The `varioref` package*

Frank Mittelbach

2004/02/27

Abstract

This package defines the commands `\vref`, `\vpageref`, `\vrefrange`, and `\vpagerefrange` for LaTeX 2_ε. `\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.

1 Introduction

In many cases it is helpful when refering 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}}
```

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 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 refering to the current page. For this purpose the package defines the `\vpageref` command. It will produce the same

*This file has version number v1.4i, last revised 2004/02/27.

strings as \vref except that it doesn't start with the \ref and except that it will produce the string that is saved in \reftextcurrent if label and reference fall onto the same page. By defining \reftextcurrent to produce "on this page" or something similar, we can avoid that

```
... see the example \vpageref{ex:foo} which shows ...
```

comes out as "... see the example which shows ...", which could be misleading.

You can put 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. 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

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

which will then come out as "... see the example above which shows ..." if we are still on the same page, but as "... see the example on the page before which shows ..." (or something similar depending on the settings of the \reftext..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.

2.1 Additions in 1998

\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 fig:a is your first figure in the sequence and 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]{%
  \vrefpagenum\firstnum{#1}%
  \vrefpagenum\secondnum{#2}%
  the definition%
  \ifthenelse{\equal\firstnum\secondnum}%
    {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 the are on different pages.

2.2 Additions in 2001

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.

`\vref*` `\vpageref*` `\vpagerefrange*` Since it is to late to change the default behaviour I've added star versions of the macros which do not add any space before the generated text (they do nevertheless remove space at the left).

2.3 Additions in 2002

\labelformat A reference via \ref produces by default the data associated with the corresponding \label command (typically a number); any additional formatting has to be provided by the user. If, for example, references to equations are always to be typeset as "equation (*number*)", one has to code "equation (\ref{*key*})". With \labelformat the varioref package offers a possibility to generate such frills automatically. The command takes two arguments: the first is the name of a counter and the second is its representation when referenced. This means that for a successful usage, one has to know the counter name being used for generating the label, though in practice this should not pose a problem. The current counter number is picked up as an argument. Here are two examples:

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

\Vref \Ref A side effect of using \labelformat is that, depending on the defined formatting, it becomes impossible to use \ref at the beginning of a sentence (if its replacement text starts with a lowercase letter). To overcome this problem varioref introduces the commands \Ref and \Vref that behave like \ref and \{vref except that they uppercase the first token of the generated string.

To make \Ref or \Vref work properly the very first token in the second argument of \labelformat has to be a simple ASCII letter, otherwise the capitalization will fail or worse, you will end up with some error messages. If you actually need something more complicated in this place (e.g., an accented letter) you have to explicitly surround it with braces, to identify the part that needs to be capitalized. For example, for figure references in the Hungarian language you might want to write \labelformat{figure}{{\'a}bra~\thefigure}.

\vpagerefnum 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 \vpagerefnum can be used. This macro hold the current page "number" when \vpageref and friends are executed. Thus, by defining, for example

```
\renewcommand\reftextfaceafter {on page~\thevpagerefnum}
\renewcommand\reftextfacebefore{on page~\thevpagerefnum}
\renewcommand\reftextafter     {on page~\thevpagerefnum}
\renewcommand\reftextbefore    {on page~\thevpagerefnum}
```

textual references can be suppressed.

3 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.) It also supports languages currently not in babel; check the section on options later on. You can also modify some or all of the strings by redefining

\reftextbefore \reftextfacebefore \reftextafter \reftextfaceafter the following commands. Backward references use \reftextbefore if the label is on the preceding page but invisible and \reftextfacebefore if it is one the facing page (i.e., if the current page number is odd). Similarily \reftextafter is used when the label comes on the next page but one has to turn the page and \reftextfaceafter if it is on the following but facing page.

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

`\reftextfaraway` Finally we have `\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 `\pageref` in its replacement text.

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

3.1 Additions in 1998

The commands `\vrefrange` 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.

`\reftextpagerange` 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}`".

`\reftextlabelrange` 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}`".

4 Options

As mentioned above the package supports all standard options offered by the Babel system to customize the strings produced. In addition it offers the option `draft` to turn error messages into warnings during development. The defaul `final` produces error message when a generated string falls onto a page boundary (see next section).

5 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.,

table 5 on the current ⟨*page break*⟩ page

would would result in an error, which needs to be resolved by the user by replacing the `\vref` command with an ordinary `\ref` just before the final run. This is not `\vrefwarning` 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

`\vrefshowerrors` 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 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.

`\fullref`