

The `zref-vario` package^{*}

Code documentation

Gustavo Barros[†]

2023-08-15

EXPERIMENTAL

Contents

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

^{*}This file describes v0.1.9, released 2023-08-15.

[†]<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} {2023-08-15} {0.1.9}
5   {Extended LaTeX page cross-references with varioref and zref-clever}
   Required packages. Note that zref-clever v0.2.0-alpha, from 2022-01-28, requires
   a kernel 2021-11-15 or newer.
6 \RequirePackage { zref-clever } [ 2022-01-28 ] % v0.2.0-alpha
7 \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>}
```

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

(End of definition for \zvsetup.)

`pageprop`

```
10 \tl_new:N \l__zrefvario_pageprop_tl
11 \keys_define:nn { zref-vario/setup }
12   {
13     pageprop .code:n =
14       {
15         \zref@ifpropundefined {#1}
16         { \msg_warning:nnn { zref-vario } { pageprop-not-declared } {#1} }
17         {
18           \zref@iflistcontainsprop { \ZREF@mainlist } {#1}
19           { \tl_set:Nn \l__zrefvario_pageprop_tl {#1} }
20           { \msg_warning:nnn { zref-vario } { pageprop-not-in-main } {#1} }
21         }
22       } ,
23     pageprop .value_required:n = true ,
24     pageprop .initial:n = { page } ,
25   }
26 \AddToHook { begindocument }
27   {
```

```

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

```

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

```

58 \tl_new:N \l_zrefvario_ref_language_tl
59 \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>}
60 \cs_new:Npn \__zrefvario_ref_opt_name_lang_tl:nn #1#2
61 { g__zrefvario_ref_opt_lang_ \__zrefvario_base_lang:n {#1} _ #2 _tl }
62 \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>}  
63 \cs_new:Npn \_zrefvario_ref_opt_name_lang_cs:nnn #1#2#3  
64   { \_zrefvario_ref_opt_lang_ \_zrefvario_base_lang:n {#1} _ #2 : #3 }  
65 \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>}  
66 \cs_new_protected:Npn \_zrefvario_ref_opt_tl_get:nnN #1#2#3  
67   {  
68     \tl_if_exist:cTF  
69     { \_zrefvario_ref_opt_name_lang_tl:nn {#1} {#2} }  
70     {  
71       \tl_set_eq:Nc #3  
72       { \_zrefvario_ref_opt_name_lang_tl:nn {#1} {#2} }  
73     }  
74     {  
75       \tl_set_eq:Nc #3  
76       { \_zrefvario_ref_opt_name_lang_tl:nn { zv_unknown_language } {#2} }  
77     }  
78   }  
79 \cs_generate_variant:Nn \_zrefvario_ref_opt_tl_get:nnN { xnN }  
80 \cs_new_protected:Npn \_zrefvario_ref_opt_cs_get:nnnN #1#2#3#4  
81   {  
82     \cs_if_exist:cTF  
83     { \_zrefvario_ref_opt_name_lang_cs:nnn {#1} {#2} {#3} }  
84     {  
85       \cs_set_eq:Nc #4  
86       { \_zrefvario_ref_opt_name_lang_cs:nnn {#1} {#2} {#3} }  
87     }  
88     {  
89       \cs_if_exist:cTF  
90       { \_zrefvario_ref_opt_default_cs:nn {#2} {#3} }  
91       {  
92         \cs_set_eq:Nc #4  
93         { \_zrefvario_ref_opt_default_cs:nn {#2} {#3} }  
94       }  
95       {  
96         \cs_set_eq:Nc #4  
97         {  
98           \_zrefvario_ref_opt_name_lang_cs:nnn  
99           { zv_unknown_language } {#2} {#3}  
100         }  
101       }  
102     }  
103   }  
104 \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.

```

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

```

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

```

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

132 \tl_new:N \l_zrefvario_setup_language_tl
133 \seq_new:N \g_zrefvario_setup_languages_seq
134 \NewDocumentCommand \zvLanguageSetup { m m }
135   {
136     \AddToHook { begindocument/before }
137     {
138       \group_begin:
139       \zrefclever_language_if_declared:nTF {#1}
140       {

```

```

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

```

(End of definition for \zvLanguageSetup.)

```

156 \cs_new_protected:Npn \__zrefvario_opt_tl_gset:Nn #1#2
157 {
158     \tl_if_exist:NF #1
159     { \tl_new:N #1 }
160     \tl_gset:Nn #1 {#2}
161 }
162 \cs_generate_variant:Nn \__zrefvario_opt_tl_gset:Nn { cn }
163 \keys_define:nn { zref-vario/langsetup }
164 {
165     reftextfaceafter .code:n =
166     {
167         \__zrefvario_opt_tl_gset:cn
168         {
169             \__zrefvario_ref_opt_name_lang_tl:en
170             { \l__zrefvario_setup_language_tl }
171             { reftextfaceafter }
172         } {#1}
173     } ,
174     reftextfacebefore .code:n =
175     {
176         \__zrefvario_opt_tl_gset:cn
177         {
178             \__zrefvario_ref_opt_name_lang_tl:en
179             { \l__zrefvario_setup_language_tl }
180             { reftextfacebefore }
181         } {#1}
182     } ,
183     reftextafter .code:n =
184     {
185         \__zrefvario_opt_tl_gset:cn
186         {
187             \__zrefvario_ref_opt_name_lang_tl:en
188             { \l__zrefvario_setup_language_tl }
189             { reftextafter }
190         } {#1}
191     } ,
192     reftextbefore .code:n =

```

```

193     {
194       \__zrefvario_opt_tl_gset:cn
195       {
196         \__zrefvario_ref_opt_name_lang_tl:en
197         { \l__zrefvario_setup_language_tl }
198         { reftextbefore }
199       } {#1}
200     } ,
201   reftextcurrent .code:n =
202     {
203       \__zrefvario_opt_tl_gset:cn
204       {
205         \__zrefvario_ref_opt_name_lang_tl:en
206         { \l__zrefvario_setup_language_tl }
207         { reftextcurrent }
208       } {#1}
209     } ,
210   reftextfaraway .code:n =
211     {
212       \cs_gset_nopar:cpn
213       {
214         \__zrefvario_ref_opt_name_lang_cs:enn
215         { \l__zrefvario_setup_language_tl }
216         { reftextfaraway } { n }
217       }
218       ##1
219       {#1}
220     } ,
221   reftextpagerange .code:n =
222     {
223       \cs_gset_nopar:cpn
224       {
225         \__zrefvario_ref_opt_name_lang_cs:enn
226         { \l__zrefvario_setup_language_tl }
227         { reftextpagerange } { nn }
228       }
229       ##1##2
230       {#1}
231     } ,
232   reftextlabelrange .code:n =
233     {
234       \cs_gset_nopar:cpn
235       {
236         \__zrefvario_ref_opt_name_lang_cs:enn
237         { \l__zrefvario_setup_language_tl }
238         { reftextlabelrange } { nn }
239       }
240       ##1##2
241       {#1}
242     } ,
243   vrefformat .code:n =
244     {
245       \cs_gset_nopar:cpn
246       {

```

```

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

```

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).

```

277 \cs_new_nopar:cpn
278 { \__zrefvario_ref_opt_default_cs:nn { vrefformat } { nn } } #1#2
279 {
280     \zcref {#2} ~
281     \zvpageref [ S=false, typeset=both, vcurrent={#1} ] {#2}
282 }
283 \cs_new_nopar:cpn
284 { \__zrefvario_ref_opt_default_cs:nn { vrefrangeformat } { nnn } } #1#2#3
285 {
286     \reftextlabelrange {#2} {#3} ~
287     \zvpagerefrange [ S=false, typeset=both, vcurrent={#1} ] {#2} {#3}
288 }
289 \cs_new_nopar:cpn
290 { \__zrefvario_ref_opt_default_cs:nn { fullrefformat } { n } } #1
291 {
292     \zcref {#1} ~
293     \zreftextfaraway [ S=false, typeset=both ] {#1}
294 }
295 \NewDocumentCommand \zreftextfaraway { s O { } m }

```



```

296 { \zref@wrapper@babel \__zrefvario_zreftextfaraway:nnn {#3} {#1} {#2} }
297 \cs_new_protected:Npn \__zrefvario_zreftextfaraway:nnn #1#2#3
298 {
299   \group_begin:
300   \keys_set:nn { zref-vario/ref } {#3}
301   \bool_if:nT {#2}
302   {
303     \keys_set:nn { zref-vario/ref } { nohyperref }
304     \bool_set_false:N \l__zrefvario_hyperlink_bool
305   }
306   \tl_set:Nn \l__zrefvario_linklabel_tl {#1}
307   \__zrefvario_varioref_setup:
308   \reftextfaraway {#1}
309   \group_end:
310 }

```

Warnings and values for missing strings / unknown languages.

```

311 \group_begin:
312 \tl_set:Nn \l__zrefvario_setup_language_tl { zv_unknown_language }
313 \AddToHook { begindocument/before }
314 {
315   \keys_set:nn { zref-vario/langsetup }
316   {
317     reftextfaceafter =
318     {
319       \msg_warning:nxxx { zref-vario } { missing-string }
320       { reftextfaceafter } { \l__zrefvario_ref_language_tl }
321       \zref@default
322     } ,
323     reftextfacebefore =
324     {
325       \msg_warning:nxxx { zref-vario } { missing-string }
326       { reftextfacebefore } { \l__zrefvario_ref_language_tl }
327       \zref@default
328     } ,
329     reftextafter =
330     {
331       \msg_warning:nxxx { zref-vario } { missing-string }
332       { reftextafter } { \l__zrefvario_ref_language_tl }
333       \zref@default
334     } ,
335     reftextbefore =
336     {
337       \msg_warning:nxxx { zref-vario } { missing-string }
338       { reftextbefore } { \l__zrefvario_ref_language_tl }
339       \zref@default
340     } ,
341     reftextcurrent =
342     {
343       \msg_warning:nxxx { zref-vario } { missing-string }
344       { reftextcurrent } { \l__zrefvario_ref_language_tl }
345       \zref@default
346     } ,
347     reftextfaraway =

```

```

348     {
349       \msg_warning:nxx { zref-vario } { missing-string }
350       { retextfaraway } { \l__zrefvario_ref_language_tl }
351       \zref@default
352     } ,
353     retextpagerange =
354     {
355       \msg_warning:nxx { zref-vario } { missing-string }
356       { retextpagerange } { \l__zrefvario_ref_language_tl }
357       \zref@default
358     } ,
359     retextlabelrange =
360     {
361       \msg_warning:nxx { zref-vario } { missing-string }
362       { retextlabelrange } { \l__zrefvario_ref_language_tl }
363       \zref@default
364     } ,
365   }
366 }
367 \group_end:
368 \msg_new:nnn { zref-vario } { missing-string }
369 { Missing~'#1'~string~for~language~'#2'~\msg_line_context:. }

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

370 \AddToHook { begindocument/before }
371 {
372   \AddToHook { begindocument }
373   {
374     \seq_if_in:NxF \g__zrefvario_setup_languages_seq
375     { \__zrefvario_base_lang:e { \l__zrefvario_ref_language_tl } }
376     {
377       \msg_warning:nxx { zref-vario } { language-not-setup }
378       { \l__zrefvario_ref_language_tl }
379     }
380   }
381 }
382 \msg_new:nnn { zref-vario } { language-not-setup }
383 {
384   Language~'#1'~is~not~set~up~for~'zref-vario'~.~
385   See~documentation~for~'\iow_char:N\zvLanguageSetup'.
386 }

```

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.

```

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

\zvhyperlink      \zvhyperlink{(text)}

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

```

(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`.

```

415 \cs_new_protected:Npn \__zrefvario_vref_pagenum:Nn #1#2
416   {
417     \exp_args:NNNo \exp_args:NNo \cs_set_nopar:Npn #1

```

```

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

```

427 \bool_new:N \l__zrefvario_varioref_setup_bool
428 \cs_new_protected:Npn \__zrefvario_varioref_setup:
429 {
430   \bool_if:NF \l__zrefvario_varioref_setup_bool
431   {
432     \bool_set_true:N \l__zrefvario_varioref_setup_bool
433     \@vrefhandlespacefalse
434     \cs_set_eq:NN \vref@pagenum \__zrefvario_vref_pagenum:Nn
435     \cs_set_eq:NN \vrefpagenum \__zrefvario_vrefpagenum:Nn
436     \cs_set_eq:NN \vref@label \__zrefvario_vref_label:n
437     \__zrefvario_ref_opt_tl_get:xnN
438     { \l__zrefvario_ref_language_tl }
439     { reftextfaceafter }
440     \reftextfaceafter
441     \__zrefvario_ref_opt_tl_get:xnN
442     { \l__zrefvario_ref_language_tl }
443     { reftextfacebefore }
444     \reftextfacebefore
445     \__zrefvario_ref_opt_tl_get:xnN
446     { \l__zrefvario_ref_language_tl }
447     { reftextafter }
448     \reftextafter
449     \__zrefvario_ref_opt_tl_get:xnN
450     { \l__zrefvario_ref_language_tl }
451     { reftextbefore }
452     \reftextbefore
453     \__zrefvario_ref_opt_tl_get:xnN
454     { \l__zrefvario_ref_language_tl }
455     { reftextcurrent }
456     \reftextcurrent
457     \__zrefvario_ref_opt_cs_get:xnnN
458     { \l__zrefvario_ref_language_tl }
459     { reftextfaraway } { n }
460     \reftextfaraway
461     \__zrefvario_ref_opt_cs_get:xnnN
462     { \l__zrefvario_ref_language_tl }
463     { reftextpagerange } { nn }
464     \reftextpagerange

```

```

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

```

(End of definition for __zrefvario_varioref_setup:.)

User commands.

```

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

```

“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.

```

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

```

(End of definition for \zvref.)

```

\zvpageref      \zvpageref*[{options}]{(label)}
502 \NewDocumentCommand \zvpageref { s O { } m }
503 { \zref@wrapper@babel \__zrefvario_zvpageref:nnn {#3} {#1} {#2} }
504 \cs_new_protected:Npn \__zrefvario_zvpageref:nnn #1#2#3

```

```

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

```

(End of definition for \zvpageref.)

```

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

```

```

555 }
556 \cs_new_protected:Npn \__zrefvario_vrefrange:nnn #1#2#3
557 { \vrefrange [#{#1}] {#2} {#3} }
558 \cs_generate_variant:Nn \__zrefvario_vrefrange:nnn { Vnn }

```

(End of definition for \zvrefrange.)

```

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

```

(End of definition for \zvpagerefrange.)

```

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

```

(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 `vcurent` 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.

```

618 \bool_new:N \g__zrefvario_zrefcheck_available_bool
619 \bool_new:N \l__zrefvario_with_vcheck_bool
620 \tl_new:N \l__zrefvario_vcheck_tl
621 \AddToHook { begindocument/before }
622   {
623     \IfPackageLoadedTF { zref-check }
624     {
625       \IfPackageAtLeastTF { zref-check } { 2022-02-08 }
626       {
627         \bool_gset_true:N \g__zrefvario_zrefcheck_available_bool
628         \keys_define:nn { zref-vario/current }
629         {
630           vcheck .choices:nn =
631             { above , below }
632             {
633               \bool_set_true:N \l__zrefvario_with_vcheck_bool
634               \tl_set:NV \l__zrefvario_vcheck_tl \l_keys_choice_tl
635             } ,
636           }
637       }
638     {
639       \bool_gset_false:N \g__zrefvario_zrefcheck_available_bool

```



```

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

```

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`.

```

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

```

```

682   reftextbefore = {auf-der-\zvhyperlink{vorherigen-Seite}} ,
683   reftextcurrent = {auf-\zvhyperlink{dieser-Seite}} ,
684   reftextfaraway = {auf-\zcpageref[d=D,g=f]{#1}} ,
685   reftextpagerange = {auf-den-\zcpageref[range,d=D,g=f]{#1,#2}} ,
686   reftextlabelrange = {\zcref[range]{#1,#2}} ,
687 }
688 \zvLanguageSetup { french }
689 {
690   reftextfaceafter =
691     {\zvhyperlink{page-\reftextvario{ci-contre}{suivante}}} ,
692   reftextfacebefore =
693     {\zvhyperlink{page-\reftextvario{ci-contre}{précédente}}} ,
694   reftextafter = {\zvhyperlink{page~suivante}} ,
695   reftextbefore = {\zvhyperlink{page~précédente}} ,
696   reftextcurrent = {de-la-\zvhyperlink{présente-page}} ,
697   reftextfaraway = {\zcpageref[g=f]{#1}} ,
698   reftextpagerange = {\zcpageref[range,g=f]{#1,#2}} ,
699   reftextlabelrange = {\zcref[range]{#1,#2}} ,
700 }
701 \zvLanguageSetup { portuguese }
702 {
703   reftextfaceafter =
704     {na-\zvhyperlink{\reftextvario{página-oposta}{próxima-página}}} ,
705   reftextfacebefore =
706     {na-\zvhyperlink{página-\reftextvario{oposta}{anterior}}} ,
707   reftextafter =
708     {na-\zvhyperlink{\reftextvario{página-seguinte}{próxima-página}}} ,
709   reftextbefore =
710     {na-\zvhyperlink{página-\reftextvario{anterior}{precedente}}} ,
711   reftextcurrent = {\zvhyperlink{nesta-página}} ,
712   reftextfaraway = {na-\zcpageref[g=f]{#1}} ,
713   reftextpagerange = {nas-\zcpageref[range,g=f]{#1,#2}} ,
714   reftextlabelrange = {\zcref[range]{#1,#2}} ,
715 }
716 \zvLanguageSetup { spanish }
717 {
718   reftextfaceafter = {en-la-\zvhyperlink{página-siguiente}} ,
719   reftextfacebefore = {en-la-\zvhyperlink{página-anterior}} ,
720   reftextafter = {en-la-\zvhyperlink{página-siguiente}} ,
721   reftextbefore = {en-la-\zvhyperlink{página-anterior}} ,
722   reftextcurrent = {en-\zvhyperlink{esta-página}} ,
723   reftextfaraway = {en-la-\zcpageref[g=f]{#1}} ,
724   reftextpagerange = {en-las-\zcpageref[range,g=f]{#1,#2}} ,
725   reftextlabelrange = {\zcref[range]{#1,#2}} ,
726 }
727 \zvLanguageSetup { dutch }
728 {
729   reftextfaceafter =
730     {op-de-\zvhyperlink{\reftextvario{rechter-pagina}{pagina-hiernaast}}} ,
731   reftextfacebefore =
732     {op-de-\zvhyperlink{\reftextvario{linker-pagina}{pagina-hiernaast}}} ,
733   reftextafter =
734     {op-de-\zvhyperlink{\reftextvario{volgende-pagina}{pagina-hierna}}} ,

```

```

735   reftextbefore =
736     {op-de-\zvhyperlink{\reftextvario{vorige-pagina}{pagina-hiervoor}}} ,
737   reftextcurrent = {op-\zvhyperlink{deze-pagina}} ,
738   reftextfaraway = {op-\zcpageref[g=f]{#1}} ,
739   reftextpagerange = {op-\zcpageref[range,g=f]{#1,#2}} ,
740   reftextlabelrange = {\zcref[range]{#1,#2}} ,
741 }

```

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

```

742 \zvLanguageSetup { italian }
743 {
744   reftextfaceafter =
745     {nella-\zvhyperlink{pagina-\reftextvario{a-fianco}{successiva}}} ,
746   reftextfacebefore =
747     {nella-\zvhyperlink{pagina-\reftextvario{a-fianco}{precedente}}} ,
748   reftextafter =
749     {nella-\zvhyperlink{pagina-\reftextvario{segunte}{successiva}}} ,
750   reftextbefore = {nella-\zvhyperlink{pagina-precedente}} ,
751   reftextcurrent = {in-\zvhyperlink{questa-pagina}} ,
752   reftextfaraway = {a-\zcpageref[g=f]{#1}} ,
753   reftextpagerange = {nelle-\zcpageref[range,g=f]{#1,#2}} ,
754   reftextlabelrange = {\zcref[range]{#1,#2}} ,
755 }
756 </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		<code>\cs_gset_nopar:Npn</code>
<code>\AddToHook</code> 26, 136, 313, 370, 372, 389, 391, 621 212, 223, 234, 245, 256, 267
B		<code>\cs_if_exist:NTF</code> 82, 89
bool commands:		<code>\cs_new:Npn</code> 49, 51, 60, 63
<code>\bool_gset_false:N</code> 639, 648		<code>\cs_new_nopar:Npn</code> 277, 283, 289
<code>\bool_gset_true:N</code> 393, 627		<code>\cs_new_protected:Npn</code> 66, 80, 156,
<code>\bool_if:NTF</code> 430		297, 400, 415, 420, 425, 428, 485,
<code>\bool_if:nTF</code> 301, 489, 508, 545, 569, 608		499, 504, 532, 541, 556, 565, 599, 604
<code>\bool_lazy_all:nTF</code> 402		<code>\cs_set_eq:NN</code> 85, 92, 96, 434, 435, 436
<code>\bool_lazy_and:nnTF</code> 517, 578		<code>\cs_set_nopar:Npn</code> 417
<code>\bool_new:N</code> . . . 387, 388, 427, 618, 619		E
<code>\bool_set_false:N</code>		exp commands:
. 304, 492, 511, 548, 572, 611		<code>\exp_args:NNNo</code> 417
<code>\bool_set_true:N</code> 394, 432, 633		<code>\exp_args:NNo</code> 417
C		F
cs commands:		<code>\fullref</code> 5, 615
<code>\cs_generate_variant:Nn</code> 57,		<code>\fullrefformat</code> 480
62, 65, 79, 104, 162, 501, 534, 558, 601		G
		group commands:
		<code>\group_begin:</code>

138, 299, 311, 487, 506, 543, 567, 606	
<code>\group_end:</code>	
147, 309, 367, 497, 530, 554, 597, 616	
H	
<code>\hyperlink</code>	409
I	
<code>\IfPackageAtLeastTF</code>	625
<code>\IfPackageLoadedTF</code>	623
iow commands:	
<code>\iow_char:N</code>	385
K	
keys commands:	
<code>\l_keys_choice_tl</code>	634
<code>\keys_define:nn</code>	11,
28, 109, 118, 123, 163, 628, 640, 649	
<code>\keys_set:nn</code>	
.. 9, 142, 300, 303, 315, 488, 491,	
507, 510, 544, 547, 568, 571, 607, 610	
M	
msg commands:	
<code>\msg_line_context:</code> 38, 41, 46, 152, 369	
<code>\msg_new:nnn</code>	
..... 37, 39, 44, 150, 368, 382, 656	
<code>\msg_warning:nn</code>	643, 652
<code>\msg_warning:nnn</code> .. 16, 20, 32, 146, 377	
<code>\msg_warning:nnnn</code>	
319, 325, 331, 337, 343, 349, 355, 361	
N	
<code>\NeedsTeXFormat</code>	3
<code>\NewDocumentCommand</code>	8,
134, 295, 398, 483, 502, 535, 559, 602	
P	
<code>\ProvidesExplPackage</code>	4
R	
<code>\reftextafter</code>	448
<code>\reftextbefore</code>	452
<code>\reftextcurrent</code>	5, 106, 456
<code>\reftextfaceafter</code>	440
<code>\reftextfacebefore</code>	444
<code>\reftextfaraway</code>	308, 460
<code>\reftextlabelrange</code>	286, 468
<code>\reftextpagerange</code>	464
<code>\reftextvario</code>	664, 666, 668,
670, 672, 691, 693, 704, 706, 708,	
710, 730, 732, 734, 736, 745, 747, 749	
<code>\RequirePackage</code>	6, 7
S	
seq commands:	
<code>\seq_gput_right:Nn</code>	143
<code>\seq_if_in:NnTF</code>	374
<code>\seq_new:N</code>	133
T	
TeX and L ^A T _E X 2 _ε commands:	
<code>\@onlypreamble</code>	155
<code>\@vrefhandlespace</code>	13
<code>\@vrefhandlespacefalse</code>	433
<code>\c@vrcnt</code>	522, 587
<code>\vr@f</code>	13
<code>\vref@label</code>	11, 436
<code>\vref@pagenum</code>	11, 434, 522, 587
<code>\vref@star</code>	13
<code>\zref@default</code>	
321, 327, 333, 339, 345, 351, 357, 363	
<code>\zref@extractdefault</code>	410, 418
<code>\zref@iflistcontainsprop</code>	18
<code>\zref@ifpropundefined</code>	15
<code>\zref@labelbyprops</code>	426
<code>\ZREF@mainlist</code>	18
<code>\zref@refused</code>	422
<code>\zref@wrapper@babel</code>	
296, 484, 503, 537, 538, 561, 562, 603	
<code>\the</code>	522, 587
tl commands:	
<code>\tl_gset:Nn</code>	160
<code>\tl_if_empty_p:n</code>	406
<code>\tl_if_eq:NNTF</code>	524, 585, 588
<code>\tl_if_exist:NNTF</code>	68, 158
<code>\tl_new:N</code>	10,
58, 105, 107, 108, 132, 159, 397, 620	
<code>\tl_set:Nn</code> 19, 59, 106, 113, 114, 141,	
306, 312, 494, 513, 552, 576, 613, 634	
<code>\tl_set_eq:NN</code>	71, 75
<code>\tl_use:N</code>	54
<code>\l_tmpa_tl</code> .. 522, 524, 583, 585, 587, 588	
<code>\l_tmpb_tl</code>	523, 524, 584, 585, 588
V	
<code>\vnameref</code>	10
<code>\vpageref</code>	5, 10, 533
<code>\vpagerefcompare</code>	551, 575
<code>\vpagerefrange</code>	5, 10, 600
<code>\vref</code>	5, 10, 11, 13
<code>\vrefformat</code>	13, 472, 500
<code>\vrefpagenum</code>	11, 435, 523, 583, 584
<code>\vrefrange</code>	5, 10, 557
<code>\vrefrangeformat</code>	476
Z	
<code>\zcheck</code>	16

`\zcpageref` 5, 673, 674, 684, 685, 697, 698, 712, 713, 723, 724, 738, 739, 752, 753
`\zcref` 5, 280, 292, 675, 686, 699, 714, 725, 740, 754
`\zfullref` 8, 16, 602
zrefcheck commands:
`\zrefcheck_zrefvario_label:` 521, 582
`\zrefcheck_zrefvario_run_check_on_label:nn` 526, 590, 592
zrefclever commands:
`\zrefclever_language_if_declared:nTF` 53, 139
`\zrefclever_language_varname:n` .. 54
`\l_zrefclever_ref_language_tl` . 3, 59
`\zreftextfaraway` 8, 293, 295
zrefvario internal commands:
`__zrefvario_base_lang:n` 3, 51, 51, 57, 61, 64, 144, 375
`__zrefvario_hyperlink:n` ... 399, 400
`\l_zrefvario_hyperlink_bool` 11, 304, 388, 394, 405, 492, 511, 548, 572, 611
`\l__zrefvario_linklabel_tl` 11, 306, 397, 406, 410, 494, 513, 552, 576, 613
`__zrefvario_opt_tl_gset:Nn` 156, 162, 167, 176, 185, 194, 203
`\l_zrefvario_pageprop_tl` 10, 19, 418, 426
`\l__zrefvario_ref_language_tl` 58, 320, 326, 332, 338, 344, 350, 356, 362, 375, 378, 438, 442, 446, 450, 454, 458, 462, 466, 470, 474, 478
`__zrefvario_ref_opt_cs_get:nnnN` 4, 8, 66, 80, 104, 457, 461, 465, 469, 473, 477
`__zrefvario_ref_opt_default_cs:nn` 3, 49, 49, 90, 93, 278, 284, 290
`__zrefvario_ref_opt_name_lang_cs:nnn` 4, 63, 63, 65, 83, 86, 98, 214, 225, 236, 247, 258, 269
`__zrefvario_ref_opt_name_lang_tl:nn` 3, 60, 60, 62, 69, 72, 76, 169, 178, 187, 196, 205
`__zrefvario_ref_opt_tl_get:nnN` . 4, 66, 66, 79, 437, 441, 445, 449, 453
`\l_zrefvario_reftextcurrent_tl` 5, 105, 106, 113, 515, 553, 577
`\l__zrefvario_reftextother_tl` 108, 120, 516
`\l_zrefvario_setup_language_tl` 132, 141, 170, 179, 188, 197, 206, 215, 226, 237, 248, 259, 270, 312
`\g__zrefvario_setup_languages_seq` 133, 143, 374
`__zrefvario_varioref_setup:` ... 307, 427, 428, 495, 514, 550, 574, 614
`\l_zrefvario_varioref_setup_bool` 12, 427, 430, 432
`\l__zrefvario_vcheck_tl` 527, 591, 593, 620, 634
`__zrefvario_vpageref:nnn` 515, 532, 534
`__zrefvario_vpagerefrange:nnn` 577, 599, 601
`__zrefvario_vref:nn` .. 496, 499, 501
`__zrefvario_vref_label:n` 415, 425, 436
`__zrefvario_vref_pagenum:Nn` 415, 415, 423, 434
`\l__zrefvario_vref_reftextcurrent_tl` 5, 107, 114, 496
`__zrefvario_vrefpagenum:Nn` 415, 420, 435
`__zrefvario_vrefrange:nnn` 553, 556, 558
`\l_zrefvario_with_vcheck_bool` 519, 580, 619, 633
`__zrefvario_zfullref:nnn` .. 603, 604
`\g__zrefvario_zref_hyperref_loaded_bool` 387, 393, 404
`\g__zrefvario_zrefcheck_available_bool` ... 518, 579, 618, 627, 639, 648
`__zrefvario_zreftextfaraway:nnn` 296, 297
`__zrefvario_zvpageref:nnn` . 503, 504
`__zrefvario_zvpagerefrange:nnnn` 562, 565
`__zrefvario_zvref:nnn` 484, 485
`__zrefvario_zvrefrange:nnnn` 538, 541
`\zvhyperlink` 11, 398, 664, 666, 668, 670, 672, 679, 680, 681, 682, 683, 691, 693, 694, 695, 696, 704, 706, 708, 710, 711, 718, 719, 720, 721, 722, 730, 732, 734, 736, 737, 745, 747, 749, 750, 751
`\zvLanguageSetup` 5, 11, 132, 661, 677, 688, 701, 716, 727, 742
`\zvpageref` 5, 8, 13, 281, 502
`\zvpagerefrange` 15, 287, 559
`\zvref` 13, 483
`\zvrefrange` 14, 535
`\zvsetup` 2, 8