

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.

```
1  <*package>
    Identify the internal prefix (LATEX3 DocStrip convention).
2  <@=zrefvario>
3  \def\zrefvario@required@kernel{2023-11-01}
4  \NeedsTeXFormat{LaTeX2e}[\zrefvario@required@kernel]
5  \providecommand\IfFormatAtLeastTF{\@ifl@t@r\fmtversion}
6  \IfFormatAtLeastTF{\zrefvario@required@kernel}
7  {}
8  {%
9      \PackageError{zref-vario}{LaTeX kernel too old}
10     {%
11         'zref-vario' requires a LaTeX kernel \zrefvario@required@kernel\space or newer.%}
12     }%
13 }
```

Identify the package.

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

Required packages.

```
16 \RequirePackage { zref-clever } [ 2022-01-28 ] % v0.2.0-alpha
17 \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 Temporary scratch variables.

```
\l__zrefvario_tmpb_tl
18 \tl_new:N \l__zrefvario_tmpa_tl
19 \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>}
20 \NewDocumentCommand \zvsetup { m }
21   { \keys_set:nn { zref-vario/setup } {#1} }
```

(End of definition for \zvsetup.)

```

pageprop

22 \tl_new:N \l__zrefvario_pageprop_tl
23 \keys_define:nn { zref-vario/setup }
24 {
25   pageprop .code:n =
26   {
27     \zref@ifpropundefined {#1}
28     { \msg_warning:nnn { zref-vario } { pageprop-not-declared } {#1} }
29     {
30       \zref@iflistcontainsprop { \ZREF@mainlist } {#1}
31       { \tl_set:Nn \l__zrefvario_pageprop_tl {#1} }
32       { \msg_warning:nnn { zref-vario } { pageprop-not-in-main } {#1} }
33     }
34   },
35   pageprop .value_required:n = true ,
36   pageprop .initial:n = { page } ,
37 }
38 \AddToHook { begindocument }
39 {
40   \keys_define:nn { zref-vario/setup }
41   {
42     pageprop .code:n =
43     {
44       \msg_warning:nnn { zref-vario }
45       { option-preamble-only } { pageprop }
46     }
47   }
48 }
49 \msg_new:nnn { zref-vario } { option-preamble-only }
50 { Option-'#1'~only-available-in-the-preamble~\msg_line_context:.. }
51 \msg_new:nnn { zref-vario } { pageprop-not-declared }
52 {
53   Property-'#1'~not~declared~\msg_line_context:..~
54   Using~default~'page'~property.
55 }
56 \msg_new:nnn { zref-vario } { pageprop-not-in-main }
57 {
58   Property-'#1'~not~in~main~list~\msg_line_context:..~
59   Using~default~'page'~property.
60 }

```

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:mn {\langle option\rangle} {\langle signature\rangle}
61 \cs_new:Npn \__zrefvario_ref_opt_default_cs:nn #1#2
62 { __zrefvario_ref_opt_default_ #1 : #2 }

(End of definition for \__zrefvario_ref_opt_default_cs:nn.)

\__zrefvario_base_lang:n
\__zrefvario_base_lang:n {\langle language\rangle}
63 \cs_new:Npn \__zrefvario_base_lang:n #1

```

```

64   {
65     \zrefclever_language_if_declared:nTF {#1}
66     { \tl_use:c { \zrefclever_language_varname:n {#1} } }
67     { zv_unknown_language }
68   }
69 \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 retrieve the current value of the variable in multiple places.

```

70 \tl_new:N \l_zrefvario_ref_language_tl
71 \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⟩}
72 \cs_new:Npn \__zrefvario_ref_opt_name_lang_tl:nn #1#2
73   { g__zrefvario_ref_opt_lang_ \__zrefvario_base_lang:n {#1} _ #2 _tl }
74 \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⟩}
75 \cs_new:Npn \__zrefvario_ref_opt_name_lang_cs:nnn #1#2#3
76   { __zrefvario_ref_opt_lang_ \__zrefvario_base_lang:n {#1} _ #2 : #3 }
77 \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 {⟨lang⟩} {⟨option⟩} {⟨tl var to set⟩}
\__zrefvario_ref_opt_cs_get:nnnN {⟨lang⟩} {⟨option⟩}
  {⟨cs signature⟩} {⟨cs to set⟩}
78 \cs_new_protected:Npn \__zrefvario_ref_opt_tl_get:nnN #1#2#3
79   {
80     \tl_if_exist:cTF
81       { \__zrefvario_ref_opt_name_lang_tl:nn {#1} {#2} }
82     {
83       \tl_set_eq:Nc #3
84         { \__zrefvario_ref_opt_name_lang_tl:nn {#1} {#2} }
85     }
86     {
87       \tl_set_eq:Nc #3
88         { \__zrefvario_ref_opt_name_lang_tl:nn { zv_unknown_language } {#2} }
89     }
90   }
91 \cs_generate_variant:Nn \__zrefvario_ref_opt_tl_get:nnN { enN }
92 \cs_new_protected:Npn \__zrefvario_ref_opt_cs_get:nnnN #1#2#3#4
93   {
94     \cs_if_exist:cTF
95       { \__zrefvario_ref_opt_name_lang_cs:nnn {#1} {#2} {#3} }
96     {
97       \cs_set_eq:Nc #4

```

```

98         { \__zrefvario_ref_opt_name_lang_cs:nnn {#1} {#2} {#3} }
99     }
100    {
101        \cs_if_exist:cTF
102            { \__zrefvario_ref_opt_default_cs:nn {#2} {#3} }
103            {
104                \cs_set_eq:Nc #4
105                    { \__zrefvario_ref_opt_default_cs:nn {#2} {#3} }
106            }
107            {
108                \cs_set_eq:Nc #4
109                    {
110                        \__zrefvario_ref_opt_name_lang_cs:nnn
111                            { zv_unknown_language } {#2} {#3}
112                    }
113            }
114        }
115    }
116 \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 variorref 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.

```

117 \tl_new:N \l__zrefvario_reftextcurrent_tl
118 \tl_set:Nn \l__zrefvario_reftextcurrent_tl { \reftextcurrent }
119 \tl_new:N \l__zrefvario_vref_reftextcurrent_tl
120 \tl_new:N \l__zrefvario_reftextother_tl
121 \keys_define:nn { zref-vario/current }
122 {
123     vcurrent .code:n =
124     {
125         \tl_set:Nn \l__zrefvario_reftextcurrent_tl {#1}
126         \tl_set:Nn \l__zrefvario_vref_reftextcurrent_tl {#1}
127     },
128     vcurrent .value_required:n = true ,
129 }
130 \keys_define:nn { zref-vario/other }
131 {
132     vother .tl_set:N = \l__zrefvario_reftextother_tl ,
133     vother .value_required:n = true ,
134 }
135 \keys_define:nn { }
136 {
137     zref-vario/ref .inherit:n =
138         { zref-clever/reference } ,

```

```

139 zref-vario/ref-oneoptarg .inherit:n =
140   { zref-clever/reference , zref-vario/current } ,
141 zref-vario/ref-twooptargs .inherit:n =
142   { zref-clever/reference , zref-vario/current , zref-vario/other } ,
143 }

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

\zvLanguageSetup{<language>}{<options>}

144 \tl_new:N \l__zrefvario_setup_language_tl
145 \seq_new:N \g__zrefvario_setup_languages_seq
146 \NewDocumentCommand \zvLanguageSetup { m m }
147 {
148   \AddToHook { begindocument/before }
149   {
150     \group_begin:
151     \zrefclever_language_if_declared:nTF {#1}
152     {
153       \tl_set:Nn \l__zrefvario_setup_language_tl {#1}
154       \keys_set:nn { zref-vario/langsetup } {#2}
155       \seq_gput_right:Ne \g__zrefvario_setup_languages_seq
156         { \__zrefvario_base_lang:n {#1} }
157     }
158     { \msg_warning:nnn { zref-vario } { unknown-language-setup } {#1} }
159     \group_end:
160   }
161 }
162 \msg_new:nnn { zref-vario } { unknown-language-setup }
163 {
164   Language~'#1'~is~unknown~\msg_line_context:..~Can't~set~it~up.~
165   The~language~must~be~declared~for~'zref-clever',~see~its~documentation.
166 }
167 \onlypreamble \zvLanguageSetup

(End of definition for \zvLanguageSetup.)

168 \cs_new_protected:Npn \__zrefvario_opt_tl_gset:Nn #1#2
169 {
170   \tl_if_exist:NF #1
171   { \tl_new:N #1 }
172   \tl_gset:Nn #1 {#2}
173 }
174 \cs_generate_variant:Nn \__zrefvario_opt_tl_gset:Nn { cn }

175 \keys_define:nn { zref-vario/langsetup }
176 {
177   reftextfaceafter .code:n =
178   {
179     \__zrefvario_opt_tl_gset:cn
180     {
181       \__zrefvario_ref_opt_name_lang_tl:en
182         { \l__zrefvario_setup_language_tl }
183         { reftextfaceafter }
184     } {#1}
185 }

```

```

185     } ,
186     reftextfacebefore .code:n =
187     {
188         \__zrefvario_opt_tl_gset:cn
189         {
190             \__zrefvario_ref_opt_name_lang_tl:en
191             { \l__zrefvario_setup_language_tl }
192             { reftextfacebefore }
193             } {#1}
194         } ,
195     reftextafter .code:n =
196     {
197         \__zrefvario_opt_tl_gset:cn
198         {
199             \__zrefvario_ref_opt_name_lang_tl:en
200             { \l__zrefvario_setup_language_tl }
201             { reftextafter }
202             } {#1}
203         } ,
204     reftextbefore .code:n =
205     {
206         \__zrefvario_opt_tl_gset:cn
207         {
208             \__zrefvario_ref_opt_name_lang_tl:en
209             { \l__zrefvario_setup_language_tl }
210             { reftextbefore }
211             } {#1}
212         } ,
213     reftextcurrent .code:n =
214     {
215         \__zrefvario_opt_tl_gset:cn
216         {
217             \__zrefvario_ref_opt_name_lang_tl:en
218             { \l__zrefvario_setup_language_tl }
219             { reftextcurrent }
220             } {#1}
221         } ,
222     reftextfaraway .code:n =
223     {
224         \cs_gset_nopar:cpn
225         {
226             \__zrefvario_ref_opt_name_lang_cs:enn
227             { \l__zrefvario_setup_language_tl }
228             { reftextfaraway } { n }
229         }
230         ##1
231         {#1}
232     } ,
233     reftextpagerange .code:n =
234     {
235         \cs_gset_nopar:cpn
236         {
237             \__zrefvario_ref_opt_name_lang_cs:enn
238             { \l__zrefvario_setup_language_tl }

```

```

239         { reftextpagerange } { nn }
240     }
241     ##1##2
242     {#1}
243   } ,
244   reftextlabelrange .code:n =
245   {
246     \cs_gset_nopar:cpn
247     {
248       \__zrefvario_ref_opt_name_lang_cs:enn
249       { \l__zrefvario_setup_language_tl }
250       { reftextlabelrange } { nn }
251     }
252     ##1##2
253     {#1}
254   } ,
255   vrefformat .code:n =
256   {
257     \cs_gset_nopar:cpn
258     {
259       \__zrefvario_ref_opt_name_lang_cs:enn
260       { \l__zrefvario_setup_language_tl }
261       { vrefformat } { nn }
262     }
263     ##1##2
264     {#1}
265   } ,
266   vrefrangeformat .code:n =
267   {
268     \cs_gset_nopar:cpn
269     {
270       \__zrefvario_ref_opt_name_lang_cs:enn
271       { \l__zrefvario_setup_language_tl }
272       { vrefrangeformat } { nnn }
273     }
274     ##1##2##3
275     {#1}
276   } ,
277   fullrefformat .code:n =
278   {
279     \cs_gset_nopar:cpn
280     {
281       \__zrefvario_ref_opt_name_lang_cs:enn
282       { \l__zrefvario_setup_language_tl }
283       { fullrefformat } { n }
284     }
285     ##1
286     {#1}
287   } ,
288 }

```

Default reference formats for all languages. These will be used by `__zrefvario_ref_opt_name_lang_cs:enn`, 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).

```

289 \cs_new_nopar:cpn
290   { \__zrefvario_ref_opt_default_cs:nn { vrefformat } { nn } } #1#2
291   {
292     \zcref {#2} ~
293     \zvpageref [ S=false, typeset=both, vcurrent={#1} ] {#2}
294   }
295 \cs_new_nopar:cpn
296   { \__zrefvario_ref_opt_default_cs:nn { vrefrangeformat } { nnn } } #1#2#3
297   {
298     \reftextlabelrange {#2} {#3} ~
299     \zvpagerefrange [ S=false, typeset=both, vcurrent={#1} ] {#2} {#3}
300   }
301 \cs_new_nopar:cpn
302   { \__zrefvario_ref_opt_default_cs:nn { fullrefformat } { n } } #1
303   {
304     \zcref {#1} ~
305     \zreftextfaraway [ S=false, typeset=both ] {#1}
306   }
307 \NewDocumentCommand \zreftextfaraway { s O { } m }
308   { \zref@wrapper@babel \__zrefvario_zreftextfaraway:nnn {#3} {#1} {#2} }
309 \cs_new_protected:Npn \__zrefvario_zreftextfaraway:nnn #1#2#3
310   {
311     \group_begin:
312     \keys_set:nn { zref-vario/ref } {#3}
313     \bool_if:nT {#2}
314     {
315       \keys_set:nn { zref-vario/ref } { nohyperref }
316       \bool_set_false:N \l__zrefvario_hyperlink_bool
317     }
318     \tl_set:Nn \l__zrefvario_linklabel_tl {#1}
319     \__zrefvario_varioref_setup:
320     \reftextfaraway {#1}
321     \group_end:
322   }

```

Warnings and values for missing strings / unknown languages.

```

323 \group_begin:
324 \tl_set:Nn \l__zrefvario_setup_language_tl { zv_unknown_language }
325 \AddToHook { begindocument/before }
326   {
327     \keys_set:nn { zref-vario/langsetup }
328     {
329       reftextfaceafter =
330       {
331         \msg_warning:nnee { zref-vario } { missing-string }
332           { reftextfaceafter } { \l__zrefvario_ref_language_tl }
333           \zref@default
334       } ,
335       reftextfacebefore =
336       {
337         \msg_warning:nnee { zref-vario } { missing-string }
338           { reftextfacebefore } { \l__zrefvario_ref_language_tl }
339           \zref@default

```

```

340     } ,
341     reftextafter =
342     {
343         \msg_warning:nnee { zref-vario } { missing-string }
344         { reftextafter } { \l__zrefvario_ref_language_tl }
345         \zref@default
346     } ,
347     reftextbefore =
348     {
349         \msg_warning:nnee { zref-vario } { missing-string }
350         { reftextbefore } { \l__zrefvario_ref_language_tl }
351         \zref@default
352     } ,
353     reftextcurrent =
354     {
355         \msg_warning:nnee { zref-vario } { missing-string }
356         { reftextcurrent } { \l__zrefvario_ref_language_tl }
357         \zref@default
358     } ,
359     reftextfaraway =
360     {
361         \msg_warning:nnee { zref-vario } { missing-string }
362         { reftextfaraway } { \l__zrefvario_ref_language_tl }
363         \zref@default
364     } ,
365     reftextpagerange =
366     {
367         \msg_warning:nnee { zref-vario } { missing-string }
368         { reftextpagerange } { \l__zrefvario_ref_language_tl }
369         \zref@default
370     } ,
371     reftextlabelrange =
372     {
373         \msg_warning:nnee { zref-vario } { missing-string }
374         { reftextlabelrange } { \l__zrefvario_ref_language_tl }
375         \zref@default
376     } ,
377 }
378 }
379 \group_end:
380 \msg_new:nnn { zref-vario } { missing-string }
381 { Missing`#1`-string-for-language`#2`-\msg_line_context:.. }

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

382 \AddToHook { begindocument/before }
383 {
384     \AddToHook { begindocument }
385     {
386         \seq_if_in:NeF \g__zrefvario_setup_languages_seq
387         { \l__zrefvario_base_lang:e { \l__zrefvario_ref_language_tl } }
388         {
389             \msg_warning:nne { zref-vario } { language-not-setup }
390             { \l__zrefvario_ref_language_tl }
391         }

```

```

392     }
393   }
394 \msg_new:nnn { zref-vario } { language-not-setup }
395   {
396     Language~'#1'~is~not~set~up~for~'zref-vario'.~
397     See~documentation~for~'\iow_char:N\\zvLanguageSetup'.
398   }

```

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.

```

399 \bool_new:N \g__zrefvario_zref_hyperref_loaded_bool
400 \bool_new:N \l__zrefvario_hyperlink_bool
401 \AddToHook { package/zref-hyperref/after }
402   {
403     \AddToHook { package/hyperref/after }
404     {
405       \bool_gset_true:N \g__zrefvario_zref_hyperref_loaded_bool
406       \bool_set_true:N \l__zrefvario_hyperlink_bool
407     }
408   }
409 \tl_new:N \l__zrefvario_linklabel_tl

\zvhyperlink      \zvhyperlink{<text>}

410 \NewDocumentCommand \zvhyperlink { m }
411   { \l__zrefvario_hyperlink:n {#1} }
412 \cs_new_protected:Npn \l__zrefvario_hyperlink:n #1
413   {
414     \bool_lazy_all:nTF
415     {
416       { \g__zrefvario_zref_hyperref_loaded_bool }
417       { \l__zrefvario_hyperlink_bool }
418       { ! \tl_if_empty_p:V \l__zrefvario_linklabel_tl }
419     }

```

```

420   {
421     \hyperlink
422       { \zref@extractdefault { \l_zrefvario_linklabel_tl } { anchor } { } }
423       {#1}
424   }
425   {#1}
426 }
```

(End of definition for `\zvhyperlink`.)

4 User interface

`_zrefvario_vref_pagenum:Nn`, `_zrefvario_vrefpagenum:Nn`, and `_zrefvario_vref_label:n` These three small functions are responsible for “connecting” variorref with zref, instead of with the standard referencing system. They are meant to locally replace their variorref counterparts: `\vref@pagenum`, `\vrefpagenum`, and `\vref@label`.

```

427 \cs_new_protected:Npn \_zrefvario_vref_pagenum:Nn #1#2
428   {
429     \exp_args:NNNo \exp_args:NNo \cs_set_nopar:Npn #1
430       { \zref@extractdefault {#2} { \l_zrefvario_pageprop_tl } { ?? } }
431   }
432 \cs_new_protected:Npn \_zrefvario_vrefpagenum:Nn #1#2
433   {
434     \zref@refused {#2}
435     \_zrefvario_vref_pagenum:Nn #1 {#2}
436   }
437 \cs_new_protected:Npn \_zrefvario_vref_label:n #
438   { \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 variorref 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.

```

439 \bool_new:N \l_zrefvario_varioref_setup_bool
440 \cs_new_protected:Npn \_zrefvario_varioref_setup:
441   {
442     \bool_if:NF \l_zrefvario_varioref_setup_bool
443     {
444       \bool_set_true:N \l_zrefvario_varioref_setup_bool
445       \vrefhandlespacefalse
446       \cs_set_eq:NN \vref@pagenum \_zrefvario_vref_pagenum:Nn
447       \cs_set_eq:NN \vrefpagenum \_zrefvario_vrefpagenum:Nn
448       \cs_set_eq:NN \vref@label \_zrefvario_vref_label:n
449       \_zrefvario_ref_opt_tl_get:enN
450         { \l_zrefvario_ref_language_tl }
451         { reftextfaceafter }
452         \reftextfaceafter
453         \_zrefvario_ref_opt_tl_get:enN
454         { \l_zrefvario_ref_language_tl }
455         { reftextfacebefore }
456         \reftextfacebefore
457         \_zrefvario_ref_opt_tl_get:enN
```

```

458     { \l_zrefvario_ref_language_tl }
459     { reftextafter }
460     \reftextafter
461     \_zrefvario_ref_opt_tl_get:enN
462     { \l_zrefvario_ref_language_tl }
463     { reftextbefore }
464     \reftextbefore
465     \_zrefvario_ref_opt_tl_get:enN
466     { \l_zrefvario_ref_language_tl }
467     { reftextcurrent }
468     \reftextcurrent
469     \_zrefvario_ref_opt_cs_get:ennN
470     { \l_zrefvario_ref_language_tl }
471     { reftextfaraway } { n }
472     \reftextfaraway
473     \_zrefvario_ref_opt_cs_get:ennN
474     { \l_zrefvario_ref_language_tl }
475     { reftextpagerange } { nn }
476     \reftextpagerange
477     \_zrefvario_ref_opt_cs_get:ennN
478     { \l_zrefvario_ref_language_tl }
479     { reftextlabelrange } { nn }
480     \reftextlabelrange
481     \_zrefvario_ref_opt_cs_get:ennN
482     { \l_zrefvario_ref_language_tl }
483     { vrefformat } { nn }
484     \vrefformat
485     \_zrefvario_ref_opt_cs_get:ennN
486     { \l_zrefvario_ref_language_tl }
487     { vrefrangeformat } { nnn }
488     \vrefrangeformat
489     \_zrefvario_ref_opt_cs_get:ennN
490     { \l_zrefvario_ref_language_tl }
491     { fullrefformat } { n }
492     \fullrefformat
493   }
494 }

```

(End of definition for `_zrefvario_varioref_setup`.)

User commands.

```

\zvref      \zvref(*)[<options>]{<label>}
495 \NewDocumentCommand \zvref { s O { } m }
496   { \zref@wrapper@babel \_zrefvario_zvref:nnn {#3} {#1} {#2} }
497 \cs_new_protected:Npn \_zrefvario_zvref:nnn #1#2#3
498   {
499     \group_begin:
500     \keys_set:nn { zref-vario/ref-oneoptarg } {#3}
501     \bool_if:nT {#2}
502     {
503       \keys_set:nn { zref-vario/ref } { nohyperref }
504       \bool_set_false:N \l_zrefvario_hyperlink_bool
505     }
506     \tl_set:Nn \l_zrefvario_linklabel_tl {#1}

```

```

507     \__zrefvario_varioref_setup:
508     \__zrefvario_vref:Vn \l__zrefvario_vref_reftextcurrent_tl {#1}
509     \group_end:
510 }

```

“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.

```

511 \cs_new_protected:Npn \__zrefvario_vref:nn #1#2
512   { \vrefformat {#1} {#2} }
513 \cs_generate_variant:Nn \__zrefvario_vref:nn { Vn }

```

(End of definition for `\zvref`.)

```

\zvpageref      \zvpageref(*)[<options>]{<label>}
514 \NewDocumentCommand \zvpageref { s O { } m }
515   { \zref@wrapper@babel \__zrefvario_zvpageref:nnn {#3} {#1} {#2} }
516 \cs_new_protected:Npn \__zrefvario_zvpageref:nnn #1#2#3
517   {
518     \group_begin:
519     \keys_set:nn { zref-vario/ref-twooptargs } {#3}
520     \bool_if:nT {#2}
521     {
522       \keys_set:nn { zref-vario/ref } { nohyperref }
523       \bool_set_false:N \l__zrefvario_hyperlink_bool
524     }
525     \tl_set:Nn \l__zrefvario_linklabel_tl {#1}
526     \__zrefvario_varioref_setup:
527     \__zrefvario_vpageref:VVn \l__zrefvario_reftextcurrent_tl
528       \l__zrefvario_reftextother_tl {#1}
529     \bool_lazy_and:nnT
530       { \g__zrefvario_zrefcheck_available_bool }
531       { \l__zrefvario_with_vcheck_bool }
532     {
533       \zrefcheck_zrefvario_label:
534       \vref@pagenum \l__zrefvario_tmpa_tl { \the\c@vrcnt @vr }
535       \vref@pagenum \l__zrefvario_tmpb_tl {#1}
536       \tl_if_eq:NNT \l__zrefvario_tmpa_tl \l__zrefvario_tmpb_tl
537       {
538         \zrefcheck_zrefvario_run_check_on_label:Vn
539           \l__zrefvario_vcheck_tl {#1}
540         }
541       }
542     \group_end:
543   }
544 \cs_new_protected:Npn \__zrefvario_vpageref:nnn #1#2#3
545   { \vpageref [{#1}] [#2] {#3} }
546 \cs_generate_variant:Nn \__zrefvario_vpageref:nnn { VVn }

```

(End of definition for `\zvpageref`.)

```

\zvrefrange    \zvrefrange(*)[<options>]{<label>}{<label>}
547 \NewDocumentCommand \zvrefrange { s O { } m m }

```

```

548 {
549   \zref@wrapper@babel
550   { \zref@wrapper@babel \__zrefvario_zvrefrange:nnnn {#3} }
551   {#4} {#1} {#2}
552 }
553 \cs_new_protected:Npn \__zrefvario_zvrefrange:nnnn #1#2#3#4
554 {
555   \group_begin:
556   \keys_set:nn { zref-vario/ref-oneoptarg } {#4}
557   \bool_if:nT {#3}
558   {
559     \keys_set:nn { zref-vario/ref } { nohyperref }
560     \bool_set_false:N \l__zrefvario_hyperlink_bool
561   }
562   \__zrefvario_varioref_setup:
563   \vpagerefcompare {#1} {#2}
564   { \tl_set:Nn \l__zrefvario_linklabel_tl {#1} } { }
565   \__zrefvario_vrefrange:Vnn \l__zrefvario_reftextcurrent_tl {#1} {#2}
566   \group_end:
567 }
568 \cs_new_protected:Npn \__zrefvario_vrefrange:nnn #1#2#3
569 { \vrefrange [{#1}] {#2} {#3} }
570 \cs_generate_variant:Nn \__zrefvario_vrefrange:nnn { Vnn }

(End of definition for \zvrefrange.)

```

```

\zpagerefrange \zpagerefrange<*>[<options>]{<label>}{<label>}
571 \NewDocumentCommand \zpagerefrange { s O { } m m }
572 {
573   \zref@wrapper@babel
574   { \zref@wrapper@babel \__zrefvario_zpagerefrange:nnnn {#3} }
575   {#4} {#1} {#2}
576 }
577 \cs_new_protected:Npn \__zrefvario_zpagerefrange:nnnn #1#2#3#4
578 {
579   \group_begin:
580   \keys_set:nn { zref-vario/ref-oneoptarg } {#4}
581   \bool_if:nT {#3}
582   {
583     \keys_set:nn { zref-vario/ref } { nohyperref }
584     \bool_set_false:N \l__zrefvario_hyperlink_bool
585   }
586   \__zrefvario_varioref_setup:
587   \vpagerefcompare {#1} {#2}
588   { \tl_set:Nn \l__zrefvario_linklabel_tl {#1} } { }
589   \__zrefvario_vpagerefrange:Vnn \l__zrefvario_reftextcurrent_tl {#1} {#2}
590   \bool_lazy_and:nnT
591   { \g__zrefvario_zrefcheck_available_bool }
592   { \l__zrefvario_with_vcheck_bool }
593   {
594     \zrefcheck_zrefvario_label:
595     \vrefpagenum \l__zrefvario_tmpa_tl {#1}
596     \vrefpagenum \l__zrefvario_tmpb_tl {#2}
597     \tl_if_eq:NNT \l__zrefvario_tmpa_tl \l__zrefvario_tmpb_tl

```

```

598   {
599     \vref@pagenum \l__zrefvario_tmpa_tl { \the\c@vrcnt @vr }
600     \tl_if_eq:NNT \l__zrefvario_tmpa_tl \l__zrefvario_tmpb_tl
601     {
602       \zrefcheck_zrefvario_run_check_on_label:Vn
603         \l__zrefvario_vcheck_tl {#1}
604       \zrefcheck_zrefvario_run_check_on_label:Vn
605         \l__zrefvario_vcheck_tl {#2}
606     }
607   }
608   \group_end:
609 }
610 \cs_new_protected:Npn \__zrefvario_vpagerefrange:nnn #1#2#3
611   { \vpagerefrange [{#1}] {#2} {#3} }
612 \cs_generate_variant:Nn \__zrefvario_vpagerefrange:nnn { Vnn }

(End of definition for \zvpagerefrange.)

```

```

\zfullref
  \zfullref(*)[<options>]{<label>}
614 \NewDocumentCommand \zfullref { s O { } m }
615   { \zref@wrapper@babel \__zrefvario_zfullref:nnn {#3} {#1} {#2} }
616 \cs_new_protected:Npn \__zrefvario_zfullref:nnn #1#2#3
617   {
618     \group_begin:
619     \keys_set:nn { zref-vario/ref } {#3}
620     \bool_if:nT {#2}
621     {
622       \keys_set:nn { zref-vario/ref } { nohyperref }
623       \bool_set_false:N \l__zrefvario_hyperlink_bool
624     }
625     \tl_set:Nn \l__zrefvario_linklabel_tl {#1}
626     \__zrefvario_varioref_setup:
627     \fullref {#1}
628     \group_end:
629   }

```

(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.

```

630 \bool_new:N \g__zrefvario_zrefcheck_available_bool
631 \bool_new:N \l__zrefvario_with_vcheck_bool
632 \tl_new:N \l__zrefvario_vcheck_tl

```

```

633 \AddToHook { begindocument/before }
634 {
635   \IfPackageLoadedTF { zref-check }
636   {
637     \IfPackageAtLeastTF { zref-check } { 2022-02-08 }
638     {
639       \bool_gset_true:N \g__zrefvario_zrefcheck_available_bool
640       \keys_define:nn { zref-vario/current }
641       {
642         vcheck .choices:nn =
643         { above , below }
644         {
645           \bool_set_true:N \l__zrefvario_with_vcheck_bool
646           \tl_set:NV \l__zrefvario_vcheck_tl \l_keys_choice_tl
647         } ,
648       }
649     }
650   {
651     \bool_gset_false:N \g__zrefvario_zrefcheck_available_bool
652     \keys_define:nn { zref-vario/current }
653     {
654       vcheck .code:n =
655       { \msg_warning:nn { zref-vario } { zref-check-unavailable } } ,
656     }
657   }
658 }
659 {
660   \bool_gset_false:N \g__zrefvario_zrefcheck_available_bool
661   \keys_define:nn { zref-vario/current }
662   {
663     vcheck .code:n =
664     { \msg_warning:nn { zref-vario } { zref-check-unavailable } } ,
665   }
666 }
667 }
668 \msg_new:nnn { zref-vario } { zref-check-unavailable }
669 {
670   Option~`vcheck'~is~only~available~when~`zref-check'~is~loaded~
671   and~newer~than~`2022-02-08-v0.2.4'.
672 }

```

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.

```

673 \zvLanguageSetup { english }
674 {

```

```

675 reftextfaceafter =
676   {on-the~\zvhyperlink{\reftextvario{facing}{next}~page}} ,
677 reftextfacebefore =
678   {on-the~\zvhyperlink{\reftextvario{facing}{preceding}~page}} ,
679 reftextafter =
680   {on-the~\zvhyperlink{\reftextvario{following}{next}~page}} ,
681 reftextbefore =
682   {on-the~\zvhyperlink{\reftextvario{preceding}{previous}~page}} ,
683 reftextcurrent =
684   {on~\reftextvario{}{the~}\zvhyperlink{\reftextvario{this}{current}~page}} ,
685 reftextfaraway = {on~\zcpageref[#1]} ,
686 reftextpagerange = {on~\zcpageref[range]{#1,#2}} ,
687 reftextlabelrange = {\zcref[range]{#1,#2}} ,
688 }

689 \zvLanguageSetup { german }
690 {
691   reftextfaceafter = {auf~der~\zvhyperlink{nächsten~Seite}} ,
692   reftextfacebefore = {auf~der~\zvhyperlink{vorherigen~Seite}} ,
693   reftextafter = {auf~der~\zvhyperlink{nächsten~Seite}} ,
694   reftextbefore = {auf~der~\zvhyperlink{vorherigen~Seite}} ,
695   reftextcurrent = {auf~\zvhyperlink{dieser~Seite}} ,
696   reftextfaraway = {auf~\zcpageref[d=D,g=f]{#1}} ,
697   reftextpagerange = {auf~den~\zcpageref[range,d=D,g=f]{#1,#2}} ,
698   reftextlabelrange = {\zcref[range]{#1,#2}} ,
699 }

700 \zvLanguageSetup { french }
701 {
702   reftextfaceafter =
703     {\zvhyperlink{page~\reftextvario{ci-contre}{suivante}}} ,
704   reftextfacebefore =
705     {\zvhyperlink{page~\reftextvario{ci-contre}{précédente}}} ,
706   reftextafter = {\zvhyperlink{page~suivante}} ,
707   reftextbefore = {\zvhyperlink{page~précédente}} ,
708   reftextcurrent = {de~la~\zvhyperlink{présente~page}} ,
709   reftextfaraway = {\zcpageref[g=f]{#1}} ,
710   reftextpagerange = {\zcpageref[range,g=f]{#1,#2}} ,
711   reftextlabelrange = {\zcref[range]{#1,#2}} ,
712 }

713 \zvLanguageSetup { portuguese }
714 {
715   reftextfaceafter =
716     {na~\zvhyperlink{\reftextvario{página~oposta}{próxima~página}}} ,
717   reftextfacebefore =
718     {na~\zvhyperlink{página~\reftextvario{oposta}{anterior}}} ,
719   reftextafter =
720     {na~\zvhyperlink{\reftextvario{página~seguinte}{próxima~página}}} ,
721   reftextbefore =
722     {na~\zvhyperlink{página~\reftextvario{anterior}{precedente}}} ,
723   reftextcurrent = {\zvhyperlink{nesta~página}} ,
724   reftextfaraway = {na~\zcpageref[g=f]{#1}} ,
725   reftextpagerange = {nas~\zcpageref[range,g=f]{#1,#2}} ,
726   reftextlabelrange = {\zcref[range]{#1,#2}} ,
727 }

```

```

728 \zvLanguageSetup { spanish }
729 {
730     reftextfaceafter = {en~la~\zvhyperlink{página-siguiente}} ,
731     reftextfacebefore = {en~la~\zvhyperlink{página-anterior}} ,
732     reftextafter = {en~la~\zvhyperlink{página-siguiente}} ,
733     reftextbefore = {en~la~\zvhyperlink{página-anterior}} ,
734     reftextcurrent = {en~\zvhyperlink{esta-página}} ,
735     reftextfaraway = {en~la~\zcpageref[g=f]{#1}} ,
736     reftextpagerange = {en~las~\zcpageref[range,g=f]{#1,#2}} ,
737     reftextlabelrange = {\zcref[range]{#1,#2}} ,
738 }
739 \zvLanguageSetup { dutch }
740 {
741     reftextfaceafter =
742         {op~de~\zvhyperlink{\reftextvario{rechter-pagina}{pagina-hiernaast}}} ,
743     reftextfacebefore =
744         {op~de~\zvhyperlink{\reftextvario{linker-pagina}{pagina-hiernaast}}} ,
745     reftextafter =
746         {op~de~\zvhyperlink{\reftextvario{volgende-pagina}{pagina-hiernaast}}} ,
747     reftextbefore =
748         {op~de~\zvhyperlink{\reftextvario{vorige-pagina}{pagina-hier voor}}} ,
749     reftextcurrent = {op~\zvhyperlink{deze-pagina}} ,
750     reftextfaraway = {op~\zcpageref[g=f]{#1}} ,
751     reftextpagerange = {op~\zcpageref[range,g=f]{#1,#2}} ,
752     reftextlabelrange = {\zcref[range]{#1,#2}} ,
753 }
754 Italian localization improved thanks to Matteo Ferrigato (at issue #1)
755 \zvLanguageSetup { italian }
756 {
757     reftextfaceafter =
758         {nella~\zvhyperlink{pagina~\reftextvario{a-fianco}{successiva}}} ,
759     reftextfacebefore =
760         {nella~\zvhyperlink{pagina~\reftextvario{a-fianco}{precedente}}} ,
761     reftextafter =
762         {nella~\zvhyperlink{pagina~\reftextvario{seguente}{successiva}}} ,
763     reftextbefore = {nella~\zvhyperlink{pagina~precedente}} ,
764     reftextcurrent = {in~\zvhyperlink{questa~pagina}} ,
765     reftextfaraway = {a~\zcpageref[g=f]{#1}} ,
766     reftextpagerange = {nelle~\zcpageref[range,g=f]{#1,#2}} ,
767     reftextlabelrange = {\zcref[range]{#1,#2}} ,
768 }
769 
```

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	B
\AddToHook	bool commands:
. 38, 148, 325, 382, 384, 401, 403, 633	\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
 \providetcommand 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 L^AT_EX 2 _{ϵ} commands:

\c@ifl@t@r 5
 \conlypreamble 167
 \c@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:nnN
 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

```