The zref-vario package[*]
Code documentation

Gustavo Barros[†]

2023-08-15

EXPERIMENTAL

Contents

[*] This file describes v0.1.9, released 2023-08-15.

[†] https://github.com/gusbrs/zref-vario

1 Initial setup

Start the DocStrip guards.

```
⟨*package⟩
```

Identify the internal prefix (LaTeX3 DocStrip convention).

```
⟨@@=zrefvario⟩
```

```
\NeedsTeXFormat{LaTeX2e}
```

Identify the package.

```
\ProvidesExplPackage {zref-vario} {2023-08-15} {0.1.9}
  {Extended LaTeX page cross-references with varioref and zref-clever}
```

Required packages. Note that zref-clever `v0.2.0-alpha`, from 2022-01-28, requires a kernel 2021-11-15 or newer.

```
\RequirePackage { zref-clever } [ 2022-01-28 ] % v0.2.0-alpha
\RequirePackage { varioref }
```

Also note that, contrary to zref-clever, zref-vario is incompatible with cleveref, because the latter patches varioref in ways that break zref-vario.

2 Options

\zvsetup

\zvsetup Provide `\zvsetup`.

```
\zvsetup{⟨options⟩}
```

```
\NewDocumentCommand \zvsetup { m }
  { \keys_set:nn { zref-vario/setup } {#1} }
```

(*End of definition for* `\zvsetup`.)

pageprop

```
\tl_new:N \l__zrefvario_pageprop_tl
\keys_define:nn { zref-vario/setup }
  {
    pageprop .code:n =
      {
        \zref@ifpropundefined {#1}
          { \msg_warning:nnn { zref-vario } { pageprop-not-declared } {#1} }
          {
            \zref@iflistcontainsprop { \ZREF@mainlist } {#1}
              { \tl_set:Nn \l__zrefvario_pageprop_tl {#1} }
              { \msg_warning:nnn { zref-vario } { pageprop-not-in-main } {#1} }
          }
      } ,
    pageprop .value_required:n = true ,
    pageprop .initial:n = { page } ,
  }
\AddToHook { begindocument }
  {
```

```
  \keys_define:nn { zref-vario/setup }
   {
    pageprop .code:n =
     {
      \msg_warning:nnn { zref-vario }
       { option-preamble-only } { pageprop }
     }
   }
 }
\msg_new:nnn { zref-vario } { option-preamble-only }
 { Option~'#1'~only~available~in~the~preamble~\msg_line_context:. }
\msg_new:nnn { zref-vario } { pageprop-not-declared }
 {
  Property~'#1'~not~declared~\msg_line_context:.~
  Using~default~'page'~property.
 }
\msg_new:nnn { zref-vario } { pageprop-not-in-main }
 {
  Property~'#1'~not~in~main~list~\msg_line_context:.~
  Using~default~'page'~property.
 }
```

3 Reference format

Provide some functions to define options names in a standard form.

__zrefvario_ref_opt_default_cs:nn

__zrefvario_ref_opt_default_cs:nn {⟨*option*⟩} {⟨*signature*⟩}

```
\cs_new:Npn \__zrefvario_ref_opt_default_cs:nn #1#2
 { __zrefvario_ref_opt_default_ #1 : #2 }
```

(*End of definition for* __zrefvario_ref_opt_default_cs:nn.)

__zrefvario_base_lang:n

__zrefvario_base_lang:n {⟨*language*⟩}

```
\cs_new:Npn \__zrefvario_base_lang:n #1
 {
  \zrefclever_language_if_declared:nTF {#1}
   { \tl_use:c { \zrefclever_language_varname:n {#1} } }
   { zv_unknown_language }
 }
\cs_generate_variant:Nn \__zrefvario_base_lang:n { e }
```

(*End of definition for* __zrefvario_base_lang:n.)

\l__zrefvario_ref_language_tl

An internal version of \l_zrefclever_ref_language_tl, since we need to retreive the current value of the variable in multiple places.

```
\tl_new:N \l__zrefvario_ref_language_tl
\tl_set:Nn \l__zrefvario_ref_language_tl { \l_zrefclever_ref_language_tl }
```

(*End of definition for* \l__zrefvario_ref_language_tl.)

__zrefvario_ref_opt_name_lang_tl:nn

__zrefvario_ref_opt_name_lang_tl:nn {⟨*lang*⟩} {⟨*option*⟩}

```
\cs_new:Npn \__zrefvario_ref_opt_name_lang_tl:nn #1#2
 { g__zrefvario_ref_opt_lang_ \__zrefvario_base_lang:n {#1} _ #2 _tl }
\cs_generate_variant:Nn \__zrefvario_ref_opt_name_lang_tl:nn { en }
```

(*End of definition for __zrefvario_ref_opt_name_lang_tl:nn.*)

__zrefvario_ref_opt_name_lang_cs:nnn

```
\__zrefvario_ref_opt_name_lang_cs:nnn {⟨lang⟩} {⟨option⟩} {⟨signature⟩}
```

```
\cs_new:Npn \__zrefvario_ref_opt_name_lang_cs:nnn #1#2#3
  { __zrefvario_ref_opt_lang_ \__zrefvario_base_lang:n {#1} _ #2 : #3 }
\cs_generate_variant:Nn \__zrefvario_ref_opt_name_lang_cs:nnn { enn }
```

(*End of definition for __zrefvario_ref_opt_name_lang_cs:nnn.*)

Retrieving options values.

__zrefvario_ref_opt_tl_get:nnN
__zrefvario_ref_opt_cs_get:nnnN

```
\__zrefvario_ref_opt_tl_get:nnN {⟨lang⟩} {⟨option⟩} {⟨tl var to set⟩}
\__zrefvario_ref_opt_cs_get:nnnN {⟨lang⟩} {⟨option⟩}
  {⟨cs signature⟩} {⟨cs to set⟩}
```

```
\cs_new_protected:Npn \__zrefvario_ref_opt_tl_get:nnN #1#2#3
  {
    \tl_if_exist:cTF
      { \__zrefvario_ref_opt_name_lang_tl:nn {#1} {#2} }
      {
        \tl_set_eq:Nc #3
          { \__zrefvario_ref_opt_name_lang_tl:nn {#1} {#2} }
      }
      {
        \tl_set_eq:Nc #3
          { \__zrefvario_ref_opt_name_lang_tl:nn { zv_unknown_language } {#2} }
      }
  }
\cs_generate_variant:Nn \__zrefvario_ref_opt_tl_get:nnN { xnN }
\cs_new_protected:Npn \__zrefvario_ref_opt_cs_get:nnnN #1#2#3#4
  {
    \cs_if_exist:cTF
      { \__zrefvario_ref_opt_name_lang_cs:nnn {#1} {#2} {#3} }
      {
        \cs_set_eq:Nc #4
          { \__zrefvario_ref_opt_name_lang_cs:nnn {#1} {#2} {#3} }
      }
      {
        \cs_if_exist:cTF
          { \__zrefvario_ref_opt_default_cs:nn {#2} {#3} }
          {
            \cs_set_eq:Nc #4
              { \__zrefvario_ref_opt_default_cs:nn {#2} {#3} }
          }
          {
            \cs_set_eq:Nc #4
              {
                \__zrefvario_ref_opt_name_lang_cs:nnn
                  { zv_unknown_language } {#2} {#3}
              }
          }
      }
  }
\cs_generate_variant:Nn \__zrefvario_ref_opt_cs_get:nnnN { xnnN }
```

(*End of definition for* `\__zrefvario_ref_opt_tl_get:nnN` *and* `\__zrefvario_ref_opt_cs_get:nnnN`*.*)

Handle the options set of the user commands, and add support for varioref optional arguments. All commands can receive the same set of options `\zcref` can, and these are processed the same way and “consumed” by any `\zcref` or `\zcpageref` calls. Besides those, `vother` is made available for `\zvpageref`, corresponding to its second optional argument, and `vcurrent` which is available to all commands except `\fullref`, corresponding to their first optional arguments. The distinction between `\l__zrefvario_reftextcurrent_tl` and `\l__zrefvario_vref_reftextcurrent_tl` is in their default/initial values. The default values for the optional arguments of `\vpageref`, `\vreffrange`, and `\vpagereffrange` are set to `\reftextcurrent`, while the default value of the optional argument of `\vref` is empty.

```
\tl_new:N \l__zrefvario_reftextcurrent_tl
\tl_set:Nn \l__zrefvario_reftextcurrent_tl { \reftextcurrent }
\tl_new:N \l__zrefvario_vref_reftextcurrent_tl
\tl_new:N \l__zrefvario_reftextother_tl
\keys_define:nn { zref-vario/current }
  {
    vcurrent .code:n =
      {
        \tl_set:Nn \l__zrefvario_reftextcurrent_tl {#1}
        \tl_set:Nn \l__zrefvario_vref_reftextcurrent_tl {#1}
      } ,
    vcurrent .value_required:n = true ,
  }
\keys_define:nn { zref-vario/other }
  {
    vother .tl_set:N = \l__zrefvario_reftextother_tl ,
    vother .value_required:n = true ,
  }
\keys_define:nn { }
  {
    zref-vario/ref .inherit:n =
      { zref-clever/reference } ,
    zref-vario/ref-oneoptarg .inherit:n =
      { zref-clever/reference , zref-vario/current } ,
    zref-vario/ref-twooptargs .inherit:n =
      { zref-clever/reference , zref-vario/current , zref-vario/other } ,
  }
```

`\zvLanguageSetup` User interface for setting “strings” (`reftext...` options) and “formats” (`...format` options) for each language.

```
\zvLanguageSetup{⟨language⟩}{⟨options⟩}
```

```
\tl_new:N \l__zrefvario_setup_language_tl
\seq_new:N \g__zrefvario_setup_languages_seq
\NewDocumentCommand \zvLanguageSetup { m m }
  {
    \AddToHook { begindocument/before }
      {
        \group_begin:
        \zrefclever_language_if_declared:nTF {#1}
          {
            \tl_set:Nn \l__zrefvario_setup_language_tl {#1}
            \keys_set:nn { zref-vario/langsetup } {#2}
            \seq_gput_right:Nx \g__zrefvario_setup_languages_seq
              { \__zrefvario_base_lang:n {#1} }
          }
          { \msg_warning:nnn { zref-vario } { unknown-language-setup } {#1} }
        \group_end:
      }
  }
\msg_new:nnn { zref-vario } { unknown-language-setup }
  {
    Language~’#1’~is~unknown~\msg_line_context:.~Can’t~set~it~up.~
    The~language~must~be~declared~for~’zref-clever’,~see~its~documentation.
  }
\@onlypreamble \zvLanguageSetup
```

(*End of definition for* \zvLanguageSetup*.*)

```
\cs_new_protected:Npn \__zrefvario_opt_tl_gset:Nn #1#2
  {
    \tl_if_exist:NF #1
      { \tl_new:N #1 }
    \tl_gset:Nn #1 {#2}
  }
\cs_generate_variant:Nn \__zrefvario_opt_tl_gset:Nn { cn }
```

```
\keys_define:nn { zref-vario/langsetup }
  {
    reftextfaceafter .code:n =
      {
        \__zrefvario_opt_tl_gset:cn
          {
            \__zrefvario_ref_opt_name_lang_tl:en
              { \l__zrefvario_setup_language_tl }
              { reftextfaceafter }
          } {#1}
      } ,
    reftextfacebefore .code:n =
      {
        \__zrefvario_opt_tl_gset:cn
          {
            \__zrefvario_ref_opt_name_lang_tl:en
              { \l__zrefvario_setup_language_tl }
              { reftextfacebefore }
          } {#1}
      } ,
    reftextafter .code:n =
      {
        \__zrefvario_opt_tl_gset:cn
          {
            \__zrefvario_ref_opt_name_lang_tl:en
              { \l__zrefvario_setup_language_tl }
              { reftextafter }
          } {#1}
      } ,
    reftextbefore .code:n =
      {
        \__zrefvario_opt_tl_gset:cn
          {
            \__zrefvario_ref_opt_name_lang_tl:en
              { \l__zrefvario_setup_language_tl }
              { reftextbefore }
          } {#1}
      } ,
    reftextcurrent .code:n =
      {
        \__zrefvario_opt_tl_gset:cn
          {
            \__zrefvario_ref_opt_name_lang_tl:en
              { \l__zrefvario_setup_language_tl }
              { reftextcurrent }
          } {#1}
      } ,
    reftextfaraway .code:n =
      {
        \cs_gset_nopar:cpn
          {
            \__zrefvario_ref_opt_name_lang_cs:enn
              { \l__zrefvario_setup_language_tl }
              { reftextfaraway } { n }
          }
          ##1
          {#1}
      } ,
    reftextpagerange .code:n =
      {
        \cs_gset_nopar:cpn
          {
            \__zrefvario_ref_opt_name_lang_cs:enn
              { \l__zrefvario_setup_language_tl }
              { reftextpagerange } { nn }
          }
          ##1##2
          {#1}
      } ,
    reftextlabelrange .code:n =
      {
        \cs_gset_nopar:cpn
          {
            \__zrefvario_ref_opt_name_lang_cs:enn
              { \l__zrefvario_setup_language_tl }
              { reftextlabelrange } { nn }
          }
          ##1##2
          {#1}
      } ,
    vrefformat .code:n =
      {
        \cs_gset_nopar:cpn
          {
```

```
          \__zrefvario_ref_opt_name_lang_cs:enn
            { \l__zrefvario_setup_language_tl }
            { vrefformat } { nn }
        }
        ##1##2
        {#1}
    } ,
  vrefrangeformat .code:n =
    {
      \cs_gset_nopar:cpn
        {
          \__zrefvario_ref_opt_name_lang_cs:enn
            { \l__zrefvario_setup_language_tl }
            { vrefrangeformat } { nnn }
        }
        ##1##2##3
        {#1}
    } ,
  fullrefformat .code:n =
    {
      \cs_gset_nopar:cpn
        {
          \__zrefvario_ref_opt_name_lang_cs:enn
            { \l__zrefvario_setup_language_tl }
            { fullrefformat } { n }
        }
        ##1
        {#1}
    } ,
}
```

Default reference formats for all languages. These will be used by `\__zrefvario_ref_opt_cs_get:nnnN`, unless a language specific setting for the same format has been set. `\zreftextfaraway` is defined so it can be called with different arguments in `\zvpageref` (where it is alone) and in `\zfullref` (where it is the second of the pair).

```
\cs_new_nopar:cpn
  { \__zrefvario_ref_opt_default_cs:nn { vrefformat } { nn } } #1#2
  {
    \zcref {#2} ~
    \zvpageref [ S=false, typeset=both, vcurrent={#1} ] {#2}
  }
\cs_new_nopar:cpn
  { \__zrefvario_ref_opt_default_cs:nn { vrefrangeformat } { nnn } } #1#2#3
  {
    \reftextlabelrange {#2} {#3} ~
    \zvpagerefrange [ S=false, typeset=both, vcurrent={#1} ] {#2} {#3}
  }
\cs_new_nopar:cpn
  { \__zrefvario_ref_opt_default_cs:nn { fullrefformat } { n } } #1
  {
    \zcref {#1} ~
    \zreftextfaraway [ S=false, typeset=both ] {#1}
  }
\NewDocumentCommand \zreftextfaraway { s O { } m }
```

```
  { \zref@wrapper@babel \__zrefvario_zreftextfaraway:nnn {#3} {#1} {#2} }
\cs_new_protected:Npn \__zrefvario_zreftextfaraway:nnn #1#2#3
  {
    \group_begin:
    \keys_set:nn { zref-vario/ref } {#3}
    \bool_if:nT {#2}
      {
        \keys_set:nn { zref-vario/ref } { nohyperref }
        \bool_set_false:N \l__zrefvario_hyperlink_bool
      }
    \tl_set:Nn \l__zrefvario_linklabel_tl {#1}
    \__zrefvario_varioref_setup:
    \reftextfaraway {#1}
    \group_end:
  }
```

Warnings and values for missing strings / unknown languages.

```
\group_begin:
\tl_set:Nn \l__zrefvario_setup_language_tl { zv_unknown_language }
\AddToHook { begindocument/before }
  {
    \keys_set:nn { zref-vario/langsetup }
      {
        reftextfaceafter =
          {
            \msg_warning:nnxx { zref-vario } { missing-string }
              { reftextfaceafter } { \l__zrefvario_ref_language_tl }
            \zref@default
          } ,
        reftextfacebefore =
          {
            \msg_warning:nnxx { zref-vario } { missing-string }
              { reftextfacebefore } { \l__zrefvario_ref_language_tl }
            \zref@default
          } ,
        reftextafter =
          {
            \msg_warning:nnxx { zref-vario } { missing-string }
              { reftextafter } { \l__zrefvario_ref_language_tl }
            \zref@default
          } ,
        reftextbefore =
          {
            \msg_warning:nnxx { zref-vario } { missing-string }
              { reftextbefore } { \l__zrefvario_ref_language_tl }
            \zref@default
          } ,
        reftextcurrent =
          {
            \msg_warning:nnxx { zref-vario } { missing-string }
              { reftextcurrent } { \l__zrefvario_ref_language_tl }
            \zref@default
          } ,
        reftextfaraway =
          {
            \msg_warning:nnxx { zref-vario } { missing-string }
              { reftextfaraway } { \l__zrefvario_ref_language_tl }
            \zref@default
          } ,
        reftextpagerange =
          {
            \msg_warning:nnxx { zref-vario } { missing-string }
              { reftextpagerange } { \l__zrefvario_ref_language_tl }
            \zref@default
          } ,
        reftextlabelrange =
          {
            \msg_warning:nnxx { zref-vario } { missing-string }
              { reftextlabelrange } { \l__zrefvario_ref_language_tl }
            \zref@default
          } ,
      }
  }
\group_end:
\msg_new:nnn { zref-vario } { missing-string }
  { Missing~’#1’~string~for~language~’#2’~\msg_line_context:. }
```

A general initial warning in case the language is altogether not set up.

```
\AddToHook { begindocument/before }
  {
    \AddToHook { begindocument }
      {
        \seq_if_in:NxF \g__zrefvario_setup_languages_seq
          { \__zrefvario_base_lang:e { \l__zrefvario_ref_language_tl } }
          {
            \msg_warning:nnx { zref-vario } { language-not-setup }
              { \l__zrefvario_ref_language_tl }
          }
      }
  }
\msg_new:nnn { zref-vario } { language-not-setup }
  {
    Language~’#1’~is~not~set~up~for~’zref-vario’.~
    See~documentation~for~’\iow_char:N\\zvLanguageSetup’.
  }
```

Provide proper hyperlinking for the “nearby” page references. hyperref makes `\vref` a *single* link, including both the standard and the page reference (see Ulrike Fischer’s answer and discussion in the comments at https://tex.stackexchange.com/a/655106). However, it does not provide the same treatment to `\vpageref`, where there’s no linking at all for nearby pages, to `\vrefrange`, `\vpagerefrange`, or `\vnameref`. In my view, this makes hyperref inconsistent. cleveref indeed discusses the issue in the “Known Bugs and Work-Arounds” section, mentioning two problems. The first, that the page reference does not honor the `nameinlink` option, which it is actually the same behavior we get from `\vnameref`. The second, that nearby page references are not included in the links or hyperlinked when by themselves. This latter problem may be considered undesirable, and even a “bug”, but at least cleveref’s behavior in this regard is consistent.

This is actually tricky and I think the best alternative here is to not even try to make a single link of both references, as hyperref does for `\vref`. But rather to offer ways to hyperlink the "nearby" page references separately, which is something that can work for paired references and for standalone page references just as well. `\zvhyperlink` is provided for such purpose, and is meant to be used either in the reference format settings of `\zvLanguageSetup`, or in the `vcurrent` and `vother` options of the reference commands. It only makes a hyperlink if used in one such place (technically, if `\l__zrefvario_linklabel_tl` is not empty), if `\l__zrefvario_hyperlink_bool` is true (to control for starred variants) and if hyperref and zref-hyperref are loaded, otherwise it passes on its argument untouched.

```
\bool_new:N \g__zrefvario_zref_hyperref_loaded_bool
\bool_new:N \l__zrefvario_hyperlink_bool
\AddToHook { package/zref-hyperref/after }
  {
    \AddToHook { package/hyperref/after }
      {
        \bool_gset_true:N \g__zrefvario_zref_hyperref_loaded_bool
        \bool_set_true:N \l__zrefvario_hyperlink_bool
      }
  }
\tl_new:N \l__zrefvario_linklabel_tl
```

`\zvhyperlink`

`\zvhyperlink{`⟨*text*⟩`}`

```
\NewDocumentCommand \zvhyperlink { m }
  { \__zrefvario_hyperlink:n {#1} }
\cs_new_protected:Npn \__zrefvario_hyperlink:n #1
  {
    \bool_lazy_all:nTF
      {
        { \g__zrefvario_zref_hyperref_loaded_bool }
        { \l__zrefvario_hyperlink_bool }
        { ! \tl_if_empty_p:V \l__zrefvario_linklabel_tl }
      }
      {
        \hyperlink
          { \zref@extractdefault { \l__zrefvario_linklabel_tl } { anchor } { } }
          {#1}
      }
      {#1}
  }
```

(*End of definition for* `\zvhyperlink`.)

4 User interface

`\__zrefvario_vref_pagenum:Nn`
`\__zrefvario_vrefpagenum:Nn`
`\__zrefvario_vref_label:n`

These three small functions are responsible for "connecting" varioref with zref, instead of with the standard referencing system. They are meant to locally replace their varioref counterparts: `\vref@pagenum`, `\vrefpagenum`, and `\vref@label`.

```
\cs_new_protected:Npn \__zrefvario_vref_pagenum:Nn #1#2
  {
    \exp_args:NNNo \exp_args:NNo \cs_set_nopar:Npn #1
```

```
      { \zref@extractdefault {#2} { \l__zrefvario_pageprop_tl } { ?? } }
  }
\cs_new_protected:Npn \__zrefvario_vrefpagenum:Nn #1#2
  {
    \zref@refused {#2}
    \__zrefvario_vref_pagenum:Nn #1 {#2}
  }
\cs_new_protected:Npn \__zrefvario_vref_label:n #1
  { \zref@labelbyprops {#1} { \l__zrefvario_pageprop_tl } }
```

(*End of definition for __zrefvario_vref_pagenum:Nn , __zrefvario_vrefpagenum:Nn , and __zrefvario_vref_label:n.*)

__zrefvario_varioref_setup:

Sets things up for varioref to work with zref/zref-clever. It is pretty straightforward, but it is the core of the package. \l__zrefvario_varioref_setup_bool ensures this setup runs only once in case of nested calls to the package's commands.

```
\bool_new:N \l__zrefvario_varioref_setup_bool
\cs_new_protected:Npn \__zrefvario_varioref_setup:
  {
    \bool_if:NF \l__zrefvario_varioref_setup_bool
      {
        \bool_set_true:N \l__zrefvario_varioref_setup_bool
        \@vrefhandlespacefalse
        \cs_set_eq:NN \vref@pagenum \__zrefvario_vref_pagenum:Nn
        \cs_set_eq:NN \vrefpagenum \__zrefvario_vrefpagenum:Nn
        \cs_set_eq:NN \vref@label \__zrefvario_vref_label:n
        \__zrefvario_ref_opt_tl_get:xnN
          { \l__zrefvario_ref_language_tl }
          { reftextfaceafter }
          \reftextfaceafter
        \__zrefvario_ref_opt_tl_get:xnN
          { \l__zrefvario_ref_language_tl }
          { reftextfacebefore }
          \reftextfacebefore
        \__zrefvario_ref_opt_tl_get:xnN
          { \l__zrefvario_ref_language_tl }
          { reftextafter }
          \reftextafter
        \__zrefvario_ref_opt_tl_get:xnN
          { \l__zrefvario_ref_language_tl }
          { reftextbefore }
          \reftextbefore
        \__zrefvario_ref_opt_tl_get:xnN
          { \l__zrefvario_ref_language_tl }
          { reftextcurrent }
          \reftextcurrent
        \__zrefvario_ref_opt_cs_get:xnnN
          { \l__zrefvario_ref_language_tl }
          { reftextfaraway } { n }
          \reftextfaraway
        \__zrefvario_ref_opt_cs_get:xnnN
          { \l__zrefvario_ref_language_tl }
          { reftextpagerange } { nn }
          \reftextpagerange
        \__zrefvario_ref_opt_cs_get:xnnN
          { \l__zrefvario_ref_language_tl }
          { reftextlabelrange } { nn }
          \reftextlabelrange
        \__zrefvario_ref_opt_cs_get:xnnN
          { \l__zrefvario_ref_language_tl }
          { vrefformat } { nn }
          \vrefformat
        \__zrefvario_ref_opt_cs_get:xnnN
          { \l__zrefvario_ref_language_tl }
          { vrefrangeformat } { nnn }
          \vrefrangeformat
        \__zrefvario_ref_opt_cs_get:xnnN
          { \l__zrefvario_ref_language_tl }
          { fullrefformat } { n }
          \fullrefformat
      }
  }
```

(*End of definition for* `\__zrefvario_varioref_setup:`.)

User commands.

\zvref

`\zvref⟨*⟩[⟨options⟩]{⟨label⟩}`

```
\NewDocumentCommand \zvref { s O { } m }
 { \zref@wrapper@babel \__zrefvario_zvref:nnn {#3} {#1} {#2} }
\cs_new_protected:Npn \__zrefvario_zvref:nnn #1#2#3
 {
  \group_begin:
  \keys_set:nn { zref-vario/ref-oneoptarg } {#3}
  \bool_if:nT {#2}
   {
    \keys_set:nn { zref-vario/ref } { nohyperref }
    \bool_set_false:N \l__zrefvario_hyperlink_bool
   }
  \tl_set:Nn \l__zrefvario_linklabel_tl {#1}
  \__zrefvario_varioref_setup:
  \__zrefvario_vref:Vn \l__zrefvario_vref_reftextcurrent_tl {#1}
  \group_end:
 }
```

"Ideally", we'd be using `\vref` here, but hyperref redefines `\vref@star` and `\vr@f`, hard-coding what was supposed to be configurable…, so we use `\vrefformat` instead, which is the original definition of `\vref@star`, and `\vr@f` also boils down to `\vref@star` when `\@vrefhandlespace` is false, which is our case.

```
\cs_new_protected:Npn \__zrefvario_vref:nn #1#2
 { \vrefformat {#1} {#2} }
\cs_generate_variant:Nn \__zrefvario_vref:nn { Vn }
```

(*End of definition for* `\zvref`.)

\zvpageref

`\zvpageref⟨*⟩[⟨options⟩]{⟨label⟩}`

```
\NewDocumentCommand \zvpageref { s O { } m }
 { \zref@wrapper@babel \__zrefvario_zvpageref:nnn {#3} {#1} {#2} }
\cs_new_protected:Npn \__zrefvario_zvpageref:nnn #1#2#3
```

```
  {
    \group_begin:
    \keys_set:nn { zref-vario/ref-twooptargs } {#3}
    \bool_if:nT {#2}
      {
        \keys_set:nn { zref-vario/ref } { nohyperref }
        \bool_set_false:N \l__zrefvario_hyperlink_bool
      }
    \tl_set:Nn \l__zrefvario_linklabel_tl {#1}
    \__zrefvario_varioref_setup:
    \__zrefvario_vpageref:VVn \l__zrefvario_reftextcurrent_tl
      \l__zrefvario_reftextother_tl {#1}
    \bool_lazy_and:nnT
      { \g__zrefvario_zrefcheck_available_bool }
      { \l__zrefvario_with_vcheck_bool }
      {
        \zrefcheck_zrefvario_label:
        \vref@pagenum \l_tmpa_tl { \the\c@vrcnt @vr }
        \vrefpagenum \l_tmpb_tl {#1}
        \tl_if_eq:NNT \l_tmpa_tl \l_tmpb_tl
          {
            \zrefcheck_zrefvario_run_check_on_label:Vn
              \l__zrefvario_vcheck_tl {#1}
          }
      }
    \group_end:
  }
\cs_new_protected:Npn \__zrefvario_vpageref:nnn #1#2#3
  { \vpageref [{#1}] [#2] {#3} }
\cs_generate_variant:Nn \__zrefvario_vpageref:nnn { VVn }
```

(*End of definition for* \zvpageref*.*)

\zvrefrange

\zvrefrange⟨*⟩[⟨*options*⟩]{⟨*label*⟩}{⟨*label*⟩}

```
\NewDocumentCommand \zvrefrange { s O { } m m }
  {
    \zref@wrapper@babel
      { \zref@wrapper@babel \__zrefvario_zvrefrange:nnnn {#3} }
      {#4} {#1} {#2}
  }
\cs_new_protected:Npn \__zrefvario_zvrefrange:nnnn #1#2#3#4
  {
    \group_begin:
    \keys_set:nn { zref-vario/ref-oneoptarg } {#4}
    \bool_if:nT {#3}
      {
        \keys_set:nn { zref-vario/ref } { nohyperref }
        \bool_set_false:N \l__zrefvario_hyperlink_bool
      }
    \__zrefvario_varioref_setup:
    \vpagerefcompare {#1} {#2}
      { \tl_set:Nn \l__zrefvario_linklabel_tl {#1} } { }
    \__zrefvario_vrefrange:Vnn \l__zrefvario_reftextcurrent_tl {#1} {#2}
    \group_end:
  }
\cs_new_protected:Npn \__zrefvario_vrefrange:nnn #1#2#3
  { \vrefrange [{#1}] {#2} {#3} }
\cs_generate_variant:Nn \__zrefvario_vrefrange:nnn { Vnn }
```

(*End of definition for* `\zvrefrange`*.*)

`\zvpagerefrange`

`\zvpagerefrange`⟨*∗*⟩`[`⟨*options*⟩`]{`⟨*label*⟩`}{`⟨*label*⟩`}`

```
\NewDocumentCommand \zvpagerefrange { s O { } m m }
  {
    \zref@wrapper@babel
      { \zref@wrapper@babel \__zrefvario_zvpagerefrange:nnnn {#3} }
      {#4} {#1} {#2}
  }
\cs_new_protected:Npn \__zrefvario_zvpagerefrange:nnnn #1#2#3#4
  {
    \group_begin:
    \keys_set:nn { zref-vario/ref-oneoptarg } {#4}
    \bool_if:nT {#3}
      {
        \keys_set:nn { zref-vario/ref } { nohyperref }
        \bool_set_false:N \l__zrefvario_hyperlink_bool
      }
    \__zrefvario_varioref_setup:
    \vpagerefcompare {#1} {#2}
      { \tl_set:Nn \l__zrefvario_linklabel_tl {#1} } { }
    \__zrefvario_vpagerefrange:Vnn \l__zrefvario_reftextcurrent_tl {#1} {#2}
    \bool_lazy_and:nnT
      { \g__zrefvario_zrefcheck_available_bool }
      { \l__zrefvario_with_vcheck_bool }
      {
        \zrefcheck_zrefvario_label:
        \vrefpagenum \l_tmpa_tl {#1}
        \vrefpagenum \l_tmpb_tl {#2}
        \tl_if_eq:NNT \l_tmpa_tl \l_tmpb_tl
          {
            \vref@pagenum \l_tmpa_tl { \the\c@vrcnt @vr }
            \tl_if_eq:NNT \l_tmpa_tl \l_tmpb_tl
              {
                \zrefcheck_zrefvario_run_check_on_label:Vn
                  \l__zrefvario_vcheck_tl {#1}
                \zrefcheck_zrefvario_run_check_on_label:Vn
                  \l__zrefvario_vcheck_tl {#2}
              }
          }
      }
    \group_end:
  }
\cs_new_protected:Npn \__zrefvario_vpagerefrange:nnn #1#2#3
  { \vpagerefrange [{#1}] {#2} {#3} }
\cs_generate_variant:Nn \__zrefvario_vpagerefrange:nnn { Vnn }
```

(*End of definition for* `\zvpagerefrange`*.*)

\zfullref

\zfullref⟨*⟩[⟨*options*⟩]{⟨*label*⟩}

```
\NewDocumentCommand \zfullref { s O { } m }
  { \zref@wrapper@babel \__zrefvario_zfullref:nnn {#3} {#1} {#2} }
\cs_new_protected:Npn \__zrefvario_zfullref:nnn #1#2#3
  {
    \group_begin:
    \keys_set:nn { zref-vario/ref } {#3}
    \bool_if:nT {#2}
      {
        \keys_set:nn { zref-vario/ref } { nohyperref }
        \bool_set_false:N \l__zrefvario_hyperlink_bool
      }
    \tl_set:Nn \l__zrefvario_linklabel_tl {#1}
    \__zrefvario_varioref_setup:
    \fullref {#1}
    \group_end:
  }
```

(*End of definition for* \zfullref*.*)

5 zref-check integration

Integration with zref-check. vcheck, differently from check, receives only two values above and below, since these are the only two checks which make sense for zref-vario. This means that it also does not take \zcheck's options, but these can always be passed to check. Besides this, vcheck's check should only be done when label(s) and reference are in the same page, that is, when vcurrent is used. Furthermore, since varioref handles the case where the reference crosses page boundaries, we only set one label for the purpose, and this is done at the end of the reference, because that's the point varioref itself uses to check if label and reference are in the same page.

```
\bool_new:N \g__zrefvario_zrefcheck_available_bool
\bool_new:N \l__zrefvario_with_vcheck_bool
\tl_new:N \l__zrefvario_vcheck_tl
\AddToHook { begindocument/before }
  {
    \IfPackageLoadedTF { zref-check }
      {
        \IfPackageAtLeastTF { zref-check } { 2022-02-08 }
          {
            \bool_gset_true:N \g__zrefvario_zrefcheck_available_bool
            \keys_define:nn { zref-vario/current }
              {
                vcheck .choices:nn =
                  { above , below }
                  {
                    \bool_set_true:N \l__zrefvario_with_vcheck_bool
                    \tl_set:NV \l__zrefvario_vcheck_tl \l_keys_choice_tl
                  } ,
              }
          }
          {
            \bool_gset_false:N \g__zrefvario_zrefcheck_available_bool
            \keys_define:nn { zref-vario/current }
              {
                vcheck .code:n =
                  { \msg_warning:nn { zref-vario } { zref-check-unavailable } } ,
              }
          }
      }
      {
        \bool_gset_false:N \g__zrefvario_zrefcheck_available_bool
        \keys_define:nn { zref-vario/current }
          {
            vcheck .code:n =
              { \msg_warning:nn { zref-vario } { zref-check-unavailable } } ,
          }
      }
  }
\msg_new:nnn { zref-vario } { zref-check-unavailable }
  {
    Option~'vcheck'~is~only~available~when~'zref-check'~is~loaded~
    and~newer~than~'2022-02-08~v0.2.4'.
  }
```

6 Languages

Localization for the languages provided by the author, except where otherwise stated. However, I could only do so thanks to varioref, in particular for languages I don't really know. So, credits go to the contributors of varioref. Of course, if any native speaker would like to review these, or contribute new ones, it is much welcome. Note, however, that since zref-vario depends on zref-clever and piggybacks on its language infrastructure, it only makes sense to provide localization here for languages already supported by zref-clever.

```
\zvLanguageSetup { english }
  {
    reftextfaceafter =
      {on~the~\zvhyperlink{\reftextvario{facing}{next}~page}} ,
    reftextfacebefore =
      {on~the~\zvhyperlink{\reftextvario{facing}{preceding}~page}} ,
    reftextafter =
      {on~the~\zvhyperlink{\reftextvario{following}{next}~page}} ,
    reftextbefore =
      {on~the~\zvhyperlink{\reftextvario{preceding}{previous}~page}} ,
    reftextcurrent =
      {on~\reftextvario{}{the~}\zvhyperlink{\reftextvario{this}{current}~page}} ,
    reftextfaraway = {on~\zcpageref{#1}} ,
    reftextpagerange = {on~\zcpageref[range]{#1,#2}} ,
    reftextlabelrange = {\zcref[range]{#1,#2}} ,
  }
\zvLanguageSetup { german }
  {
    reftextfaceafter = {auf~der~\zvhyperlink{nächsten~Seite}} ,
    reftextfacebefore = {auf~der~\zvhyperlink{vorherigen~Seite}} ,
    reftextafter = {auf~der~\zvhyperlink{nächsten~Seite}} ,
    reftextbefore = {auf~der~\zvhyperlink{vorherigen~Seite}} ,
    reftextcurrent = {auf~\zvhyperlink{dieser~Seite}} ,
    reftextfaraway = {auf~\zcpageref[d=D,g=f]{#1}} ,
    reftextpagerange = {auf~den~\zcpageref[range,d=D,g=f]{#1,#2}} ,
    reftextlabelrange = {\zcref[range]{#1,#2}} ,
  }
\zvLanguageSetup { french }
  {
    reftextfaceafter =
      {\zvhyperlink{page~\reftextvario{ci-contre}{suivante}}} ,
    reftextfacebefore =
      {\zvhyperlink{page~\reftextvario{ci-contre}{précédente}}} ,
    reftextafter = {\zvhyperlink{page~suivante}} ,
    reftextbefore = {\zvhyperlink{page~précédente}} ,
    reftextcurrent = {de~la~\zvhyperlink{présente~page}} ,
    reftextfaraway = {\zcpageref[g=f]{#1}} ,
    reftextpagerange = {\zcpageref[range,g=f]{#1,#2}} ,
    reftextlabelrange = {\zcref[range]{#1,#2}} ,
  }
\zvLanguageSetup { portuguese }
  {
    reftextfaceafter =
      {na~\zvhyperlink{\reftextvario{página~oposta}{próxima~página}}} ,
    reftextfacebefore =
      {na~\zvhyperlink{página~\reftextvario{oposta}{anterior}}} ,
    reftextafter =
      {na~\zvhyperlink{\reftextvario{página~seguinte}{próxima~página}}} ,
    reftextbefore =
      {na~\zvhyperlink{página~\reftextvario{anterior}{precedente}}} ,
    reftextcurrent = {\zvhyperlink{nesta~página}} ,
    reftextfaraway = {na~\zcpageref[g=f]{#1}} ,
    reftextpagerange = {nas~\zcpageref[range,g=f]{#1,#2}} ,
    reftextlabelrange = {\zcref[range]{#1,#2}} ,
  }
\zvLanguageSetup { spanish }
  {
    reftextfaceafter = {en~la~\zvhyperlink{página~siguiente}} ,
    reftextfacebefore = {en~la~\zvhyperlink{página~anterior}} ,
    reftextafter = {en~la~\zvhyperlink{página~siguiente}} ,
    reftextbefore = {en~la~\zvhyperlink{página~anterior}} ,
    reftextcurrent = {en~\zvhyperlink{esta~página}} ,
    reftextfaraway = {en~la~\zcpageref[g=f]{#1}} ,
    reftextpagerange = {en~las~\zcpageref[range,g=f]{#1,#2}} ,
    reftextlabelrange = {\zcref[range]{#1,#2}} ,
  }
\zvLanguageSetup { dutch }
  {
    reftextfaceafter =
      {op~de~\zvhyperlink{\reftextvario{rechter~pagina}{pagina~hiernaast}}} ,
    reftextfacebefore =
      {op~de~\zvhyperlink{\reftextvario{linker~pagina}{pagina~hiernaast}}} ,
    reftextafter =
      {op~de~\zvhyperlink{\reftextvario{volgende~pagina}{pagina~hierna}}} ,
    reftextbefore =
      {op~de~\zvhyperlink{\reftextvario{vorige~pagina}{pagina~hiervoor}}} ,
    reftextcurrent = {op~\zvhyperlink{deze~pagina}} ,
    reftextfaraway = {op~\zcpageref[g=f]{#1}} ,
    reftextpagerange = {op~\zcpageref[range,g=f]{#1,#2}} ,
    reftextlabelrange = {\zcref[range]{#1,#2}} ,
  }
```

Italian localization improved thanks to Matteo Ferrigato (at issue #1)

```
\zvLanguageSetup { italian }
  {
    reftextfaceafter =
      {nella~\zvhyperlink{pagina~\reftextvario{a~fianco}{successiva}}} ,
    reftextfacebefore =
      {nella~\zvhyperlink{pagina~\reftextvario{a~fianco}{precedente}}} ,
    reftextafter =
      {nella~\zvhyperlink{pagina~\reftextvario{seguente}{successiva}}} ,
    reftextbefore = {nella~\zvhyperlink{pagina~precedente}} ,
    reftextcurrent = {in~\zvhyperlink{questa~pagina}} ,
    reftextfaraway = {a~\zcpageref[g=f]{#1}} ,
    reftextpagerange = {nelle~\zcpageref[range,g=f]{#1,#2}} ,
    reftextlabelrange = {\zcref[range]{#1,#2}} ,
  }
```

⟨/package⟩

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.