

# The **zref-vario** package<sup>\*</sup>

## Code documentation

Gustavo Barros<sup>†</sup>

2022-02-11

**EXPERIMENTAL**

### Contents

<b>1</b>	<b>Initial setup</b>	<b>2</b>
<b>2</b>	<b>Customization</b>	<b>2</b>
<b>3</b>	<b>User interface</b>	<b>9</b>
<b>4</b>	<b>zref-check integration</b>	<b>13</b>
<b>5</b>	<b>Languages</b>	<b>14</b>
	<b>Index</b>	<b>16</b>

---

<sup>\*</sup>This file describes v0.1.3-alpha, released 2022-02-11.

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

# 1 Initial setup

Start the DocStrip guards.

```

1 <*package>
   Identify the internal prefix (LATEX3 DocStrip convention).
2 <@@=zrefvario>
   Identify the package.
3 \ProvidesExplPackage {zref-vario} {2022-02-11} {0.1.3-alpha}
4   {Extended LATEX page cross-references with varioref and zref-clever}
   Required packages.
5 \AddToHook { begindocument/before }
6   {
7     \RequirePackage { varioref }
8     \RequirePackage { zref-clever } [ 2022-01-28 ] % v0.2.0-alpha
9   }

```

# 2 Customization

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>}
10 \cs_new:Npn \__zrefvario_ref_opt_default_cs:nn #1#2
11   { __zrefvario_ref_opt_default_ #1 : #2 }

(End definition for \__zrefvario_ref_opt_default_cs:nn.)

```

```

\__zrefvario_base_lang:n                \__zrefvario_base_lang:n {<language>}
12 \cs_new:Npn \__zrefvario_base_lang:n #1
13   {
14     \zrefclever_language_if_declared:nTF {#1}
15       { \tl_use:c { \zrefclever_language_varname:n {#1} } }
16       { zv_unknown_language }
17   }
18 \cs_generate_variant:Nn \__zrefvario_base_lang:n { e }

(End 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.
19 \tl_new:N \l__zrefvario_ref_language_tl
20 \tl_set:Nn \l__zrefvario_ref_language_tl { \l_zrefclever_ref_language_tl }

(End definition for \l__zrefvario_ref_language_tl.)

```

```

\__zrefvario_ref_opt_name_lang_tl:nn    \__zrefvario_ref_opt_name_lang_tl:nn {<lang>} {<option>}
21 \cs_new:Npn \__zrefvario_ref_opt_name_lang_tl:nn #1#2
22   { g__zrefvario_ref_opt_lang_ \__zrefvario_base_lang:n {#1} _ #2 _tl }
23 \cs_generate_variant:Nn \__zrefvario_ref_opt_name_lang_tl:nn { en }

(End definition for \__zrefvario_ref_opt_name_lang_tl:nn.)

```

```

\__zrefvario_ref_opt_name_lang_cs:nnn {\lang} {\option} {\signature}}
24 \cs_new:Npn \__zrefvario_ref_opt_name_lang_cs:nnn #1#2#3
25   { \__zrefvario_ref_opt_lang_ \__zrefvario_base_lang:n {#1} _ #2 : #3 }
26 \cs_generate_variant:Nn \__zrefvario_ref_opt_name_lang_cs:nnn { enn }

```

(End 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}}
27 \cs_new_protected:Npn \__zrefvario_ref_opt_tl_get:nnN #1#2#3
28   {
29     \tl_if_exist:cTF
30     { \__zrefvario_ref_opt_name_lang_tl:nn {#1} {#2} }
31     {
32       \tl_set_eq:Nc #3
33       { \__zrefvario_ref_opt_name_lang_tl:nn {#1} {#2} }
34     }
35     {
36       \tl_set_eq:Nc #3
37       { \__zrefvario_ref_opt_name_lang_tl:nn { zv_unknown_language } {#2} }
38     }
39   }
40 \cs_generate_variant:Nn \__zrefvario_ref_opt_tl_get:nnN { xnN }
41 \cs_new_protected:Npn \__zrefvario_ref_opt_cs_get:nnnN #1#2#3#4
42   {
43     \cs_if_exist:cTF
44     { \__zrefvario_ref_opt_name_lang_cs:nnn {#1} {#2} {#3} }
45     {
46       \cs_set_eq:Nc #4
47       { \__zrefvario_ref_opt_name_lang_cs:nnn {#1} {#2} {#3} }
48     }
49     {
50       \cs_if_exist:cTF
51       { \__zrefvario_ref_opt_default_cs:nn {#2} {#3} }
52       {
53         \cs_set_eq:Nc #4
54         { \__zrefvario_ref_opt_default_cs:nn {#2} {#3} }
55       }
56       {
57         \cs_set_eq:Nc #4
58         {
59           \__zrefvario_ref_opt_name_lang_cs:nnn
60           { zv_unknown_language } {#2} {#3}
61         }
62       }
63     }
64   }
65 \cs_generate_variant:Nn \__zrefvario_ref_opt_cs_get:nnnN { xnnN }

```

(End 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.

```

66 \tl_new:N \l__zrefvario_reftextcurrent_tl
67 \tl_set:Nn \l__zrefvario_reftextcurrent_tl { \reftextcurrent }
68 \tl_new:N \l__zrefvario_vref_reftextcurrent_tl
69 \tl_new:N \l__zrefvario_reftextother_tl
70 \keys_define:nn { zref-vario/current }
71 {
72   vcurrent .code:n =
73   {
74     \tl_set:Nn \l__zrefvario_reftextcurrent_tl {#1}
75     \tl_set:Nn \l__zrefvario_vref_reftextcurrent_tl {#1}
76   } ,
77   vcurrent .value_required:n = true ,
78 }
79 \keys_define:nn { zref-vario/other }
80 {
81   vother .tl_set:N = \l__zrefvario_reftextother_tl ,
82   vother .value_required:n = true ,
83 }
84 \keys_define:nn { }
85 {
86   zref-vario/ref .inherit:n =
87   { zref-clever/reference } ,
88   zref-vario/ref-oneoptarg .inherit:n =
89   { zref-clever/reference , zref-vario/current } ,
90   zref-vario/ref-twooptargs .inherit:n =
91   { zref-clever/reference , zref-vario/current , zref-vario/other } ,
92 }

```

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

```

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

93 \tl_new:N \l__zrefvario_setup_language_tl
94 \seq_new:N \g__zrefvario_setup_languages_seq
95 \NewDocumentCommand \zvLanguageSetup { m m }
96 {
97   \AddToHook { begindocument/before }
98   {
99     \group_begin:
100     \zrefclever_language_if_declared:nTF {#1}
101     {
102       \tl_set:Nn \l__zrefvario_setup_language_tl {#1}
103       \keys_set:nn { zref-vario/langsetup } {#2}
104       \seq_gput_right:Nx \g__zrefvario_setup_languages_seq

```

```

105         { \_zrefvario_base_lang:n {#1} }
106     }
107     { \msg_warning:nnn { zref-vario } { unknown-language-setup } {#1} }
108     \group_end:
109 }
110 }
111 \msg_new:nnn { zref-vario } { unknown-language-setup }
112 {
113     Language~'~#1'~is-unknown~\msg_line_context:~Can't~set~it~up.~
114     The~language~must~be~declared~for~'zref-clever',~see~its~documentation.
115 }
116 \@onlypreamble \zvLanguageSetup

```

*(End definition for \zvLanguageSetup.)*

```

117 \cs_new_protected:Npn \_zrefvario_opt_tl_gset:Nn #1#2
118 {
119     \tl_if_exist:NF #1
120     { \tl_new:N #1 }
121     \tl_gset:Nn #1 {#2}
122 }
123 \cs_generate_variant:Nn \_zrefvario_opt_tl_gset:Nn { cn }
124 \keys_define:nn { zref-vario/langsetup }
125 {
126     retextfaceafter .code:n =
127     {
128         \_zrefvario_opt_tl_gset:cn
129         {
130             \_zrefvario_ref_opt_name_lang_tl:en
131             { \l__zrefvario_setup_language_tl }
132             { retextfaceafter }
133         } {#1}
134     } ,
135     retextfacebefore .code:n =
136     {
137         \_zrefvario_opt_tl_gset:cn
138         {
139             \_zrefvario_ref_opt_name_lang_tl:en
140             { \l__zrefvario_setup_language_tl }
141             { retextfacebefore }
142         } {#1}
143     } ,
144     retextafter .code:n =
145     {
146         \_zrefvario_opt_tl_gset:cn
147         {
148             \_zrefvario_ref_opt_name_lang_tl:en
149             { \l__zrefvario_setup_language_tl }
150             { retextafter }
151         } {#1}
152     } ,
153     retextbefore .code:n =
154     {
155         \_zrefvario_opt_tl_gset:cn
156         {

```

```

157         \__zrefvario_ref_opt_name_lang_tl:en
158         { \l__zrefvario_setup_language_tl }
159         { reftextbefore }
160     } {#1}
161 },
162 reftextcurrent .code:n =
163 {
164     \__zrefvario_opt_tl_gset:cn
165     {
166         \__zrefvario_ref_opt_name_lang_tl:en
167         { \l__zrefvario_setup_language_tl }
168         { reftextcurrent }
169     } {#1}
170 },
171 reftextfaraway .code:n =
172 {
173     \cs_gset_nopar:cpn
174     {
175         \__zrefvario_ref_opt_name_lang_cs:enn
176         { \l__zrefvario_setup_language_tl }
177         { reftextfaraway } { n }
178     }
179     ##1
180     {#1}
181 },
182 reftextpagerange .code:n =
183 {
184     \cs_gset_nopar:cpn
185     {
186         \__zrefvario_ref_opt_name_lang_cs:enn
187         { \l__zrefvario_setup_language_tl }
188         { reftextpagerange } { nn }
189     }
190     ##1##2
191     {#1}
192 },
193 reftextlabelrange .code:n =
194 {
195     \cs_gset_nopar:cpn
196     {
197         \__zrefvario_ref_opt_name_lang_cs:enn
198         { \l__zrefvario_setup_language_tl }
199         { reftextlabelrange } { nn }
200     }
201     ##1##2
202     {#1}
203 },
204 vrefformat .code:n =
205 {
206     \cs_gset_nopar:cpn
207     {
208         \__zrefvario_ref_opt_name_lang_cs:enn
209         { \l__zrefvario_setup_language_tl }
210         { vrefformat } { nn }

```

```

211     }
212     ##1#2
213     {#1}
214   } ,
215   vrefrangeformat .code:n =
216   {
217     \cs_gset_nopar:cpn
218     {
219       \__zrefvario_ref_opt_name_lang_cs:enn
220       { \l__zrefvario_setup_language_tl }
221       { vrefrangeformat } { nnn }
222     }
223     ##1#2##3
224     {#1}
225   } ,
226   fullreformat .code:n =
227   {
228     \cs_gset_nopar:cpn
229     {
230       \__zrefvario_ref_opt_name_lang_cs:enn
231       { \l__zrefvario_setup_language_tl }
232       { fullreformat } { n }
233     }
234     ##1
235     {#1}
236   } ,
237 }

```

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).

```

238 \cs_new_nopar:cpn
239 { \__zrefvario_ref_opt_default_cs:nn { vrefformat } { nn } } #1#2
240 {
241   \zcref {#2} ~
242   \zvpageref [ S=false, typeset=both, vcurrent={#1} ] {#2}
243 }
244 \cs_new_nopar:cpn
245 { \__zrefvario_ref_opt_default_cs:nn { vrefrangeformat } { nnn } } #1#2#3
246 {
247   \reftextlabelrange {#2} {#3} ~
248   \zvpagerefrange [ S=false, typeset=both, vcurrent={#1} ] {#2} {#3}
249 }
250 \cs_new_nopar:cpn
251 { \__zrefvario_ref_opt_default_cs:nn { fullreformat } { n } } #1
252 {
253   \zcref {#1} ~
254   \zreftextfaraway [ S=false, typeset=both ] {#1}
255 }
256 \NewDocumentCommand \zreftextfaraway { s O { } m }
257 { \zref@wrapper@babel \__zrefvario_zreftextfaraway:nnn {#3} {#1} {#2} }
258 \cs_new_protected:Npn \__zrefvario_zreftextfaraway:nnn #1#2#3
259 {

```

```

260 \group_begin:
261 \keys_set:nn { zref-vario/ref } {#3}
262 \bool_if:nT {#2}
263 { \keys_set:nn { zref-vario/ref } { nohyperref } }
264 \__zrefvario_varioref_setup:
265 \reftextfaraway {#1}
266 \group_end:
267 }

```

Warnings and values for missing strings / unknown languages.

```

268 \group_begin:
269 \tl_set:Nn \l__zrefvario_setup_language_tl { zv_unknown_language }
270 \AddToHook { begindocument/before }
271 {
272   \keys_set:nn { zref-vario/langsetup }
273   {
274     retextfaceafter =
275     {
276       \msg_warning:nxxx { zref-vario } { missing-string }
277       { retextfaceafter } { \l__zrefvario_ref_language_tl }
278       \zref@default
279     } ,
280     retextfacebefore =
281     {
282       \msg_warning:nxxx { zref-vario } { missing-string }
283       { retextfacebefore } { \l__zrefvario_ref_language_tl }
284       \zref@default
285     } ,
286     retextafter =
287     {
288       \msg_warning:nxxx { zref-vario } { missing-string }
289       { retextafter } { \l__zrefvario_ref_language_tl }
290       \zref@default
291     } ,
292     retextbefore =
293     {
294       \msg_warning:nxxx { zref-vario } { missing-string }
295       { retextbefore } { \l__zrefvario_ref_language_tl }
296       \zref@default
297     } ,
298     retextcurrent =
299     {
300       \msg_warning:nxxx { zref-vario } { missing-string }
301       { retextcurrent } { \l__zrefvario_ref_language_tl }
302       \zref@default
303     } ,
304     retextfaraway =
305     {
306       \msg_warning:nxxx { zref-vario } { missing-string }
307       { retextfaraway } { \l__zrefvario_ref_language_tl }
308       \zref@default
309     } ,
310     retextpagerange =
311     {

```

```

312         \msg_warning:nxxx { zref-vario } { missing-string }
313         { reftextpagerange } { \l__zrefvario_ref_language_tl }
314         \zref@default
315     } ,
316     reftextlabelrange =
317     {
318         \msg_warning:nxxx { zref-vario } { missing-string }
319         { reftextlabelrange } { \l__zrefvario_ref_language_tl }
320         \zref@default
321     } ,
322 }
323 }
324 \group_end:
325 \msg_new:nnn { zref-vario } { missing-string }
326 { Missing~'#1'~string~for~language~'#2'~\msg_line_context:. }

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

327 \AddToHook { begindocument/before }
328 {
329     \AddToHook { begindocument }
330     {
331         \seq_if_in:NxF \g__zrefvario_setup_languages_seq
332         { \__zrefvario_base_lang:e { \l__zrefvario_ref_language_tl } }
333         {
334             \msg_warning:nxx { zref-vario } { language-not-setup }
335             { \l__zrefvario_ref_language_tl }
336         }
337     }
338 }
339 \msg_new:nnn { zref-vario } { language-not-setup }
340 {
341     Language~'#1'~is-not-set-up~for~'zref-vario'~
342     See~documentation~for~'\iow_char:N\zvLanguageSetup'.
343 }

```

### 3 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`, `\vrefpagenum`, and `\vref@label`.

```

344 \cs_new_protected:Npn \__zrefvario_vref_pagenum:Nn #1#2
345 {
346     \exp_args:NNNo \exp_args:NNo \def #1
347     { \zref@extractdefault {#2} { page } { ?? } }
348 }
349 \cs_new_protected:Npn \__zrefvario_vrefpagenum:Nn #1#2
350 {
351     \zref@refused {#2}
352     \__zrefvario_vref_pagenum:Nn #1 {#2}
353 }
354 \cs_new_protected:Npn \__zrefvario_vref_label:n #1
355 { \zref@labelbyprops {#1} { page } }

```

(End 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.

```

356 \bool_new:N \l__zrefvario_varioref_setup_bool
357 \cs_new_protected:Npn \__zrefvario_varioref_setup:
358 {
359   \bool_if:NF \l__zrefvario_varioref_setup_bool
360   {
361     \bool_set_true:N \l__zrefvario_varioref_setup_bool
362     \@vrefhandlespacefalse
363     \cs_set_eq:NN \vref@pagenum \__zrefvario_vref_pagenum:Nn
364     \cs_set_eq:NN \vrefpagenum \__zrefvario_vrefpagenum:Nn
365     \cs_set_eq:NN \vref@label \__zrefvario_vref_label:n
366     \__zrefvario_ref_opt_tl_get:xnN
367       { \l__zrefvario_ref_language_tl }
368       { retextfaceafter }
369       \retextfaceafter
370     \__zrefvario_ref_opt_tl_get:xnN
371       { \l__zrefvario_ref_language_tl }
372       { retextfacebefore }
373       \retextfacebefore
374     \__zrefvario_ref_opt_tl_get:xnN
375       { \l__zrefvario_ref_language_tl }
376       { retextafter }
377       \retextafter
378     \__zrefvario_ref_opt_tl_get:xnN
379       { \l__zrefvario_ref_language_tl }
380       { retextbefore }
381       \retextbefore
382     \__zrefvario_ref_opt_tl_get:xnN
383       { \l__zrefvario_ref_language_tl }
384       { retextcurrent }
385       \retextcurrent
386     \__zrefvario_ref_opt_cs_get:xnnN
387       { \l__zrefvario_ref_language_tl }
388       { retextfaraway } { n }
389       \retextfaraway
390     \__zrefvario_ref_opt_cs_get:xnnN
391       { \l__zrefvario_ref_language_tl }
392       { retextpagerange } { nn }
393       \retextpagerange
394     \__zrefvario_ref_opt_cs_get:xnnN
395       { \l__zrefvario_ref_language_tl }
396       { retextlabelrange } { nn }
397       \retextlabelrange
398     \__zrefvario_ref_opt_cs_get:xnnN
399       { \l__zrefvario_ref_language_tl }
400       { vrefformat } { nn }
401       \vrefformat
402     \__zrefvario_ref_opt_cs_get:xnnN
403       { \l__zrefvario_ref_language_tl }

```

```

404         { vrefrangeformat } { nnn }
405         \vrefrangeformat
406         \__zrefvario_ref_opt_cs_get:xnnN
407         { \l__zrefvario_ref_language_tl }
408         { fullrefformat } { n }
409         \fullrefformat
410     }
411 }

```

(End definition for \\_\_zrefvario\_varioref\_setup.)  
User commands.

```

\zvref          \zvref{<*>[<options>]}{<label>}
412 \NewDocumentCommand \zvref { s O { } m }
413 { \zref@wrapper@babel \__zrefvario_zvref:nnn {#3} {#1} {#2} }
414 \cs_new_protected:Npn \__zrefvario_zvref:nnn #1#2#3
415 {
416     \group_begin:
417     \keys_set:nn { zref-vario/ref-oneoptarg } {#3}
418     \bool_if:nT {#2}
419     { \keys_set:nn { zref-vario/ref } { nohyperref } }
420     \__zrefvario_varioref_setup:
421     \__zrefvario_vref:Vn \l__zrefvario_vref_reftextcurrent_tl {#1}
422     \group_end:
423 }

```

“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.

```

424 \cs_new_protected:Npn \__zrefvario_vref:nn #1#2
425 { \vrefformat {#1} {#2} }
426 \cs_generate_variant:Nn \__zrefvario_vref:nn { Vn }

```

(End definition for \zvref.)

```

\zvpageref      \zvpageref{<*>[<options>]}{<label>}
427 \NewDocumentCommand \zvpageref { s O { } m }
428 { \zref@wrapper@babel \__zrefvario_zvpageref:nnn {#3} {#1} {#2} }
429 \cs_new_protected:Npn \__zrefvario_zvpageref:nnn #1#2#3
430 {
431     \group_begin:
432     \keys_set:nn { zref-vario/ref-twooptargs } {#3}
433     \bool_if:nT {#2}
434     { \keys_set:nn { zref-vario/ref } { nohyperref } }
435     \__zrefvario_varioref_setup:
436     \__zrefvario_vpageref:Vn \l__zrefvario_reftextcurrent_tl
437     \l__zrefvario_reftextother_tl {#1}
438     \bool_lazy_and:nnT
439     { \g__zrefvario_zrefcheck_available_bool }
440     { \l__zrefvario_with_vcheck_bool }
441     {
442         \zrefcheck_zrefvario_label:
443         \vref@pagenum \l_tmpa_tl { \the\c@vrcnt @vr }

```

```

444     \vrefpagenum \l_tmpb_tl {#1}
445     \tl_if_eq:NNT \l_tmpa_tl \l_tmpb_tl
446     {
447         \zrefcheck_zrefvario_run_check_on_label:Vn
448         \l__zrefvario_vcheck_tl {#1}
449     }
450 }
451 \group_end:
452 }
453 \cs_new_protected:Npn \__zrefvario_vpageref:nnn #1#2#3
454 { \vpageref [{#1}] [{#2}] [{#3}] }
455 \cs_generate_variant:Nn \__zrefvario_vpageref:nnn { VVn }

```

(End definition for \zvpageref.)

```

\zvrefrange      \zvrefrange{*}[\langle options \rangle]{\langle label \rangle}{\langle label \rangle}
456 \NewDocumentCommand \zvrefrange { s O { } m m }
457 {
458     \zref@wrapper@babel
459     { \zref@wrapper@babel \__zrefvario_zvrefrange:nnnn {#3} }
460     {#4} {#1} {#2}
461 }
462 \cs_new_protected:Npn \__zrefvario_zvrefrange:nnnn #1#2#3#4
463 {
464     \group_begin:
465     \keys_set:nn { zref-vario/ref-oneoptarg } {#4}
466     \bool_if:nT {#3}
467     { \keys_set:nn { zref-vario/ref } { nohyperref } }
468     \__zrefvario_varioref_setup:
469     \__zrefvario_vrefrange:Vnn \l__zrefvario_reftextcurrent_tl {#1} {#2}
470     \group_end:
471 }
472 \cs_new_protected:Npn \__zrefvario_vrefrange:nnn #1#2#3
473 { \vrefrange [{#1}] [{#2}] [{#3}] }
474 \cs_generate_variant:Nn \__zrefvario_vrefrange:nnn { Vnn }

```

(End definition for \zvrefrange.)

```

\zvpagerefrange  \zvpagerefrange{*}[\langle options \rangle]{\langle label \rangle}{\langle label \rangle}
475 \NewDocumentCommand \zvpagerefrange { s O { } m m }
476 {
477     \zref@wrapper@babel
478     { \zref@wrapper@babel \__zrefvario_zvpagerefrange:nnnn {#3} }
479     {#4} {#1} {#2}
480 }
481 \cs_new_protected:Npn \__zrefvario_zvpagerefrange:nnnn #1#2#3#4
482 {
483     \group_begin:
484     \keys_set:nn { zref-vario/ref-oneoptarg } {#4}
485     \bool_if:nT {#3}
486     { \keys_set:nn { zref-vario/ref } { nohyperref } }
487     \__zrefvario_varioref_setup:
488     \__zrefvario_vpagerefrange:Vnn \l__zrefvario_reftextcurrent_tl {#1} {#2}
489     \bool_lazy_and:nnT

```

```

490 { \g__zrefvario_zrefcheck_available_bool }
491 { \l__zrefvario_with_vcheck_bool }
492 {
493   \zrefcheck_zrefvario_label:
494   \vrefpagenum \l_tmpa_tl {#1}
495   \vrefpagenum \l_tmpb_tl {#2}
496   \tl_if_eq:NNT \l_tmpa_tl \l_tmpb_tl
497   {
498     \vref@pagenum \l_tmpa_tl { \the\c@vrcnt @vr }
499     \tl_if_eq:NNT \l_tmpa_tl \l_tmpb_tl
500     {
501       \zrefcheck_zrefvario_run_check_on_label:Vn
502       \l__zrefvario_vcheck_tl {#1}
503       \zrefcheck_zrefvario_run_check_on_label:Vn
504       \l__zrefvario_vcheck_tl {#2}
505     }
506   }
507 }
508 \group_end:
509 }
510 \cs_new_protected:Npn \__zrefvario_vpagerefrange:nnn #1#2#3
511 { \vpagerefrange [{#1}] {#2} {#3} }
512 \cs_generate_variant:Nn \__zrefvario_vpagerefrange:nnn { Vnn }

```

(End definition for \zvpagerefrange.)

```

\zfullref      \zfullref{*}[\options]{\label{}}
513 \NewDocumentCommand \zfullref { s O { } m }
514 { \zref@wrapper@babel \__zrefvario_zfullref:nnn {#3} {#1} {#2} }
515 \cs_new_protected:Npn \__zrefvario_zfullref:nnn #1#2#3
516 {
517   \group_begin:
518   \keys_set:nn { zref-vario/ref } {#3}
519   \bool_if:nT {#2}
520   { \keys_set:nn { zref-vario/ref } { nohyperref } }
521   \__zrefvario_varioref_setup:
522   \fullref {#1}
523   \group_end:
524 }

```

(End definition for \zfullref.)

## 4 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.

```

525 \bool_new:N \g__zrefvario_zrefcheck_available_bool
526 \bool_new:N \l__zrefvario_with_vcheck_bool
527 \tl_new:N \l__zrefvario_vcheck_tl
528 \AddToHook { begindocument/before }
529 {
530   \IfPackageLoadedTF { zref-check }
531   {
532     \IfPackageAtLeastTF { zref-check } { 2022-02-08 }
533     {
534       \bool_gset_true:N \g__zrefvario_zrefcheck_available_bool
535       \keys_define:nn { zref-vario/current }
536       {
537         vcheck .choices:nn =
538           { above , below }
539           {
540             \bool_set_true:N \l__zrefvario_with_vcheck_bool
541             \tl_set:NV \l__zrefvario_vcheck_tl \l_keys_choice_tl
542           } ,
543       }
544     }
545     {
546       \bool_gset_false:N \g__zrefvario_zrefcheck_available_bool
547       \keys_define:nn { zref-vario/current }
548       {
549         vcheck .code:n =
550           { \msg_warning:nn { zref-vario } { zref-check-unavailable } } ,
551       }
552     }
553   }
554   {
555     \bool_gset_false:N \g__zrefvario_zrefcheck_available_bool
556     \keys_define:nn { zref-vario/current }
557     {
558       vcheck .code:n =
559         { \msg_warning:nn { zref-vario } { zref-check-unavailable } } ,
560     }
561   }
562 }
563 \msg_new:nnn { zref-vario } { zref-check-unavailable }
564 {
565   Option~'vcheck'~is~only~available~when~'zref-check'~is~loaded~
566   and~newer~than~'2022-02-08-v0.2.4'.
567 }

```

## 5 Languages

```

568 \zvLanguageSetup { english }
569 {
570   retextfaceafter = {on~the~\retextvario{facing}{next}~page} ,
571   retextfacebefore = {on~the~\retextvario{facing}{preceding}~page} ,
572   retextafter = {on~the~\retextvario{following}{next}~page} ,
573   retextbefore = {on~the~\retextvario{preceding}{previous}~page} ,
574   retextcurrent = {on~\retextvario{this}{the~current}~page} ,

```

```

575     retextfaraway = {on~\zcpageref{#1}} ,
576     retextpagerange = {on~\zcpageref[range]{#1,#2}} ,
577     retextlabelrange = {\zcref[range]{#1,#2}} ,
578 }
579 \zvLanguageSetup { german }
580 {
581     retextfaceafter = {auf~der~nächsten~Seite} ,
582     retextfacebefore = {auf~der~vorherigen~Seite} ,
583     retextafter = {auf~der~nächsten~Seite} ,
584     retextbefore = {auf~der~vorherigen~Seite} ,
585     retextcurrent = {auf~dieser~Seite} ,
586     retextfaraway = {auf~\zcpageref[d=D,g=f]{#1}} ,
587     retextpagerange = {auf~den~\zcpageref[range,d=D,g=f]{#1,#2}} ,
588     retextlabelrange = {\zcref[range]{#1,#2}} ,
589 }
590 \zvLanguageSetup { french }
591 {
592     retextfaceafter = {page~\retextvario{ci-contre}{suivante}} ,
593     retextfacebefore = {page~\retextvario{ci-contre}{précédente}} ,
594     retextafter = {page~suivante} ,
595     retextbefore = {page~précédente} ,
596     retextcurrent = {de~la~présente~page} ,
597     retextfaraway = {\zcpageref[g=f]{#1}} ,
598     retextpagerange = {\zcpageref[range,g=f]{#1,#2}} ,
599     retextlabelrange = {\zcref[range]{#1,#2}} ,
600 }
601 \zvLanguageSetup { portuguese }
602 {
603     retextfaceafter = {na~\retextvario{página-oposta}{próxima-página}} ,
604     retextfacebefore = {na~página~\retextvario{oposta}{anterior}} ,
605     retextafter = {na~\retextvario{página-seguinte}{próxima-página}} ,
606     retextbefore = {na~página~\retextvario{anterior}{precedente}} ,
607     retextcurrent = {nesta~página} ,
608     retextfaraway = {na~\zcpageref[g=f]{#1}} ,
609     retextpagerange = {nas~\zcpageref[range,g=f]{#1,#2}} ,
610     retextlabelrange = {\zcref[range]{#1,#2}} ,
611 }
612 \zvLanguageSetup { spanish }
613 {
614     retextfaceafter = {en~la~página~siguiente} ,
615     retextfacebefore = {en~la~página~anterior} ,
616     retextafter = {en~la~página~siguiente} ,
617     retextbefore = {en~la~página~anterior} ,
618     retextcurrent = {en~esta~página} ,
619     retextfaraway = {en~la~\zcpageref[g=f]{#1}} ,
620     retextpagerange = {en~las~\zcpageref[range,g=f]{#1,#2}} ,
621     retextlabelrange = {\zcref[range]{#1,#2}} ,
622 }
623 \zvLanguageSetup { dutch }
624 {
625     retextfaceafter = {op~de~\retextvario{rechter-pagina}{pagina-hiernaast}} ,
626     retextfacebefore = {op~de~\retextvario{linker-pagina}{pagina-hiernaast}} ,

```

```

627   reftextafter = {op~de~\reftextvario{volgende~pagina}{pagina~hierna}} ,
628   reftextbefore = {op~de~\reftextvario{vorige~pagina}{pagina~hiervoor}} ,
629   reftextcurrent = {op~deze~pagina} ,
630   reftextfaraway = {op~\zcpageref[g=f]{#1}} ,
631   reftextpagerange = {op~\zcpageref[range,g=f]{#1,#2}} ,
632   reftextlabelrange = {\zcref[range]{#1,#2}} ,
633 }
634 \</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.

<b>A</b>		<b>G</b>	
\AddToHook	5, 97, 270, 327, 329, 528	group commands:	
<b>B</b>		\group_begin:	99, 260, 268, 416, 431, 464, 483, 517
bool commands:		\group_end:	108, 266, 324, 422, 451, 470, 508, 523
\bool_gset_false:N	546, 555	<b>I</b>	
\bool_gset_true:N	534	\IfPackageAtLeastTF	532
\bool_if:NTF	359	\IfPackageLoadedTF	530
\bool_if:nTF	262, 418, 433, 466, 485, 519	iow commands:	
\bool_lazy_and:nnTF	438, 489	\iow_char:N	342
\bool_new:N	356, 525, 526	<b>K</b>	
\bool_set_true:N	361, 540	keys commands:	
<b>C</b>		\l_keys_choice_tl	541
cs commands:		\keys_define:nn	70, 79, 84, 124, 535, 547, 556
\cs_generate_variant:Nn	18, 23, 26, 40, 65, 123, 426, 455, 474, 512	\keys_set:nn	103, 261, 263, 272, 417, 419, 432, 434, 465, 467, 484, 486, 518, 520
\cs_gset_nopar:Npn	173, 184, 195, 206, 217, 228	<b>M</b>	
\cs_if_exist:NTF	43, 50	msg commands:	
\cs_new:Npn	10, 12, 21, 24	\msg_line_context:	113, 326
\cs_new_nopar:Npn	238, 244, 250	\msg_new:nnn	111, 325, 339, 563
\cs_new_protected:Npn	27, 41, 117, 258, 344, 349, 354, 357, 414, 424, 429, 453, 462, 472, 481, 510, 515	\msg_warning:nn	550, 559
\cs_set_eq:NN	46, 53, 57, 363, 364, 365	\msg_warning:nnn	107, 334
<b>D</b>		\msg_warning:nnnn	276, 282, 288, 294, 300, 306, 312, 318
\def	346	<b>N</b>	
<b>E</b>		\NewDocumentCommand	95, 256, 412, 427, 456, 475, 513
exp commands:		<b>P</b>	
\exp_args:NNNo	346	\ProvidesExplPackage	3
\exp_args:NNo	346		
<b>F</b>			
\fullref	4, 522		
\fullrefformat	409		

<b>R</b>	
<code>\reftextafter</code> .....	377
<code>\reftextbefore</code> .....	381
<code>\reftextcurrent</code> .....	4, 67, 385
<code>\reftextfaceafter</code> .....	369
<code>\reftextfacebefore</code> .....	373
<code>\reftextfaraway</code> .....	265, 389
<code>\reftextlabelrange</code> .....	247, 397
<code>\reftextpagerange</code> .....	393
<code>\reftextvario</code> .....	570, 571, 572, 573, 574, 592, 593, 603, 604, 605, 606, 625, 626, 627, 628
<code>\RequirePackage</code> .....	7, 8
<b>S</b>	
seq commands:	
<code>\seq_gput_right:Nn</code> .....	104
<code>\seq_if_in:NnTF</code> .....	331
<code>\seq_new:N</code> .....	94
<b>T</b>	
TeX and L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> commands:	
<code>\@onlypreamble</code> .....	116
<code>\@vrefhandlespace</code> .....	11
<code>\@vrefhandlespacefalse</code> .....	362
<code>\c@vrcnt</code> .....	443, 498
<code>\vr@f</code> .....	11
<code>\vref@label</code> .....	9, 365
<code>\vref@pagenum</code> .....	9, 363, 443, 498
<code>\vref@star</code> .....	11
<code>\zref@default</code> .....	278, 284, 290, 296, 302, 308, 314, 320
<code>\zref@extractdefault</code> .....	347
<code>\zref@labelbyprops</code> .....	355
<code>\zref@refused</code> .....	351
<code>\zref@wrapper@babel</code> .....	257, 413, 428, 458, 459, 477, 478, 514
<code>\the</code> .....	443, 498
tl commands:	
<code>\tl_gset:Nn</code> .....	121
<code>\tl_if_eq:NNTF</code> .....	445, 496, 499
<code>\tl_if_exist:NTF</code> .....	29, 119
<code>\tl_new:N</code> ..	19, 66, 68, 69, 93, 120, 527
<code>\tl_set:Nn</code> ..	20, 67, 74, 75, 102, 269, 541
<code>\tl_set_eq:NN</code> .....	32, 36
<code>\tl_use:N</code> .....	15
<code>\l_tmpa_tl</code> ..	443, 445, 494, 496, 498, 499
<code>\l_tmpb_tl</code> ....	444, 445, 495, 496, 499
<b>V</b>	
<code>\vpageref</code> .....	4, 454
<code>\vpagerefrange</code> .....	4, 511
<code>\vref</code> .....	4, 11
<code>\vrefformat</code> .....	11, 401, 425
<code>\vrefpagenum</code> .....	9, 364, 444, 494, 495
<code>\vrefrange</code> .....	4, 473
<code>\vrefrangeformat</code> .....	405
<b>Z</b>	
<code>\zcheck</code> .....	13
<code>\zcpageref</code> .....	4, 575, 576, 586, 587, 597, 598, 608, 609, 619, 620, 630, 631
<code>\zceref</code> .....	3, 4, 241, 253, 577, 588, 599, 610, 621, 632
<code>\zfullref</code> .....	7, 13, 513
zrefcheck commands:	
<code>\zrefcheck_zrefvario_label:</code>	442, 493
<code>\zrefcheck_zrefvario_run_check_- on_label:nn</code> .....	447, 501, 503
zrefclever commands:	
<code>\zrefclever_language_if_declared:nTF</code> .....	14, 100
<code>\zrefclever_language_varname:n</code> ..	15
<code>\l_zrefclever_ref_language_tl</code> ..	2, 20
<code>\zreftextfaraway</code> .....	7, 254, 256
zrefvario internal commands:	
<code>\__zrefvario_base_lang:n</code> .....	2, 12, 22, 25, 105, 332
<code>\__zrefvario_opt_tl_gset:Nn</code> ....	117, 123, 128, 137, 146, 155, 164
<code>\l__zrefvario_ref_language_tl</code> ...	19, 277, 283, 289, 295, 301, 307, 313, 319, 332, 335, 367, 371, 375, 379, 383, 387, 391, 395, 399, 403, 407
<code>\__zrefvario_ref_opt_cs_get:nnnN</code> ..	3, 7, 27, 386, 390, 394, 398, 402, 406
<code>\__zrefvario_ref_opt_default_- cs:nn</code> ...	2, 10, 51, 54, 239, 245, 251
<code>\__zrefvario_ref_opt_name_lang_- cs:nnn</code> .....	3, 24, 44, 47, 59, 175, 186, 197, 208, 219, 230
<code>\__zrefvario_ref_opt_name_lang_- tl:nn</code> .....	2, 21, 30, 33, 37, 130, 139, 148, 157, 166
<code>\__zrefvario_ref_opt_tl_get:nnN</code> ..	3, 27, 366, 370, 374, 378, 382
<code>\l__zrefvario_reftextcurrent_tl</code> ..	4, 66, 67, 74, 436, 469, 488
<code>\l__zrefvario_reftextother_tl</code> ...	69, 81, 437
<code>\l__zrefvario_setup_language_tl</code> ..	93, 102, 131, 140, 149, 158, 167, 176, 187, 198, 209, 220, 231, 269
<code>\g__zrefvario_setup_languages_- seq</code> .....	94, 104, 331
<code>\__zrefvario_varioref_setup:</code> ...	264, 356, 420, 435, 468, 487, 521

\l_zrefvario_varioref_setup_-	440, 491, 526, 540
bool .....	10, 356, 359, 361
\l_zrefvario_vcheck_tl .....	448, 502, 504, 527, 541
\_zrefvario_vpageref:nnn .....	436, 453, 455
\_zrefvario_vpagerefrange:nnn ..	488, 510, 512
\_zrefvario_vref:nn ..	421, 424, 426
\_zrefvario_vref_label:n ..	344, 365
\_zrefvario_vref_pagenum:Nn	344, 363
\l_zrefvario_vref_reftextcurrent_-	
tl .....	4, 68, 75, 421
\_zrefvario_vrefpagenum:Nn	344, 364
\_zrefvario_vrefrange:nnn .....	469, 472, 474
\l_zrefvario_with_vcheck_bool ..	
\_zrefvario_zfullref:nnn ..	514, 515
\g_zrefvario_zrefcheck_available_-	
bool ...	439, 490, 525, 534, 546, 555
\_zrefvario_zreftextfaraway:nnn	
.....	257, 258
\_zrefvario_zvpageref:nnn ..	428, 429
\_zrefvario_zvpagerefrange:nnnn	
.....	478, 481
\_zrefvario_zvref:nnn .....	413, 414
\_zrefvario_zvrefrange:nnnn	459, 462
\zvLanguageSetup .....	
... 4, 93, 568, 579, 590, 601, 612, 623	
\zvpageref .....	4, 7, 11, 242, 427
\zvpagerefrange .....	12, 248, 475
\zvref .....	11, 412
\zvrefrange .....	12, 456