

The **zref-vario** package^{*}

Code documentation

Gustavo Barros[†]

2023-06-14

EXPERIMENTAL

Contents

| | | |
|----------|-------------------------------|-----------|
| 1 | Initial setup | 2 |
| 2 | Options | 2 |
| 3 | Reference format | 3 |
| 4 | User interface | 11 |
| 5 | zref-check integration | 16 |
| 6 | Languages | 17 |
| | Index | 19 |

^{*}This file describes v0.1.8, released 2023-06-14.

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

1 Initial setup

Start the DocStrip guards.

```
1 <*package>
   Identify the internal prefix (LATEX3 DocStrip convention).
2 <@@=zrefvario>
3 \NeedsTeXFormat{LaTeX2e}
   Identify the package.
4 \ProvidesExplPackage {zref-vario} {2023-06-14} {0.1.8}
5 {Extended LaTeX page cross-references with varioref and zref-clever}
```

Required packages. The loading of `varioref` is delayed until `begindocument/before` since it has potential load order issues (e.g. with `hyperref`) so, ideally, it should be explicitly loaded by the user (and the manual recommends it). But we do it, if it hasn't been done.

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

```
6 \RequirePackage { zref-clever } [ 2022-01-28 ] % v0.2.0-alpha
7 \AddToHook { begindocument/before }
8 { \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>}
```

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

(End of definition for `\zvsetup`.)

pageprop

```
11 \tl_new:N \l__zrefvario_pageprop_tl
12 \keys_define:nn { zref-vario/setup }
13 {
14   pageprop .code:n =
15   {
16     \zref@ifpropundefined {#1}
17     { \msg_warning:nnn { zref-vario } { pageprop-not-declared } {#1} }
18     {
19       \zref@iflistcontainsprop { \ZREF@mainlist } {#1}
20       { \tl_set:Nn \l__zrefvario_pageprop_tl {#1} }
21       { \msg_warning:nnn { zref-vario } { pageprop-not-in-main } {#1} }
22     }
23   } ,
```

```

24     pageprop .value_required:n = true ,
25     pageprop .initial:n = { page } ,
26   }
27   \AddToHook { begindocument }
28   {
29     \keys_define:nn { zref-vario/setup }
30     {
31       pageprop .code:n =
32       {
33         \msg_warning:nnn { zref-vario }
34         { option-preamble-only } { pageprop }
35       }
36     }
37   }
38   \msg_new:nnn { zref-vario } { option-preamble-only }
39   { Option~'#1'~only~available~in~the~preamble~\msg_line_context:. }
40   \msg_new:nnn { zref-vario } { pageprop-not-declared }
41   {
42     Property~'#1'~not~declared~\msg_line_context:.~
43     Using~default~'page'~property.
44   }
45   \msg_new:nnn { zref-vario } { pageprop-not-in-main }
46   {
47     Property~'#1'~not~in~main~list~\msg_line_context:.~
48     Using~default~'page'~property.
49   }

```

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>}
50 \cs_new:Npn \__zrefvario_ref_opt_default_cs:nn #1#2
51   { __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>}
52 \cs_new:Npn \__zrefvario_base_lang:n #1
53   {
54     \zrefclever_language_if_declared:nTF {#1}
55     { \tl_use:c { \zrefclever_language_varname:n {#1} } }
56     { zv_unknown_language }
57   }
58 \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.

```

59 \tl_new:N \l_zrefvario_ref_language_tl
60 \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}}

61 \cs_new:Npn \__zrefvario_ref_opt_name_lang_tl:nn #1#2
62   { g__zrefvario_ref_opt_lang_ \__zrefvario_base_lang:n {#1} _ #2 _tl }
63 \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}}

64 \cs_new:Npn \__zrefvario_ref_opt_name_lang_cs:nnn #1#2#3
65   { __zrefvario_ref_opt_lang_ \__zrefvario_base_lang:n {#1} _ #2 : #3 }
66 \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}}
\__zrefvario_ref_opt_cs_get:nnnN      {\cs signature}} {\cs to set}}

67 \cs_new_protected:Npn \__zrefvario_ref_opt_tl_get:nnN #1#2#3
68   {
69     \tl_if_exist:cTF
70       { \__zrefvario_ref_opt_name_lang_tl:nn {#1} {#2} }
71       {
72         \tl_set_eq:Nc #3
73         { \__zrefvario_ref_opt_name_lang_tl:nn {#1} {#2} }
74       }
75       {
76         \tl_set_eq:Nc #3
77         { \__zrefvario_ref_opt_name_lang_tl:nn { zv_unknown_language } {#2} }
78       }
79   }
80 \cs_generate_variant:Nn \__zrefvario_ref_opt_tl_get:nnN { xnN }
81 \cs_new_protected:Npn \__zrefvario_ref_opt_cs_get:nnnN #1#2#3#4
82   {
83     \cs_if_exist:cTF
84       { \__zrefvario_ref_opt_name_lang_cs:nnn {#1} {#2} {#3} }
85       {
86         \cs_set_eq:Nc #4
87         { \__zrefvario_ref_opt_name_lang_cs:nnn {#1} {#2} {#3} }
88       }
89       {
90         \cs_if_exist:cTF
91           { \__zrefvario_ref_opt_default_cs:nn {#2} {#3} }
92           {
93             \cs_set_eq:Nc #4
94             { \__zrefvario_ref_opt_default_cs:nn {#2} {#3} }
95           }
96           {
97             \cs_set_eq:Nc #4
98             {
99               \__zrefvario_ref_opt_name_lang_cs:nnn
100               { zv_unknown_language } {#2} {#3}
101             }

```

```

102     }
103   }
104 }
105 \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`, `\vrefrange`, and `\vpagerefrange` are set to `\reftextcurrent`, while the default value of the optional argument of `\vref` is empty.

```

106 \tl_new:N \l__zrefvario_reftextcurrent_tl
107 \tl_set:Nn \l__zrefvario_reftextcurrent_tl { \reftextcurrent }
108 \tl_new:N \l__zrefvario_vref_reftextcurrent_tl
109 \tl_new:N \l__zrefvario_reftextother_tl
110 \keys_define:nn { zref-vario/current }
111 {
112   vcurrent .code:n =
113   {
114     \tl_set:Nn \l__zrefvario_reftextcurrent_tl {#1}
115     \tl_set:Nn \l__zrefvario_vref_reftextcurrent_tl {#1}
116   } ,
117   vcurrent .value_required:n = true ,
118 }
119 \keys_define:nn { zref-vario/other }
120 {
121   vother .tl_set:N = \l__zrefvario_reftextother_tl ,
122   vother .value_required:n = true ,
123 }
124 \keys_define:nn { }
125 {
126   zref-vario/ref .inherit:n =
127   { zref-clever/reference } ,
128   zref-vario/ref-oneoptarg .inherit:n =
129   { zref-clever/reference , zref-vario/current } ,
130   zref-vario/ref-twooptargs .inherit:n =
131   { zref-clever/reference , zref-vario/current , zref-vario/other } ,
132 }

```

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

```

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

133 \tl_new:N \l__zrefvario_setup_language_tl
134 \seq_new:N \g__zrefvario_setup_languages_seq
135 \NewDocumentCommand \zvLanguageSetup { m m }
136 {
137   \AddToHook { begindocument/before }

```

```

138 {
139   \group_begin:
140   \zrefclever_language_if_declared:nTF {#1}
141   {
142     \tl_set:Nn \l__zrefvario_setup_language_tl {#1}
143     \keys_set:nn { zref-vario/langsetup } {#2}
144     \seq_gput_right:Nx \g__zrefvario_setup_languages_seq
145       { \__zrefvario_base_lang:n {#1} }
146   }
147   { \msg_warning:nnn { zref-vario } { unknown-language-setup } {#1} }
148   \group_end:
149 }
150 }
151 \msg_new:nnn { zref-vario } { unknown-language-setup }
152 {
153   Language~'~{#1}'~is-unknown~\msg_line_context:..~Can't~set~it~up.~
154   The~language~must~be~declared~for~'zref-clever',~see~its~documentation.
155 }
156 \@onlypreamble \zvLanguageSetup

```

(End of definition for \zvLanguageSetup.)

```

157 \cs_new_protected:Npn \__zrefvario_opt_tl_gset:Nn #1#2
158 {
159   \tl_if_exist:NF #1
160   { \tl_new:N #1 }
161   \tl_gset:Nn #1 {#2}
162 }
163 \cs_generate_variant:Nn \__zrefvario_opt_tl_gset:Nn { cn }
164 \keys_define:nn { zref-vario/langsetup }
165 {
166   retextfaceafter .code:n =
167   {
168     \__zrefvario_opt_tl_gset:cn
169     {
170       \__zrefvario_ref_opt_name_lang_tl:en
171       { \l__zrefvario_setup_language_tl }
172       { retextfaceafter }
173     } {#1}
174   } ,
175   retextfacebefore .code:n =
176   {
177     \__zrefvario_opt_tl_gset:cn
178     {
179       \__zrefvario_ref_opt_name_lang_tl:en
180       { \l__zrefvario_setup_language_tl }
181       { retextfacebefore }
182     } {#1}
183   } ,
184   retextafter .code:n =
185   {
186     \__zrefvario_opt_tl_gset:cn
187     {
188       \__zrefvario_ref_opt_name_lang_tl:en
189       { \l__zrefvario_setup_language_tl }

```

```

190         { reftextafter }
191     } {#1}
192 },
193 reftextbefore .code:n =
194 {
195     \__zrefvario_opt_tl_gset:cn
196     {
197         \__zrefvario_ref_opt_name_lang_tl:en
198         { \l__zrefvario_setup_language_tl }
199         { reftextbefore }
200     } {#1}
201 },
202 reftextcurrent .code:n =
203 {
204     \__zrefvario_opt_tl_gset:cn
205     {
206         \__zrefvario_ref_opt_name_lang_tl:en
207         { \l__zrefvario_setup_language_tl }
208         { reftextcurrent }
209     } {#1}
210 },
211 reftextfaraway .code:n =
212 {
213     \cs_gset_nopar:cpn
214     {
215         \__zrefvario_ref_opt_name_lang_cs:enn
216         { \l__zrefvario_setup_language_tl }
217         { reftextfaraway } { n }
218     }
219     ##1
220     {#1}
221 },
222 reftextpagerange .code:n =
223 {
224     \cs_gset_nopar:cpn
225     {
226         \__zrefvario_ref_opt_name_lang_cs:enn
227         { \l__zrefvario_setup_language_tl }
228         { reftextpagerange } { nn }
229     }
230     ##1##2
231     {#1}
232 },
233 reftextlabelrange .code:n =
234 {
235     \cs_gset_nopar:cpn
236     {
237         \__zrefvario_ref_opt_name_lang_cs:enn
238         { \l__zrefvario_setup_language_tl }
239         { reftextlabelrange } { nn }
240     }
241     ##1##2
242     {#1}
243 },

```

```

244 vrefformat .code:n =
245 {
246   \cs_gset_nopar:cpn
247   {
248     \__zrefvario_ref_opt_name_lang_cs:enn
249     { \l__zrefvario_setup_language_tl }
250     { vrefformat } { nn }
251   }
252   ##1##2
253   {#1}
254 } ,
255 vrefrangeformat .code:n =
256 {
257   \cs_gset_nopar:cpn
258   {
259     \__zrefvario_ref_opt_name_lang_cs:enn
260     { \l__zrefvario_setup_language_tl }
261     { vrefrangeformat } { nnn }
262   }
263   ##1##2##3
264   {#1}
265 } ,
266 fullrefformat .code:n =
267 {
268   \cs_gset_nopar:cpn
269   {
270     \__zrefvario_ref_opt_name_lang_cs:enn
271     { \l__zrefvario_setup_language_tl }
272     { fullrefformat } { n }
273   }
274   ##1
275   {#1}
276 } ,
277 }

```

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).

```

278 \cs_new_nopar:cpn
279 { \__zrefvario_ref_opt_default_cs:nn { vrefformat } { nn } } #1#2
280 {
281   \zcref {#2} ~
282   \zvpageref [ S=false, typeset=both, vcurrent={#1} ] {#2}
283 }
284 \cs_new_nopar:cpn
285 { \__zrefvario_ref_opt_default_cs:nn { vrefrangeformat } { nnn } } #1#2#3
286 {
287   \reftextlabelrange {#2} {#3} ~
288   \zvpagerefrange [ S=false, typeset=both, vcurrent={#1} ] {#2} {#3}
289 }
290 \cs_new_nopar:cpn
291 { \__zrefvario_ref_opt_default_cs:nn { fullrefformat } { n } } #1
292 {

```



```

293 \zcref {#1} ~
294 \zreftextfaraway [ S=false, typeset=both ] {#1}
295 }
296 \NewDocumentCommand \zreftextfaraway { s O { } m }
297 { \zref@wrapper@babel \l__zrefvario_zreftextfaraway:nnn {#3} {#1} {#2} }
298 \cs_new_protected:Npn \l__zrefvario_zreftextfaraway:nnn #1#2#3
299 {
300   \group_begin:
301   \keys_set:nn { zref-vario/ref } {#3}
302   \bool_if:nT {#2}
303   {
304     \keys_set:nn { zref-vario/ref } { nohyperref }
305     \bool_set_false:N \l__zrefvario_hyperlink_bool
306   }
307   \tl_set:Nn \l__zrefvario_linklabel_tl {#1}
308   \l__zrefvario_varioref_setup:
309   \reftextfaraway {#1}
310   \group_end:
311 }

```

Warnings and values for missing strings / unknown languages.

```

312 \group_begin:
313 \tl_set:Nn \l__zrefvario_setup_language_tl { zv_unknown_language }
314 \AddToHook { begindocument/before }
315 {
316   \keys_set:nn { zref-vario/langsetup }
317   {
318     retextfaceafter =
319     {
320       \msg_warning:nxxx { zref-vario } { missing-string }
321       { retextfaceafter } { \l__zrefvario_ref_language_tl }
322       \zref@default
323     } ,
324     retextfacebefore =
325     {
326       \msg_warning:nxxx { zref-vario } { missing-string }
327       { retextfacebefore } { \l__zrefvario_ref_language_tl }
328       \zref@default
329     } ,
330     retextafter =
331     {
332       \msg_warning:nxxx { zref-vario } { missing-string }
333       { retextafter } { \l__zrefvario_ref_language_tl }
334       \zref@default
335     } ,
336     retextbefore =
337     {
338       \msg_warning:nxxx { zref-vario } { missing-string }
339       { retextbefore } { \l__zrefvario_ref_language_tl }
340       \zref@default
341     } ,
342     retextcurrent =
343     {
344       \msg_warning:nxxx { zref-vario } { missing-string }

```

```

345         { reftextcurrent } { \l__zrefvario_ref_language_tl }
346         \zref@default
347     } ,
348     reftextfaraway =
349     {
350         \msg_warning:nxxx { zref-vario } { missing-string }
351         { reftextfaraway } { \l__zrefvario_ref_language_tl }
352         \zref@default
353     } ,
354     reftextpagerange =
355     {
356         \msg_warning:nxxx { zref-vario } { missing-string }
357         { reftextpagerange } { \l__zrefvario_ref_language_tl }
358         \zref@default
359     } ,
360     reftextlabelrange =
361     {
362         \msg_warning:nxxx { zref-vario } { missing-string }
363         { reftextlabelrange } { \l__zrefvario_ref_language_tl }
364         \zref@default
365     } ,
366 }
367 }
368 \group_end:
369 \msg_new:nnn { zref-vario } { missing-string }
370 { Missing~'#1'~string~for~language~'#2'~\msg_line_context:. }
371
372 A general initial warning in case the language is altogether not set up.
373
374 \AddToHook { begindocument/before }
375 {
376     \AddToHook { begindocument }
377     {
378         \seq_if_in:NxF \g__zrefvario_setup_languages_seq
379         { \__zrefvario_base_lang:e { \l__zrefvario_ref_language_tl } }
380         {
381             \msg_warning:nxx { zref-vario } { language-not-setup }
382             { \l__zrefvario_ref_language_tl }
383         }
384     }
385 }
386
387 \msg_new:nnn { zref-vario } { language-not-setup }
388 {
389     Language~'#1'~is-not-set-up~for~'zref-vario'~
390     See~documentation~for~'\iow_char:N\zvLanguageSetup'.
391 }

```

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.

```

388 \bool_new:N \g__zrefvario_zref_hyperref_loaded_bool
389 \bool_new:N \l__zrefvario_hyperlink_bool
390 \AddToHook { package/zref-hyperref/after }
391 {
392   \AddToHook { package/hyperref/after }
393   {
394     \bool_gset_true:N \g__zrefvario_zref_hyperref_loaded_bool
395     \bool_set_true:N \l__zrefvario_hyperlink_bool
396   }
397 }
398 \tl_new:N \l__zrefvario_linklabel_tl

\zvhyperlink      \zvhyperlink{<text>}

399 \NewDocumentCommand \zvhyperlink { m }
400 { \__zrefvario_hyperlink:n {#1} }
401 \cs_new_protected:Npn \__zrefvario_hyperlink:n #1
402 {
403   \bool_lazy_all:nTF
404   {
405     { \g__zrefvario_zref_hyperref_loaded_bool }
406     { \l__zrefvario_hyperlink_bool }
407     { ! \tl_if_empty_p:V \l__zrefvario_linklabel_tl }
408   }
409   {
410     \hyperlink
411     { \zref@extractdefault { \l__zrefvario_linklabel_tl } { anchor } { } }
412     {#1}
413   }
414   {#1}
415 }

(End of definition for \zvhyperlink.)

```

4 User interface

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`, `\vrefpagemum`, and `\vref@label`.

```

416 \cs_new_protected:Npn \__zrefvario_vref_pagenum:Nn #1#2
417 {
418   \exp_args:NNNo \exp_args:NNo \cs_set_nopar:Npn #1
419   { \zref@extractdefault {#2} { \l__zrefvario_pageprop_tl } { ?? } }
420 }
421 \cs_new_protected:Npn \__zrefvario_vrefpagenum:Nn #1#2
422 {
423   \zref@refused {#2}
424   \__zrefvario_vref_pagenum:Nn #1 {#2}
425 }
426 \cs_new_protected:Npn \__zrefvario_vref_label:n #1
427 { \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.

```

428 \bool_new:N \l__zrefvario_varioref_setup_bool
429 \cs_new_protected:Npn \__zrefvario_varioref_setup:
430 {
431   \bool_if:NF \l__zrefvario_varioref_setup_bool
432   {
433     \bool_set_true:N \l__zrefvario_varioref_setup_bool
434     \@vrefhandlespacefalse
435     \cs_set_eq:NN \vref@pagenum \__zrefvario_vref_pagenum:Nn
436     \cs_set_eq:NN \vrefpagenum \__zrefvario_vrefpagenum:Nn
437     \cs_set_eq:NN \vref@label \__zrefvario_vref_label:n
438     \__zrefvario_ref_opt_tl_get:xnN
439     { \l__zrefvario_ref_language_tl }
440     { retextfaceafter }
441     \retextfaceafter
442     \__zrefvario_ref_opt_tl_get:xnN
443     { \l__zrefvario_ref_language_tl }
444     { retextfacebefore }
445     \retextfacebefore
446     \__zrefvario_ref_opt_tl_get:xnN
447     { \l__zrefvario_ref_language_tl }
448     { retextafter }
449     \retextafter
450     \__zrefvario_ref_opt_tl_get:xnN
451     { \l__zrefvario_ref_language_tl }
452     { retextbefore }
453     \retextbefore
454     \__zrefvario_ref_opt_tl_get:xnN
455     { \l__zrefvario_ref_language_tl }
456     { retextcurrent }
457     \retextcurrent
458     \__zrefvario_ref_opt_cs_get:xnnN
459     { \l__zrefvario_ref_language_tl }
460     { retextfaraway } { n }
461     \retextfaraway
462     \__zrefvario_ref_opt_cs_get:xnnN

```

```

463     { \l__zrefvario_ref_language_tl }
464     { reftextpagerange } { nn }
465     \reftextpagerange
466   \__zrefvario_ref_opt_cs_get:xnnN
467     { \l__zrefvario_ref_language_tl }
468     { reftextlabelrange } { nn }
469     \reftextlabelrange
470   \__zrefvario_ref_opt_cs_get:xnnN
471     { \l__zrefvario_ref_language_tl }
472     { vrefformat } { nn }
473     \vrefformat
474   \__zrefvario_ref_opt_cs_get:xnnN
475     { \l__zrefvario_ref_language_tl }
476     { vrefrangeformat } { nnn }
477     \vrefrangeformat
478   \__zrefvario_ref_opt_cs_get:xnnN
479     { \l__zrefvario_ref_language_tl }
480     { fullrefformat } { n }
481     \fullrefformat
482 }
483 }

```

(End of definition for __zrefvario_varioref_setup:.)

User commands.

```

\zvref      \zvref<*>[<options>]{<label>}
484 \NewDocumentCommand \zvref { s O { } m }
485   { \zref@wrapper@babel \__zrefvario_zvref:nnn {#3} {#1} {#2} }
486 \cs_new_protected:Npn \__zrefvario_zvref:nnn #1#2#3
487   {
488     \group_begin:
489     \keys_set:nn { zref-vario/ref-oneoptarg } {#3}
490     \bool_if:nT {#2}
491     {
492       \keys_set:nn { zref-vario/ref } { nohyperref }
493       \bool_set_false:N \l__zrefvario_hyperlink_bool
494     }
495     \tl_set:Nn \l__zrefvario_linklabel_tl {#1}
496     \__zrefvario_varioref_setup:
497     \__zrefvario_vref:Vn \l__zrefvario_vref_reftextcurrent_tl {#1}
498     \group_end:
499   }

```

“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.

```

500 \cs_new_protected:Npn \__zrefvario_vref:nn #1#2
501   { \vrefformat {#1} {#2} }
502 \cs_generate_variant:Nn \__zrefvario_vref:nn { Vn }

```

(End of definition for \zvref.)

```

\zvpageref          \zvpageref<*>[<options>]{<label>}
503 \NewDocumentCommand \zvpageref { s O { } m }
504 { \zref@wrapper@babel \__zrefvario_zvpageref:nnn {#3} {#1} {#2} }
505 \cs_new_protected:Npn \__zrefvario_zvpageref:nnn #1#2#3
506 {
507   \group_begin:
508   \keys_set:nn { zref-vario/ref-twooptargs } {#3}
509   \bool_if:nT {#2}
510   {
511     \keys_set:nn { zref-vario/ref } { nohyperref }
512     \bool_set_false:N \l__zrefvario_hyperlink_bool
513   }
514   \tl_set:Nn \l__zrefvario_linklabel_tl {#1}
515   \__zrefvario_varioref_setup:
516   \__zrefvario_vpageref:Vn \l__zrefvario_reftextcurrent_tl
517   \l__zrefvario_reftextother_tl {#1}
518   \bool_lazy_and:nnT
519   { \g__zrefvario_zrefcheck_available_bool }
520   { \l__zrefvario_with_vcheck_bool }
521   {
522     \zrefcheck_zrefvario_label:
523     \vref@pagenum \l_tmpa_tl { \the\c@vrcnt @vr }
524     \vref@pagenum \l_tmpb_tl {#1}
525     \tl_if_eq:NNT \l_tmpa_tl \l_tmpb_tl
526     {
527       \zrefcheck_zrefvario_run_check_on_label:Vn
528       \l__zrefvario_vcheck_tl {#1}
529     }
530   }
531   \group_end:
532 }
533 \cs_new_protected:Npn \__zrefvario_vpageref:nnn #1#2#3
534 { \vpageref [{#1}] [{#2}] {#3} }
535 \cs_generate_variant:Nn \__zrefvario_vpageref:nnn { VVn }

```

(End of definition for \zvpageref.)

```

\zvrefrange          \zvrefrange<*>[<options>]{<label>}{<label>}
536 \NewDocumentCommand \zvrefrange { s O { } m m }
537 {
538   \zref@wrapper@babel
539   { \zref@wrapper@babel \__zrefvario_zvrefrange:nnnn {#3} }
540   {#4} {#1} {#2}
541 }
542 \cs_new_protected:Npn \__zrefvario_zvrefrange:nnnn #1#2#3#4
543 {
544   \group_begin:
545   \keys_set:nn { zref-vario/ref-oneoptarg } {#4}
546   \bool_if:nT {#3}
547   {
548     \keys_set:nn { zref-vario/ref } { nohyperref }
549     \bool_set_false:N \l__zrefvario_hyperlink_bool
550   }
551   \__zrefvario_varioref_setup:

```

```

552 \vpagerefcompare {#1} {#2}
553 { \tl_set:Nn \l__zrefvario_linklabel_tl {#1} } { }
554 \__zrefvario_vrefrange:Vnn \l__zrefvario_reftextcurrent_tl {#1} {#2}
555 \group_end:
556 }
557 \cs_new_protected:Npn \__zrefvario_vrefrange:nnn #1#2#3
558 { \vrefrange [{#1}] {#2} {#3} }
559 \cs_generate_variant:Nn \__zrefvario_vrefrange:nnn { Vnn }

```

(End of definition for \zvrefrange.)

```

\zvpagerefrange \zvpagerefrange{*}[\options]{\label}{\label}
560 \NewDocumentCommand \zvpagerefrange { s O { } m m }
561 {
562 \zref@wrapper@babel
563 { \zref@wrapper@babel \__zrefvario_zvpagerefrange:nnnn {#3} }
564 {#4} {#1} {#2}
565 }
566 \cs_new_protected:Npn \__zrefvario_zvpagerefrange:nnnn #1#2#3#4
567 {
568 \group_begin:
569 \keys_set:nn { zref-vario/ref-oneoptarg } {#4}
570 \bool_if:nT {#3}
571 {
572 \keys_set:nn { zref-vario/ref } { nohyperref }
573 \bool_set_false:N \l__zrefvario_hyperlink_bool
574 }
575 \__zrefvario_varioref_setup:
576 \vpagerefcompare {#1} {#2}
577 { \tl_set:Nn \l__zrefvario_linklabel_tl {#1} } { }
578 \__zrefvario_vpagerefrange:Vnn \l__zrefvario_reftextcurrent_tl {#1} {#2}
579 \bool_lazy_and:nnT
580 { \g__zrefvario_zrefcheck_available_bool }
581 { \l__zrefvario_with_vcheck_bool }
582 {
583 \zrefcheck_zrefvario_label:
584 \vrefpagemum \l_tmpa_tl {#1}
585 \vrefpagemum \l_tmpb_tl {#2}
586 \tl_if_eq:NNT \l_tmpa_tl \l_tmpb_tl
587 {
588 \vref@pagemum \l_tmpa_tl { \the\c@vrcnt @vr }
589 \tl_if_eq:NNT \l_tmpa_tl \l_tmpb_tl
590 {
591 \zrefcheck_zrefvario_run_check_on_label:Vn
592 \l__zrefvario_vcheck_tl {#1}
593 \zrefcheck_zrefvario_run_check_on_label:Vn
594 \l__zrefvario_vcheck_tl {#2}
595 }
596 }
597 }
598 \group_end:
599 }
600 \cs_new_protected:Npn \__zrefvario_vpagerefrange:nnn #1#2#3
601 { \vpagerefrange [{#1}] {#2} {#3} }

```

```

602 \cs_generate_variant:Nn \__zrefvario_vpagerefrange:nnn { Vnn }
(End of definition for \zvpagerefrange.)

\zfullref      \zfullref{*}[\<options>]{\<label>}
603 \NewDocumentCommand \zfullref { s O { } m }
604 { \zref@wrapper@babel \__zrefvario_zfullref:nnn {#3} {#1} {#2} }
605 \cs_new_protected:Npn \__zrefvario_zfullref:nnn #1#2#3
606 {
607   \group_begin:
608   \keys_set:nn { zref-vario/ref } {#3}
609   \bool_if:nT {#2}
610   {
611     \keys_set:nn { zref-vario/ref } { nohyperref }
612     \bool_set_false:N \l__zrefvario_hyperlink_bool
613   }
614   \tl_set:Nn \l__zrefvario_linklabel_tl {#1}
615   \__zrefvario_varioref_setup:
616   \fullref {#1}
617   \group_end:
618 }
(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.

```

619 \bool_new:N \g__zrefvario_zrefcheck_available_bool
620 \bool_new:N \l__zrefvario_with_vcheck_bool
621 \tl_new:N \l__zrefvario_vcheck_tl
622 \AddToHook { begindocument/before }
623 {
624   \IfPackageLoadedTF { zref-check }
625   {
626     \IfPackageAtLeastTF { zref-check } { 2022-02-08 }
627     {
628       \bool_gset_true:N \g__zrefvario_zrefcheck_available_bool
629       \keys_define:nn { zref-vario/current }
630       {
631         vcheck .choices:nn =
632         { above , below }
633         {
634           \bool_set_true:N \l__zrefvario_with_vcheck_bool
635           \tl_set:NV \l__zrefvario_vcheck_tl \l_keys_choice_tl
636         } ,

```



```

637     }
638   }
639   {
640     \bool_gset_false:N \g__zrefvario_zrefcheck_available_bool
641     \keys_define:nn { zref-vario/current }
642     {
643       vcheck .code:n =
644         { \msg_warning:nn { zref-vario } { zref-check-unavailable } } ,
645     }
646   }
647 }
648 {
649   \bool_gset_false:N \g__zrefvario_zrefcheck_available_bool
650   \keys_define:nn { zref-vario/current }
651   {
652     vcheck .code:n =
653       { \msg_warning:nn { zref-vario } { zref-check-unavailable } } ,
654   }
655 }
656 }
657 \msg_new:nnn { zref-vario } { zref-check-unavailable }
658 {
659   Option~'vcheck'~is~only~available~when~'zref-check'~is~loaded~
660   and~newer~than~'2022-02-08~v0.2.4'.
661 }

```

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`.

```

662 \zvLanguageSetup { english }
663 {
664   reftextfaceafter =
665     {on~the~\zvhyperlink{\reftextvario{facing}{next}~page}} ,
666   reftextfacebefore =
667     {on~the~\zvhyperlink{\reftextvario{facing}{preceding}~page}} ,
668   reftextafter =
669     {on~the~\zvhyperlink{\reftextvario{following}{next}~page}} ,
670   reftextbefore =
671     {on~the~\zvhyperlink{\reftextvario{preceding}{previous}~page}} ,
672   reftextcurrent =
673     {on~\reftextvario{}{the~}\zvhyperlink{\reftextvario{this}{current}~page}} ,
674   reftextfaraway = {on~\zcpageref{#1}} ,
675   reftextpagerange = {on~\zcpageref[range]{#1,#2}} ,
676   reftextlabelrange = {\zcref[range]{#1,#2}} ,
677 }
678 \zvLanguageSetup { german }

```

```

679 {
680     retextfaceafter = {auf~der~\zvhyperlink{nächsten~Seite}} ,
681     retextfacebefore = {auf~der~\zvhyperlink{vorherigen~Seite}} ,
682     retextafter = {auf~der~\zvhyperlink{nächsten~Seite}} ,
683     retextbefore = {auf~der~\zvhyperlink{vorherigen~Seite}} ,
684     retextcurrent = {auf~\zvhyperlink{dieser~Seite}} ,
685     retextfaraway = {auf~\zcpageref[d=D,g=f]{#1}} ,
686     retextpagerange = {auf~den~\zcpageref[range,d=D,g=f]{#1,#2}} ,
687     retextlabelrange = {\zcref[range]{#1,#2}} ,
688 }
689 \zvLanguageSetup { french }
690 {
691     retextfaceafter =
692         {\zvhyperlink{page~\retextvario{ci-contre}{suivante}}} ,
693     retextfacebefore =
694         {\zvhyperlink{page~\retextvario{ci-contre}{précédente}}} ,
695     retextafter = {\zvhyperlink{page~suivante}} ,
696     retextbefore = {\zvhyperlink{page~précédente}} ,
697     retextcurrent = {de~la~\zvhyperlink{présente~page}} ,
698     retextfaraway = {\zcpageref[g=f]{#1}} ,
699     retextpagerange = {\zcpageref[range,g=f]{#1,#2}} ,
700     retextlabelrange = {\zcref[range]{#1,#2}} ,
701 }
702 \zvLanguageSetup { portuguese }
703 {
704     retextfaceafter =
705         {na~\zvhyperlink{\retextvario{página-oposta}{próxima-página}}} ,
706     retextfacebefore =
707         {na~\zvhyperlink{página~\retextvario{oposta}{anterior}}} ,
708     retextafter =
709         {na~\zvhyperlink{\retextvario{página-seguinte}{próxima-página}}} ,
710     retextbefore =
711         {na~\zvhyperlink{página~\retextvario{anterior}{precedente}}} ,
712     retextcurrent = {\zvhyperlink{nesta~página}} ,
713     retextfaraway = {na~\zcpageref[g=f]{#1}} ,
714     retextpagerange = {nas~\zcpageref[range,g=f]{#1,#2}} ,
715     retextlabelrange = {\zcref[range]{#1,#2}} ,
716 }
717 \zvLanguageSetup { spanish }
718 {
719     retextfaceafter = {en~la~\zvhyperlink{página-siguiente}} ,
720     retextfacebefore = {en~la~\zvhyperlink{página-anterior}} ,
721     retextafter = {en~la~\zvhyperlink{página-siguiente}} ,
722     retextbefore = {en~la~\zvhyperlink{página-anterior}} ,
723     retextcurrent = {en~\zvhyperlink{esta~página}} ,
724     retextfaraway = {en~la~\zcpageref[g=f]{#1}} ,
725     retextpagerange = {en~las~\zcpageref[range,g=f]{#1,#2}} ,
726     retextlabelrange = {\zcref[range]{#1,#2}} ,
727 }
728 \zvLanguageSetup { dutch }
729 {
730     retextfaceafter =
731         {op~de~\zvhyperlink{\retextvario{rechter~pagina}{pagina-hiernaast}}} ,

```

```

732 retextfacebefore =
733   {op~de~\zvhyperlink{\retextvario{linker~pagina}{pagina~hiernaast}}} ,
734 retextafter =
735   {op~de~\zvhyperlink{\retextvario{volgende~pagina}{pagina~hierna}}} ,
736 retextbefore =
737   {op~de~\zvhyperlink{\retextvario{vorige~pagina}{pagina~hiervoor}}} ,
738 retextcurrent = {op~\zvhyperlink{deze~pagina}} ,
739 retextfaraway = {op~\zcpageref[g=f]{#1}} ,
740 retextpagerange = {op~\zcpageref[range,g=f]{#1,#2}} ,
741 retextlabelrange = {\zceref[range]{#1,#2}} ,
742 }

```

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

```

743 \zvLanguageSetup { italian }
744 {
745   retextfaceafter =
746     {nella~\zvhyperlink{pagina~\retextvario{a~fianco}{successiva}}} ,
747   retextfacebefore =
748     {nella~\zvhyperlink{pagina~\retextvario{a~fianco}{precedente}}} ,
749   retextafter =
750     {nella~\zvhyperlink{pagina~\retextvario{seguente}{successiva}}} ,
751   retextbefore = {nella~\zvhyperlink{pagina~precedente}} ,
752   retextcurrent = {in~\zvhyperlink{questa~pagina}} ,
753   retextfaraway = {a~\zcpageref[g=f]{#1}} ,
754   retextpagerange = {nelle~\zcpageref[range,g=f]{#1,#2}} ,
755   retextlabelrange = {\zceref[range]{#1,#2}} ,
756 }
757 \end{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 | | C | |
|---------------------|--|-------------------------|--|
| \AddToHook | 7, 27, 137, 314, 371, 373, 390, 392, 622 | cs commands: | |
| | | \cs_generate_variant:Nn | 58, 63, 66, 80, 105, 163, 502, 535, 559, 602 |
| | | \cs_gset_nopar:Npn | 213, 224, 235, 246, 257, 268 |
| | | \cs_if_exist:NTF | 83, 90 |
| | | \cs_new:Npn | 50, 52, 61, 64 |
| | | \cs_new_nopar:Npn | 278, 284, 290 |
| | | \cs_new_protected:Npn | 67, 81, 157, 298, 401, 416, 421, 426, 429, 486, 500, 505, 533, 542, 557, 566, 600, 605 |
| | | \cs_set_eq:NN | 86, 93, 97, 435, 436, 437 |
| | | \cs_set_nopar:Npn | 418 |
| B | | E | |
| bool commands: | | exp commands: | |
| \bool_gset_false:N | 640, 649 | \exp_args:NNNo | 418 |
| \bool_gset_true:N | 394, 628 | | |
| \bool_if:NTF | 431 | | |
| \bool_if:nTF | 302, 490, 509, 546, 570, 609 | | |
| \bool_lazy_all:nTF | 403 | | |
| \bool_lazy_and:nnTF | 518, 579 | | |
| \bool_new:N | 388, 389, 428, 619, 620 | | |
| \bool_set_false:N | 305, 493, 512, 549, 573, 612 | | |
| \bool_set_true:N | 395, 433, 634 | | |

| | | | |
|---|--|---|--|
| <code>\exp_args:NNo</code> | 418 | <code>\reftextfaraway</code> | 309, 461 |
| F | | <code>\reftextlabelrange</code> | 287, 469 |
| <code>\fullref</code> | 5, 616 | <code>\reftextpagerange</code> | 465 |
| <code>\fullreformat</code> | 481 | <code>\reftextvario</code> | 665, 667, 669, 671, 673, 692, 694, 705, 707, 709, 711, 731, 733, 735, 737, 746, 748, 750 |
| G | | <code>\RequirePackage</code> | 6, 8 |
| group commands: | | | |
| <code>\group_begin:</code> | | S | |
| 139, 300, 312, 488, 507, 544, 568, 607 | | seq commands: | |
| <code>\group_end:</code> | | <code>\seq_gput_right:Nn</code> | 144 |
| 148, 310, 368, 498, 531, 555, 598, 617 | | <code>\seq_if_in:NnTF</code> | 375 |
| H | | <code>\seq_new:N</code> | 134 |
| <code>\hyperlink</code> | 410 | T | |
| I | | TeX and L ^A T _E X 2 _ε commands: | |
| <code>\IfPackageAtLeastTF</code> | 626 | <code>\@onlypreamble</code> | 156 |
| <code>\IfPackageLoadedTF</code> | 624 | <code>\@vrefhandlespace</code> | 13 |
| iow commands: | | <code>\@vrefhandlespacefalse</code> | 434 |
| <code>\iow_char:N</code> | 386 | <code>\c@vrcnt</code> | 523, 588 |
| K | | <code>\vr@f</code> | 13 |
| keys commands: | | <code>\vref@label</code> | 11, 437 |
| <code>\l_keys_choice_tl</code> | 635 | <code>\vref@pagenum</code> | 11, 435, 523, 588 |
| <code>\keys_define:nn</code> | 12, 29, 110, 119, 124, 164, 629, 641, 650 | <code>\vref@star</code> | 13 |
| <code>\keys_set:nn</code> | | <code>\zref@default</code> | |
| 10, 143, 301, 304, 316, 489, 492, 508, 511, 545, 548, 569, 572, 608, 611 | | 322, 328, 334, 340, 346, 352, 358, 364 | |
| M | | <code>\zref@extractdefault</code> | 411, 419 |
| msg commands: | | <code>\zref@iflistcontainsprop</code> | 19 |
| <code>\msg_line_context:</code> 39, 42, 47, 153, 370 | | <code>\zref@ifpropundefined</code> | 16 |
| <code>\msg_new:nnn</code> | | <code>\zref@labelbyprops</code> | 427 |
| 38, 40, 45, 151, 369, 383, 657 | | <code>\ZREF@mainlist</code> | 19 |
| <code>\msg_warning:nn</code> | 644, 653 | <code>\zref@refused</code> | 423 |
| <code>\msg_warning:nnn</code> .. | 17, 21, 33, 147, 378 | <code>\zref@wrapper@babel</code> | |
| <code>\msg_warning:nnnn</code> | | 297, 485, 504, 538, 539, 562, 563, 604 | |
| 320, 326, 332, 338, 344, 350, 356, 362 | | <code>\the</code> | 523, 588 |
| N | | tl commands: | |
| <code>\NeedsTeXFormat</code> | 3 | <code>\tl_gset:Nn</code> | 161 |
| <code>\NewDocumentCommand</code> | 9, 135, 296, 399, 484, 503, 536, 560, 603 | <code>\tl_if_empty_p:n</code> | 407 |
| P | | <code>\tl_if_eq:NNTF</code> | 525, 586, 589 |
| <code>\ProvidesExplPackage</code> | 4 | <code>\tl_if_exist:NTF</code> | 69, 159 |
| R | | <code>\tl_new:N</code> | 11, 59, 106, 108, 109, 133, 160, 398, 621 |
| <code>\reftextafter</code> | 449 | <code>\tl_set:Nn</code> 20, 60, 107, 114, 115, 142, 307, 313, 495, 514, 553, 577, 614, 635 | |
| <code>\reftextbefore</code> | 453 | <code>\tl_set_eq:NN</code> | 72, 76 |
| <code>\reftextcurrent</code> | 5, 107, 457 | <code>\tl_use:N</code> | 55 |
| <code>\reftextfaceafter</code> | 441 | <code>\l_tmpa_tl</code> . 523, 525, 584, 586, 588, 589 | |
| <code>\reftextfacebefore</code> | 445 | <code>\l_tmpb_tl</code> | 524, 525, 585, 586, 589 |
| | | V | |
| | | <code>\vnameref</code> | 10, 11 |
| | | <code>\vpageref</code> | 5, 10, 534 |
| | | <code>\vpagerefcompare</code> | 552, 576 |
| | | <code>\vpagerefrange</code> | 5, 10, 601 |

`\vref` 5, 10, 11, 13
`\vrefformat` 13, 473, 501
`\vrefpagenum` 11, 436, 524, 584, 585
`\vrefrange` 5, 10, 558
`\vrefrangeformat` 477

Z

`\zcheck` 16
`\zcpageref` 5, 674, 675, 685, 686, 698, 699,
 713, 714, 724, 725, 739, 740, 753, 754
`\zceref` 5, 281,
 293, 676, 687, 700, 715, 726, 741, 755
`\zfullref` 8, 16, 603
 zrefcheck commands:
 `\zrefcheck_zrefvario_label:` 522, 583
 `\zrefcheck_zrefvario_run_check_-`
 `on_label:nn` 527, 591, 593
 zrefclever commands:
 `\zrefclever_language_if_declared:nTF`
 54, 140
 `\zrefclever_language_varname:n` .. 55
 `\l_zrefclever_ref_language_tl` . 3, 60
`\zreftextfaraway` 8, 294, 296
 zrefvario internal commands:
 `__zrefvario_base_lang:n`
 3, 52, 52, 58, 62, 65, 145, 376
 `__zrefvario_hyperlink:n` 400, 401
 `\l_zrefvario_hyperlink_bool` ...
 11, 305,
 389, 395, 406, 493, 512, 549, 573, 612
 `\l_zrefvario_linklabel_tl`
 11, 307,
 398, 407, 411, 495, 514, 553, 577, 614
 `__zrefvario_opt_tl_gset:Nn`
 157, 163, 168, 177, 186, 195, 204
 `\l_zrefvario_pageprop_tl`
 11, 20, 419, 427
 `\l_zrefvario_ref_language_tl` ...
 . 59, 321, 327, 333, 339, 345, 351,
 357, 363, 376, 379, 439, 443, 447,
 451, 455, 459, 463, 467, 471, 475, 479
 `__zrefvario_ref_opt_cs_get:nnnN`
 4, 8,
 67, 81, 105, 458, 462, 466, 470, 474, 478
 `__zrefvario_ref_opt_default_-`
 `cs:nn` 3, 50, 50, 91, 94, 279, 285, 291
 `__zrefvario_ref_opt_name_lang_-`
 `cs:nnn` 4, 64, 64, 66,
 84, 87, 99, 215, 226, 237, 248, 259, 270
 `__zrefvario_ref_opt_name_lang_-`
 `tl:nn` 4, 61, 61,
 63, 70, 73, 77, 170, 179, 188, 197, 206
 `__zrefvario_ref_opt_tl_get:nnN` .
 . 4, 67, 67, 80, 438, 442, 446, 450, 454
 `\l__zrefvario_reftextcurrent_tl` .
 5, 106, 107, 114, 516, 554, 578
 `\l_zrefvario_reftextother_tl` ...
 109, 121, 517
 `\l_zrefvario_setup_language_tl` .
 133, 142, 171, 180, 189, 198,
 207, 216, 227, 238, 249, 260, 271, 313
 `\g_zrefvario_setup_languages_-`
 `seq` 134, 144, 375
 `__zrefvario_varioref_setup:` ...
 308, 428, 429, 496, 515, 551, 575, 615
 `\l_zrefvario_varioref_setup_-`
 `bool` 12, 428, 431, 433
 `\l_zrefvario_vcheck_tl`
 528, 592, 594, 621, 635
 `__zrefvario_vpageref:nnn`
 516, 533, 535
 `__zrefvario_vpagerefrange:nnn` ..
 578, 600, 602
 `__zrefvario_vref:nn` .. 497, 500, 502
 `__zrefvario_vref_label:n`
 416, 426, 437
 `__zrefvario_vref_pagenum:Nn` ...
 416, 416, 424, 435
 `\l_zrefvario_vref_reftextcurrent_-`
 `tl` 5, 108, 115, 497
 `__zrefvario_vrefpagenum:Nn`
 416, 421, 436
 `__zrefvario_vrefrange:nnn`
 554, 557, 559
 `\l_zrefvario_with_vcheck_bool` ..
 520, 581, 620, 634
 `__zrefvario_zfullref:nnn` .. 604, 605
 `\g_zrefvario_zref_hyperref_-`
 `loaded_bool` 388, 394, 405
 `\g_zrefvario_zrefcheck_available_-`
 `bool` ... 519, 580, 619, 628, 640, 649
 `__zrefvario_zreftextfaraway:nnn`
 297, 298
 `__zrefvario_zvpageref:nnn` . 504, 505
 `__zrefvario_zvpagerefrange:nnnn`
 563, 566
 `__zrefvario_zvref:nnn` 485, 486
 `__zrefvario_zvrefrange:nnnn` 539, 542
`\zvhyperlink` 11,
 399, 665, 667, 669, 671, 673, 680,
 681, 682, 683, 684, 692, 694, 695,
 696, 697, 705, 707, 709, 711, 712,
 719, 720, 721, 722, 723, 731, 733,
 735, 737, 738, 746, 748, 750, 751, 752
`\zvLanguageSetup` 5, 11,
 133, 662, 678, 689, 702, 717, 728, 743
`\zvpageref` 5, 8, 14, 282, 503
`\zvpagerefrange` 15, 288, 560

| | | | |
|--------------------------------|--|-----------------------------|---------------------------------------|
| <code>\zvref</code> | 13 , 484 | <code>\zvsetup</code> | 2 , 9 |
| <code>\zvrefrange</code> | 14 , 536 | | |