The zref-vario package
Code documentation

gusbrs

https://github.com/gusbrs/zref-vario
https://www.ctan.org/pkg/zref-vario

Version v0.1.10 – 2023-11-14

EXPERIMENTAL

Contents

1 Initial setup 2

2 Options 2

3 Reference format 3

4 User interface 12

5 zref-check integration 16

6 Languages 17

Index 19

1 Initial setup

Start the DocStrip guards.

```
⟨*package⟩
```

Identify the internal prefix (LaTeX3 DocStrip convention).

```
⟨@@=zrefvario⟩
```

```
\def\zrefvario@required@kernel{2023-11-01}
\NeedsTeXFormat{LaTeX2e}[\zrefvario@required@kernel]
\providecommand\IfFormatAtLeastTF{\@ifl@t@r\fmtversion}
\IfFormatAtLeastTF{\zrefvario@required@kernel}
  {}
  {%
    \PackageError{zref-vario}{LaTeX kernel too old}
      {%
        'zref-vario' requires a LaTeX kernel \zrefvario@required@kernel\space or newer.%
      }%
  }%
```

Identify the package.

```
\ProvidesExplPackage {zref-vario} {2023-11-14} {0.1.10}
  {Extended LaTeX page cross-references with varioref and zref-clever}
```

Required packages.

```
\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.

`\l__zrefvario_tmpa_tl` `\l__zrefvario_tmpb_tl` Temporary scratch variables.

```
\tl_new:N \l__zrefvario_tmpa_tl
\tl_new:N \l__zrefvario_tmpb_tl
```

(*End of definition for* `\l__zrefvario_tmpa_tl` *and* `\l__zrefvario_tmpb_tl`.)

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_tl_get:nnN {⟨lang⟩} {⟨option⟩} {⟨tl var to set⟩}`
`\__zrefvario_ref_opt_cs_get:nnnN` `\__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 { enN }
\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 { ennN }
```

(*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`, `\vrefrange`, and `\vpagerefrange` 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:Ne \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:nnee { zref-vario } { missing-string }
              { reftextfaceafter } { \l__zrefvario_ref_language_tl }
            \zref@default
          } ,
        reftextfacebefore =
          {
            \msg_warning:nnee { zref-vario } { missing-string }
              { reftextfacebefore } { \l__zrefvario_ref_language_tl }
            \zref@default
```

```
      } ,
    reftextafter =
      {
        \msg_warning:nnee { zref-vario } { missing-string }
          { reftextafter } { \l__zrefvario_ref_language_tl }
        \zref@default
      } ,
    reftextbefore =
      {
        \msg_warning:nnee { zref-vario } { missing-string }
          { reftextbefore } { \l__zrefvario_ref_language_tl }
        \zref@default
      } ,
    reftextcurrent =
      {
        \msg_warning:nnee { zref-vario } { missing-string }
          { reftextcurrent } { \l__zrefvario_ref_language_tl }
        \zref@default
      } ,
    reftextfaraway =
      {
        \msg_warning:nnee { zref-vario } { missing-string }
          { reftextfaraway } { \l__zrefvario_ref_language_tl }
        \zref@default
      } ,
    reftextpagerange =
      {
        \msg_warning:nnee { zref-vario } { missing-string }
          { reftextpagerange } { \l__zrefvario_ref_language_tl }
        \zref@default
      } ,
    reftextlabelrange =
      {
        \msg_warning:nnee { 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:NeF \g__zrefvario_setup_languages_seq
          { \__zrefvario_base_lang:e { \l__zrefvario_ref_language_tl } }
          {
            \msg_warning:nne { 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:enN
          { \l__zrefvario_ref_language_tl }
          { reftextfaceafter }
          \reftextfaceafter
        \__zrefvario_ref_opt_tl_get:enN
          { \l__zrefvario_ref_language_tl }
          { reftextfacebefore }
          \reftextfacebefore
        \__zrefvario_ref_opt_tl_get:enN
```

```
          { \l__zrefvario_ref_language_tl }
          { reftextafter }
          \reftextafter
        \__zrefvario_ref_opt_tl_get:enN
          { \l__zrefvario_ref_language_tl }
          { reftextbefore }
          \reftextbefore
        \__zrefvario_ref_opt_tl_get:enN
          { \l__zrefvario_ref_language_tl }
          { reftextcurrent }
          \reftextcurrent
        \__zrefvario_ref_opt_cs_get:ennN
          { \l__zrefvario_ref_language_tl }
          { reftextfaraway } { n }
          \reftextfaraway
        \__zrefvario_ref_opt_cs_get:ennN
          { \l__zrefvario_ref_language_tl }
          { reftextpagerange } { nn }
          \reftextpagerange
        \__zrefvario_ref_opt_cs_get:ennN
          { \l__zrefvario_ref_language_tl }
          { reftextlabelrange } { nn }
          \reftextlabelrange
        \__zrefvario_ref_opt_cs_get:ennN
          { \l__zrefvario_ref_language_tl }
          { vrefformat } { nn }
          \vrefformat
        \__zrefvario_ref_opt_cs_get:ennN
          { \l__zrefvario_ref_language_tl }
          { vrefrangeformat } { nnn }
          \vrefrangeformat
        \__zrefvario_ref_opt_cs_get:ennN
          { \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__zrefvario_tmpa_tl { \the\c@vrcnt @vr }
        \vrefpagenum \l__zrefvario_tmpb_tl {#1}
        \tl_if_eq:NNT \l__zrefvario_tmpa_tl \l__zrefvario_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__zrefvario_tmpa_tl {#1}
        \vrefpagenum \l__zrefvario_tmpb_tl {#2}
        \tl_if_eq:NNT \l__zrefvario_tmpa_tl \l__zrefvario_tmpb_tl
```

```
          {
            \vref@pagenum \l__zrefvario_tmpa_tl { \the\c@vrcnt @vr }
            \tl_if_eq:NNT \l__zrefvario_tmpa_tl \l__zrefvario_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.

A

\AddToHook 38, 148, 325, 382, 384, 401, 403, 633

B

bool commands:
\bool_gset_false:N 651, 660

\bool_gset_true:N 405, 639
\bool_if:NTF 442
\bool_if:nTF 313, 501, 520, 557, 581, 620
\bool_lazy_all:nTF 414
\bool_lazy_and:nnTF 529, 590
\bool_new:N . . . 399, 400, 439, 630, 631
\bool_set_false:N 316, 504, 523, 560, 584, 623
\bool_set_true:N 406, 444, 645

C

cs commands:
\cs_generate_variant:Nn 69, 74, 77, 91, 116, 174, 513, 546, 570, 613
\cs_gset_nopar:Npn 224, 235, 246, 257, 268, 279
\cs_if_exist:NTF 94, 101
\cs_new:Npn 61, 63, 72, 75
\cs_new_nopar:Npn 289, 295, 301
\cs_new_protected:Npn 78, 92, 168, 309, 412, 427, 432, 437, 440, 497, 511, 516, 544, 553, 568, 577, 611, 616
\cs_set_eq:NN 97, 104, 108, 446, 447, 448
\cs_set_nopar:Npn 429

D

\def 3

E

exp commands:
\exp_args:NNNo 429
\exp_args:NNo 429

F

\fmtversion 5
\fullref *5*, 627
\fullrefformat 492

G

group commands:
\group_begin: 150, 311, 323, 499, 518, 555, 579, 618
\group_end: 159, 321, 379, 509, 542, 566, 609, 628

H

\hyperlink 421

I

\IfFormatAtLeastTF 5, 6
\IfPackageAtLeastTF 637
\IfPackageLoadedTF 635
iow commands:
\iow_char:N 397

K

keys commands:
\l_keys_choice_tl 646
\keys_define:nn 23, 40, 121, 130, 135, 175, 640, 652, 661
\keys_set:nn 21, 154, 312, 315, 327, 500, 503, 519, 522, 556, 559, 580, 583, 619, 622

M

msg commands:
\msg_line_context: 50, 53, 58, 164, 381
\msg_new:nnn 49, 51, 56, 162, 380, 394, 668
\msg_warning:nn 655, 664
\msg_warning:nnn . . 28, 32, 44, 158, 389
\msg_warning:nnnn 331, 337, 343, 349, 355, 361, 367, 373

N

\NeedsTeXFormat 4
\NewDocumentCommand 20, 146, 307, 410, 495, 514, 547, 571, 614

P

\PackageError 9
\providecommand 5
\ProvidesExplPackage 14

R

\reftextafter 460
\reftextbefore 464
\reftextcurrent *5*, 118, 468
\reftextfaceafter 452
\reftextfacebefore 456
\reftextfaraway 320, 472
\reftextlabelrange 298, 480
\reftextpagerange 476
\reftextvario 676, 678, 680, 682, 684, 703, 705, 716, 718, 720, 722, 742, 744, 746, 748, 757, 759, 761
\RequirePackage 16, 17

S

seq commands:
\seq_gput_right:Nn 155
\seq_if_in:NnTF 386
\seq_new:N 145
\space 11

T

TEX and LATEX 2ε commands:
\@ifl@t@r 5
\@onlypreamble 167
\@vrefhandlespace *14*

\@vrefhandlespacefalse 445
\c@vrcnt 534, 599
\vr@f *14*
\vref@label *12*, 448
\vref@pagenum *12*, 446, 534, 599
\vref@star *14*
\zref@default 333, 339, 345, 351, 357, 363, 369, 375
\zref@extractdefault 422, 430
\zref@iflistcontainsprop 30
\zref@ifpropundefined 27
\zref@labelbyprops 438
\ZREF@mainlist 30
\zref@refused 434
\zref@wrapper@babel 308, 496, 515, 549, 550, 573, 574, 615
\zrefvario@required@kernel 3, 4, 6, 11
\the 534, 599
tl commands:
\tl_gset:Nn 172
\tl_if_empty_p:n 418
\tl_if_eq:NNTF 536, 597, 600
\tl_if_exist:NTF 80, 170
\tl_new:N 18, 19, 22, 70, 117, 119, 120, 144, 171, 409, 632
\tl_set:Nn 31, 71, 118, 125, 126, 153, 318, 324, 506, 525, 564, 588, 625, 646
\tl_set_eq:NN 83, 87
\tl_use:N 66

V

\vnameref *11*
\vpageref *5*, *11*, 545
\vpagerefcompare 563, 587
\vpagerefrange *5*, *11*, 612
\vref *5*, *11*, *14*
\vrefformat *14*, 484, 512
\vrefpagenum *12*, 447, 535, 595, 596
\vrefrange *5*, *11*, 569
\vrefrangeformat 488

Z

\zcheck *16*
\zcpageref *5*, 685, 686, 696, 697, 709, 710, 724, 725, 735, 736, 750, 751, 764, 765
\zcref *5*, 292, 304, 687, 698, 711, 726, 737, 752, 766
\zfullref *9*, *16*, 614
zrefcheck commands:
\zrefcheck_zrefvario_label: 533, 594
\zrefcheck_zrefvario_run_check_on_label:nn 538, 602, 604
zrefclever commands:
\zrefclever_language_if_declared:nTF 65, 151
\zrefclever_language_varname:n . . 66
\l_zrefclever_ref_language_tl . *4*, 71
\zreftextfaraway *8*, 305, 307
zrefvario internal commands:
__zrefvario_base_lang:n *3*, 63, 63, 69, 73, 76, 156, 387
__zrefvario_hyperlink:n . . . 411, 412
\l__zrefvario_hyperlink_bool *11*, 316, 400, 406, 417, 504, 523, 560, 584, 623
\l__zrefvario_linklabel_tl *11*, 318, 409, 418, 422, 506, 525, 564, 588, 625
__zrefvario_opt_tl_gset:Nn 168, 174, 179, 188, 197, 206, 215
\l__zrefvario_pageprop_tl 22, 31, 430, 438
\l__zrefvario_ref_language_tl . . . 70, 332, 338, 344, 350, 356, 362, 368, 374, 387, 390, 450, 454, 458, 462, 466, 470, 474, 478, 482, 486, 490
__zrefvario_ref_opt_cs_get:nnnN *4*, *8*, 78, 92, 116, 469, 473, 477, 481, 485, 489
__zrefvario_ref_opt_default_cs:nn *3*, 61, 61, 102, 105, 290, 296, 302
__zrefvario_ref_opt_name_lang_cs:nnn *4*, 75, 75, 77, 95, 98, 110, 226, 237, 248, 259, 270, 281
__zrefvario_ref_opt_name_lang_tl:nn *4*, 72, 72, 74, 81, 84, 88, 181, 190, 199, 208, 217
__zrefvario_ref_opt_tl_get:nnN . . *4*, 78, 78, 91, 449, 453, 457, 461, 465
\l__zrefvario_reftextcurrent_tl *5*, 117, 118, 125, 527, 565, 589
\l__zrefvario_reftextother_tl 120, 132, 528
\l__zrefvario_setup_language_tl 144, 153, 182, 191, 200, 209, 218, 227, 238, 249, 260, 271, 282, 324
\g__zrefvario_setup_languages_seq 145, 155, 386
\l__zrefvario_tmpa_tl 18, 534, 536, 595, 597, 599, 600
\l__zrefvario_tmpb_tl 18, 535, 536, 596, 597, 600
__zrefvario_varioref_setup: . . . 319, 439, 440, 507, 526, 562, 586, 626
\l__zrefvario_varioref_setup_bool *12*, 439, 442, 444

\l__zrefvario_vcheck_tl 539, 603, 605, 632, 646
__zrefvario_vpageref:nnn 527, 544, 546
__zrefvario_vpagerefrange:nnn 589, 611, 613
__zrefvario_vref:nn . . 508, 511, 513
__zrefvario_vref_label:n 427, 437, 448
__zrefvario_vref_pagenum:Nn 427, 427, 435, 446
\l__zrefvario_vref_reftextcurrent_tl 5, 119, 126, 508
__zrefvario_vrefpagenum:Nn 427, 432, 447
__zrefvario_vrefrange:nnn 565, 568, 570
\l__zrefvario_with_vcheck_bool 531, 592, 631, 645
__zrefvario_zfullref:nnn . . 615, 616
\g__zrefvario_zref_hyperref_loaded_bool 399, 405, 416
\g__zrefvario_zrefcheck_available_bool . . . 530, 591, 630, 639, 651, 660
__zrefvario_zreftextfaraway:nnn 308, 309
__zrefvario_zvpageref:nnn . 515, 516
__zrefvario_zvpagerefrange:nnnn 574, 577
__zrefvario_zvref:nnn 496, 497
__zrefvario_zvrefrange:nnnn 550, 553
\zvhyperlink *11*, 410, 676, 678, 680, 682, 684, 691, 692, 693, 694, 695, 703, 705, 706, 707, 708, 716, 718, 720, 722, 723, 730, 731, 732, 733, 734, 742, 744, 746, 748, 749, 757, 759, 761, 762, 763
\zvLanguageSetup *6*, *11*, 144, 673, 689, 700, 713, 728, 739, 754
\zvpageref *5*, *9*, *14*, 293, 514
\zvpagerefrange *15*, 299, 571
\zvref *13*, 495
\zvrefrange *14*, 547
\zvsetup *2*, 20