefformat\fullrefdefaultformat
\let\vrefrangeformat\vrefrangedefaultformat
}}

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.

\DeclareOption{polish}
{\vref@addto\extraspolish{
\def\reftextfaceafter {na \reftextvario{s\aob{}siedniej}{} siedniej}{} stronie}\%
\def\reftextfacebefore{na \reftextvario{s\aob{}siedniej}{} siedniej}{} stronie}\%
\def\reftextbefore {na \reftextvario{poprzedniej}{} stronie}\%
\def\reftextcurrent {na \reftextvario{tej}{} stronie}\%
\def\reftextfaraway#1{na stronie\pageref{#1}}%
\def\reftextpagerange#1#2{na stronach\pageref{#1}--\pageref{#2}}%
\def\reftextlabelrange#1#2{od\ref{#1} do\ref{#2}}%
\let\vrefformat\vrefdefaultformat
\let\Vrefformat\Vrefdefaultformat
\let\fullrefformat\fullrefdefaultformat
\let\vrefrangeformat\vrefrangedefaultformat
}}
The Portugese default text has been provided by José Carlos Oliveira Santos jcsantos@fc.up.pt.

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).

Updates to the default text for Romanian have been suggested by Florin Oprina (<florin.oprina@gmail.com>).
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).

\DeclareOption{slovak}{\vref@addto\extrasslovak{%}
{\def\reftextfaceafter {na \reftextvario{proti\v{ }lahlej}%
{nasleduj\^{}\'ucej} strane}%
{\def\reftextfacebefore{na \reftextvario{proti\v{ }lahlej}%
{predch\^{}\'adzaj\'ucej} strane}%
{\def\reftextafter {na \reftextvario{nasleduj\^{}\'ucej}%
{{\v{ } dal\^{}\'v sej} strane}%
{\def\reftextbefore {na \reftextvario{predch\^{}\'adzaj\'ucej}%
{{\v{ } slej} strane}%
{\def\reftextcurrent {na tejto strane}%
{\def\reftextfaraway#1{na strane \pageref{#1}}%
{\def\reftextpagerange#1#2{na straneh \pageref{#1}--\pageref{#2}}%
{\let\vrefformat\vrefdefaultformat%
{\let\Vrefformat\vrefdefaultformat%
{\let\fullrefformat\fullrefdefaultformat%
{\let\vrefrangeformat\vrefrangedefaultformat%}}

\DeclareOption{slovene}{\vref@addto\extrasslovene{%}
{\def\reftextfaceafter {na \reftextvario{prednih}%
{naslednjih} stranih}%
{\def\reftextfacebefore{na \reftextvario{prednih}%
{prednih} stranih}%
{\let\reftextafter \reftextfaceafter%
{\let\reftextbefore \reftextfacebefore%
{\def\reftextcurrent {na \reftextvario{trenutni} strani}%
{\def\reftextfaraway#1{na strani \pageref{#1}}%
{\def\reftextpagerange#1#2{na strani \pageref{#1}--\pageref{#2}}%
{\let\vrefformat\vrefdefaultformat%
{\let\Vrefformat\vrefdefaultformat%
{\let\fullrefformat\fullrefdefaultformat%
{\let\vrefrangeformat\vrefrangedefaultformat%}}}}}
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 Turkish was contributed by Taha Yildirim (t.yildirim@boun.edu.tr).
The text for Ukrainian defaults was contributed by Mykola Lyakhovsky
(<lyakhovsky@mail.ru>).
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.

\newcounter{vrcnt}

\vpageref\vp@geref
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 `hyperef` to redefine it and use it for its own tasks (like denoting that no link should be produced).

\DeclareRobustCommand\vpageref\@ifstar
\DeclareRobustCommand\vpageref\@ifnextchar[

The default for the first optional argument is the string hidden in the macro `\reftextcurrent`.

\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 `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 should hopefully produce the same behavior as before).

\let\vref@space\space\let\vref@maybe@space\space\@vpageref
More parsing...

\def\vpageref[#1]\@ifnextchar[

By default the second optional argument is empty.
The \@vpageref macro finally generates the references by comparing the value of an internal label with the value of the user label.

First it switches to horizontal mode if necessary. If there was a preceding space we remember its size in \@tempskipa and then remove it for reinsertion later.

Then it advances the counter \c@vrcnt by one which is used to generate internal label names.

Then it checks if for the current value of \c@vrcnt a label command was issued in the last run. If not it pretends that there was one with the value {??}{??}. Thereafter it stores the pageref value for this label in the macro \@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 \label command. In the babel system the argument of \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 \vref@label instead of \label as this avoids problems with Babel or 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 space was specified, we insert our own space, namely \vref@maybe@space. This will typeset a space unless the star form of the current command was used.

But if 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 amsmath package the \label command is redefined which makes the test for loops incorrect if a \vref is used in, say, \intertext. So we test this condition first and only do the test if \label doesn’t have a special meaning.

[This is no longer necessary as \label isn’t used any more]
\let\vref@may@loop\relax
\else
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.
\edef\vref@may@loop{%
\noexpand\vref@err{\noexpand\vref or \noexpand\vpageref
at page boundary
\@tempb-\@tempa space (may loop)}%
}%
\fi
% \fi
Same game for the user requested label; this time the page reference is saved in
\thevpagerefnum.
\vrefpagenum\thevpagerefnum{#3}%
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@current@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 `\@tempcnta` and add one to it.

```latex
\if@pos@number\@tempa
\{\@tempcnta\@tempa
\advance\@tempcnta\@one
\}
```

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

```latex
\{\@tempcnta\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.

```latex
\ifeven\@tempcnta
\fi
```

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.

```latex
\ifodd\@tempcnta
\reftextfaceafter
\else
\reftextafter
\fi
```

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.

```latex
\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, theorems, etc. To test this we now subtract two from the current value of `\@tempcnta` (which was set to one higher as the reference page number). Note, that subtraction is also possible if the value was `\maxdimen` — we still get something that is much larger than any sensible page number.

```latex
\else
\advance\@tempcnta\@two
```

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

```latex
\ifeven\thevpagerefnum\@tempcnta
```

29
Again we have to check for odd or even pages to distinguish between the facing and the non-facing situation.

\ifodd\@tempcnta
  \reftextbefore
\else
  \reftextfacebefore
\fi

Again the \reftextfacebefore only if \@twoside evaluates to true.

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

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.

\else
  \let\vref@may@loop\relax % added
  \reftextfaraway{#3}%
\fi

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

\else
  \let\vref@may@loop\relax \% added
  \reftextfaraway{#3}\%
\fi

Finally we generate the internal label so that it can be check 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.

\vref@label{\the\c@vrcnt \@vr}\%

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

\vref@may@loop

\}

\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 \ifdefined\vp@gerefstar
... code for new varioref
\else
... code for old varioref
\fi

856 \let\vp@geref\vp@geref

\ref@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.

857 \def\ref@label#1{\@bsphack
858 \protected@write\@auxout{}\%
859 \{\string\newlabel{#1}{{}{\thepage}}\%
860 \@esphack}

\vpagerefnum This macro allows to define \reftextafter and friends to refer to the page number rather than producing a textual reference.

861 \let\thevpagerefnum@empty

\vrefwarning 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.

862 \def\vrefwarning{\def\vref@err{\PackageWarning{varioref}}} 863 \def\vrefshowerrors{\def\vref@err##1{\PackageError{varioref}{##1} \%
864 \{Please check the pages in question. You might need to replace \string\vref\MessageBreak
865 \text\string\vpageref by a normal \noexpand\(page)ref to stop LaTeX running forever.\}}}}

\vref We start by scanning for a star form and then continue in a different macro depending on the result.

868 \DeclareRobustCommand\vref{\@ifstar
869 {\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

870 \newcommand{\vref@star[1]}[1]{\vrefformat{#1}}

\vrefdefaultformat Japanese needs a different word order in \vref@star so this is separated out of \vrefformat that it can be changed on language level.

871 \def\vrefdefaultformat#1#2{\%
872 \ref{#2}
873 \vpageref[#1]{#2}\}

Most languages need the above definition.

874 \let\vrefformat\vrefdefaultformat

\vr@f The normal case is a little bit more elaborated ...

875 \def\vr@f{%
876 \leavevmode

31
If option space is used then we remove a preceding space and add our own un-
breakable 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 \nobreak\space 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

\is@pos@number 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.

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

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.

\def\is@pos@num{
\afterassignment\is@pos@num@
\count@}

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.

\def\is@pos@num@#1\@nil{%
Now we have to check if the argument is empty. We can do this with an \ifx (i.e.,
in an expandable way) since we know for sure that it can not start with \@nil.
\ifx\@nil#1\@nil
Depending on the result we execute the second or third argument of the command
\is@pos@number using a technique that removes the \else or \fi from the input
stream first, so that in principle input stream parsing could be done from within
the arguments.

\expandafter@firstoftwo \else
\expandafter@secondoftwo \fi}
\reftextvario  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}
\fullref  And here is the primitive command that always produces a \ref and a \pageref.
\begin{verbatim}
def\fullrefdefaultformat#1{\ref{#1} \reftextfaraway{#1}}
\end{verbatim}
\fullrefformat  Japanese needs a different word order in \fullref so this is separated out o that it can be changed on language level.
\begin{verbatim}
def\fullrefformat#1{\fullrefformatdefaultformat{#1}}
\end{verbatim}
\fullrefformatdefaultformat  Most languages need the above definition.
\begin{verbatim}
\let\fullrefformat\fullrefformatdefaultformat
\end{verbatim}

9.2.1  Supporting ranges
\vrefpagenum  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\vrefpagenum#1#2{%\@ifundefined{r@#2}{{??}{??}{}{}{}}{}%\vref@pagenum{#1}{#2}}
\end{verbatim}
\vref@pagenum  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):
\begin{verbatim}
def\vref@pagenum#1#2#3\@nil#4{%\expandafter\expandafter\expandafter\vref@@pagenum
\csname r@#2\endcsname{}{}\@nil#1%}
def\vref@@pagenum#1#2#3#4{%\def#4{#2}%
\end{verbatim}
\vrefpgennum  Same as the above but this time we generate a warning for undefined labels.
\begin{verbatim}
def\vrefpgennum#1#2{%\@ifundefined{r@#2}{%\latex@warning{Reference ‘#2’ on page \thepage \space undefined}}{%}
\vref@pagenum{#1}{#2}}
\end{verbatim}
\vpagerefrange  If both labels refer to the same page then we produce a \vpageref. Otherwise we \svpagerefrange produce the result of \reftextpgenrange.

33
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:

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

And the same but using `\vpageref`:

```latex
\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.

```latex
\newcommand\vpagerefrange{\@ifstar\@svpagerefrange\@vpagerefrange}
```

Japanese needs a different word order in `\vrefrange` so this is separated out so that it can be changed on language level.

```latex
\def\vrefrangedefaultformat#1#2#3{\reftextlabelrange{#2}{#3} \vpagerefrange[#1]{#2}{#3}}
```

Most languages need the above definition.

```latex
\let\vrefrangeformat\vrefrangedefaultformat
```

`\Vref` is the counterpart to `\vref` with `\ref` replaced by `\Ref` in its definition.

```latex
\DeclareRobustCommand\Vref{\@ifstar
  \Vref@star
  \Vrefdefaultformat}
```

Japanese needs a different word order in `\Vref` so this is separated out so that it can be changed on language level.

```latex
\def\Vrefdefaultformat#1#2{\Ref{#2} \vpageref[#1]{#2}}
```

Most languages need the above definition.

```latex
\let\Vrefformat\Vrefdefaultformat
```
\def\Vr@f{\if@vrefhandlespace \unskip \vref@space \fi \Vref@star} % in hyperref this needs to be explicit

\DeclareRobustCommand\vpagerefcompare[2]{\vrefpagenum\@tempa{#1}\vrefpagenum\@tempb{#2}\ifx\@tempa\@tempb \expandafter\@firstoftwo \else \expandafter\@secondoftwo \fi}

\DeclareRobustCommand\vpagerefnearby[1]{\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 {\vrefpagenum\@tempb{\the\c@vrcnt @vn}\is@pos@number\@tempb \ifcase\numexpr\@tempa-\@tempb+1\relax \or \or \else \expandafter\@firstoftwo \@thirdoftwo \fi \@firstoftwo }% \@secondoftwo \% \@secondoftwo %}

\DeclareRobustCommand\vpagerefcomparenearby{True if the two pages have Arabic page numbers are at most off by one so that \vpageref would produce a special string. This is in some sense a general version of \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 \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 \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.

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

\section{Compatibility with older \LaTeX\ formats}
\labelformat\refstepcounter We don't need to define \labelformat and \Ref or change the definition of
\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.
\ifx\labelformat\@undefined % assume we are in an older kernel
\def\labelformat#1{\expandafter\def\csname p@#1\endcsname##1}
\DeclareRobustCommand\Ref[1]{\protected@edef\@tempa{\ref{#1}}%
\expandafter\MakeUppercase\@tempa}
\def\refstepcounter#1{\stepcounter{#1}%
\protected@edef\@currentlabel
{\csname p@#1\expandafter\endcsname\csname the#1\endcsname}}
\fi

\ProcessOptions*

\endinput

\begin{changehistory}
\section{Change History}

v0.1e
\vr@cnt: use \LaTeX\ counter .. 26
v0.1f
\@vpageref: Add another label
to catch cross-overs ........ 27
v0.1g
\vr@err: Macro added ....... 31
\vr@warn: Macro added ....... 31
v0.1h
General: Small documentation
changes ..................... 8

v1.0a
General: Upgrade to \LaTeXXe .. 1
v1.0b
\reftextcurrent: Replace incorrect \texttt{def} by \texttt{let} ........ 17
\vr@add: Avoid using \texttt{l}abel \texttt{\`addto} ............ 9
The \texttt{l}abel \texttt{\`addto} commands are
starting with \texttt{\@addto} ............ 9
v1.0f
\reftextcurrent: Inserted
missing \texttt{hash mark} ........ 24
v1.0g  
General: Use \DeclareRobustCommand \ @@vpageref: Looping pages shown in wrong order 28 \reftextcurrent: Brazil defaults added 12 \vreferr: Macro added 31

v1.0h  \@@vpageref: Change internal interface 27 \@vpageref: Pass first argument in both cases 26

v1.0i \@vpageref: Do it correctly 26

v1.0j \reftextcurrent: Added french defaults 16

v1.0k \reftextcurrent: Corrected misspelled cmd ‘extrasspanish 24

v1.0l \@@vpageref: Explicitly expand ‘\label argument for babel 30
Explicitly expand ‘\label argument for babel 27 \reftextcurrent: Added dutch defaults 15

v1.0m \reftextcurrent: Added finnish strings 16

v1.0n \reftextcurrent: Added danish defaults 14

v1.0o \reftextcurrent: Added defaults for breton 12

v1.0p \reftextcurrent: Added swedish strings 24

v1.0q \reftextcurrent: Added defaults for norsk 21
Added defaults for nynorsk 21 \vrefexcuse: Use ‘PackageWarning 9

v1.1b \reftextcurrent: Added defaults for Italian 19

v1.1c \reftextcurrent: Added defaults for catalan 13

v1.1d \reftextcurrent: Added defaults for polish 21

v1.1e \vref: Don’t use ‘\nobreakspace (pr/1866) 32

v1.1f \reftextcurrent: Added galician defaults 17

v1.1g \@@vpageref: Test if inside AMS display (pr2175) 27

v1.1h General: Added draft and final options 25 \vrefshowerrors: Macro added 31

v1.2a \@svpagerefrange: Macro added 33

\reftextcurrent: Added American defaults for \reftextpagerange and \reftextlabelrange 9
Added austrian defaults (identical to german) 11
Added defaults for \reftextpagerange and \reftextlabelrange 22
Added defaults for greek 18
Added defaults for magyar 20
Added dutch defaults for \reftextpagerange and \reftextlabelrange 15
Added english defaults for \reftextpagerange and \reftextlabelrange 15
Added french defaults for \reftextpagerange and \reftextlabelrange 16
Added norsk defaults for \reftextpagerange and \reftextlabelrange 21
Added nynorsk defaults for \reftextpagerange and \reftextlabelrange 21
Added polish defaults for \reftextpagerange and \reftextlabelrange 21
Added russian strings 22
Added swedish defaults for \reftextpagerange and \reftextlabelrange 24
Removed incorrect warning 19
Updated defaults for norsk 21
Updated defaults for nynorsk 21
v1.2b
\reftextcurrent: Added portuguese defaults .......... 22
v1.2c
\reftextcurrent: Added more portuguese defaults .. 22
v1.2d
\reftextcurrent: Added ngerman defaults (identical to german) ............... 18
v1.2e
\reftextcurrent: Added czech defaults .................. 14
v1.2f
@@vpageref: Don’t output space if optional argument empty (pr/3230) .......... 28
v1.2g
\reftextcurrent: Updated defaults for catalan .......... 13
v1.2h
\reftextcurrent: Updates to Danish defaults .......... 14
v1.2i
\reftextcurrent: Updated nynorsk defaults .............. 21
v1.3a
\vpageref: Scan for star form ................. 26
\vref: Scan for star form .............. 31
v1.3b
@svpagerefrange: Scan for star form ..................... 33
v1.3c
\vr@f: Correct typo in name .......... 31
v1.4a
\reftextcurrent: Contributed defaults added .......... 23
\reftextcurrent: Missing defaults added ....... 24
\vr@f: Macro added ............ 35
\vref: Macro added .......... 34
\vref@maybe@space: Default added .............. 26
v1.4b
\reftextcurrent: Contributed defaults added .......... 23
v1.4c
@@vpageref: Use thevpagerefnumber instead of \@tempb ............... 28
\vpagerefnum: Macro added ... 31
v1.4d
\reftextcurrent: Changed default for \reftextbefore 9, 15
\reftextcurrent: Changed default for \reftextlabelrange ...... 15
v1.4e
\@@vpageref: Use vref@label instead of \label (pr/3489) .............. 27, 30
\vref@label: Macro added (pr/3489) .................. 31
v1.4g
\reftextcurrent: Added afrikaans option .......... 9
v1.4i
\reftextcurrent: Moved \AtBeginDocument .......... 20
v1.4j
\is@pos@number: Use \nil for testing ............... 32
\reftextcurrent: Defined \reftextfaceafter and \reftextfacebefore instead of \reftextafter and \reftextbefore ............. 20
v1.4k
\reftextcurrent: Added missing defaults supplied by Lapo Mori .................. 19
\reftextcurrent: Basque defaults added. ...... 11
v1.4l
\reftextcurrent: Contributed defaults added .......... 25
v1.4m
\reftextcurrent: Fixed \reftextlabelrange (pr/3791) .................. 15
v1.4n
\reftextcurrent: Changed default for \reftextfaraway 21
\reftextcurrent: Small change .............. 17–19
v1.4o
\reftextcurrent: Added a few more brazil strings .......... 12
\reftextcurrent: Added a few more finnish strings .............. 16
\reftextcurrent: Added further galician defaults 17
\reftextcurrent: Bahasa Malaysia defaults added. .......... 12
v1.4p \texttt{\refextcurrent}: Added defaults for croatian \hfill 14
\vrefrange: Prevent loss of brackets \hfill 34
\vref@pagenum: Use \texttt{\protected@edef} to avoid problems in complicated setups, e.g., microtype \hfill 33
v1.4q \texttt{\refextcurrent}: Added defaults for bulgarian \hfill 13
v1.4r \texttt{\refextcurrent}: Misspelling of \texttt{\extrabrazil} \hfill 12
\texttt{\extraportuges}: Misspelling \hfill 22
v1.4s \texttt{@\vpagerefrange}: Prevent loss of brackets \hfill 33
v1.4t \texttt{\refextcurrent}: Added defaults for icelandic \hfill 18
v1.4u \texttt{\refextcurrent}: Correction for \texttt{\reffaraway} suggested by Lorenzo Pantieri \hfill 19
v1.4v \texttt{\refextcurrent}: Incorrect text in \texttt{\reffafter} (pr/4070) \hfill 22
v1.4w \texttt{\reffpagenum}: Use \texttt{\protected@edef} to pacify french babel \hfill 33
v1.4x \texttt{\refextcurrent}: Defaults for esperanto added \hfill 16
\texttt{\vref}: Correct spacing – same mistake as in \texttt{\ref} (pr/4123) \hfill 35
v1.4y \texttt{\reffpagenum}: Avoid even \texttt{\protected@edef} to pacify french babel \hfill 33
v1.4z \texttt{\reffpagenum}: Previous fix need one more expansion \hfill 33
v1.5a \texttt{\refextcurrent}: Correct text in \texttt{\reffbefore} \hfill 22
v1.5b \texttt{\refextcurrent}: Fixed blunder from 2005 - galician option clearly never used \hfill 17
v1.5c
\texttt{\refextcurrent}: General: Revamped some of the documentation to better explain how varioref could be used in a multi-lingual context \hfill 1
\texttt{\v@vpageref}: \hfill 30
\texttt{\delay}: Delay generating the looping warning until we know more \hfill 28
v1.6a \texttt{\labeldef} and \texttt{\ref} now available in the kernel so removed from this package \hfill 1
\texttt{\v@vpageref}: \hfill 25
\texttt{\vpagerefrange}: Don’t drop a preceding space completely but reinsert it later after setting a label \hfill 27
v1.6b \texttt{\v@vpageref}: Default for optional argument changed \hfill 26
\texttt{\vpagerէrefstar}: Command added to support hyperref better \hfill 31
\texttt{\v@vpageref}: Macro added \hfill 35
\texttt{\v@vpageref}: Macro added \hfill 35
\texttt{\v@vpageref}: Macro added \hfill 35
\texttt{\v@vpageref}: Macro added \hfill 35
\texttt{\v@vpageref}: Macro added \hfill 35
\texttt{\v@vpageref}: Macro added \hfill 35
\texttt{\v@vpageref}: Macro added \hfill 35
\texttt{\v@vpageref}: Macro added \hfill 35
\texttt{\v@vpageref}: Command added to support hyperref better \hfill 34
\texttt{\v@vpageref}: Command added to support hyperref better \hfill 31
\texttt{\v@vpageref}: Command added to support hyperref better \hfill 31
v1.6c \texttt{\reffpagenum}: Option arabic added \hfill 10
\texttt{\v@vpageref}: \hfill 26
\texttt{\v@maybewe@space}: Support \hfill 26
\texttt{\v@vpageref}: \hfill 26
\texttt{\v@vpageref}: (mis)use of internal interface by fancyref \hfill 26
v1.6d \texttt{\refextcurrent}: Option japanese added (gh/352) \hfill 19
\texttt{\v@vpageref}: Changed for Japanese \hfill 34
\texttt{\v@vpageref}: \hfill 34
\texttt{\v@vpageref}: Macro added (gh/352) \hfill 34
v1.6e
General: Move to handle the language-specific reference formats. (gh/352) ........ 36
\fullref: Changed for Japanese (gh/352) .................... 33
\fullrefformat: Macro added (gh/352) ............... 33
\reftextcurrent: Option
japanese changed (gh/352) . 19
\Vref@star: Changed for Japanese (gh/352) ............ 34
\vref@star: Changed for Japanese (gh/352) ............ 31
\Vrefdefaultformat: Macro added (gh/352) ............. 34
\vrefdefaultformat: Macro added (gh/352) ............. 31
v1.6f
\vref@pagenum: Use 3 additional (empty) arguments as default in case nameref is active . . 33