

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

Code documentation

Gustavo Barros[†]

2022-02-08

EXPERIMENTAL

Contents

1	Initial setup	2
2	Customization	2
3	User interface	9
4	zref-check integration	13
5	Languages	14
	Index	17

^{*}This file describes v0.1.2-alpha, released 2022-02-08.

[†]<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} {2022-02-08} {0.1.2-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 \keys_define:nn { zref-vario/langsetup }
118 {
119     retextfaceafter .code:n =
120     {
121         \tl_gset:cn
122         {
123             \_zrefvario_ref_opt_name_lang_tl:en
124             { \l__zrefvario_setup_language_tl }
125             { retextfaceafter }
126         } {#1}
127     } ,
128     retextfacebefore .code:n =
129     {
130         \tl_gset:cn
131         {
132             \_zrefvario_ref_opt_name_lang_tl:en
133             { \l__zrefvario_setup_language_tl }
134             { retextfacebefore }
135         } {#1}
136     } ,
137     retextafter .code:n =
138     {
139         \tl_gset:cn
140         {
141             \_zrefvario_ref_opt_name_lang_tl:en
142             { \l__zrefvario_setup_language_tl }
143             { retextafter }
144         } {#1}
145     } ,
146     retextbefore .code:n =
147     {
148         \tl_gset:cn
149         {
150             \_zrefvario_ref_opt_name_lang_tl:en
151             { \l__zrefvario_setup_language_tl }
152             { retextbefore }
153         } {#1}
154     } ,
155     retextcurrent .code:n =
156     {

```



```

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

```



```

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

```

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).

```

231 \cs_new_nopar:cpn
232 { \__zrefvario_ref_opt_default_cs:nn { vrefformat } { nn } } #1#2
233 {
234     \zcref {#2} ~
235     \zvpageref [ S=false, typeset=both, vcurrent={#1} ] {#2}
236 }
237 \cs_new_nopar:cpn
238 { \__zrefvario_ref_opt_default_cs:nn { vrefrangeformat } { nnn } } #1#2#3
239 {
240     \reftextlabelrange {#2} {#3} ~
241     \zvpagerefrange [ S=false, typeset=both, vcurrent={#1} ] {#2} {#3}
242 }
243 \cs_new_nopar:cpn
244 { \__zrefvario_ref_opt_default_cs:nn { fullrefformat } { n } } #1
245 {
246     \zcref {#1} ~
247     \zreftextfaraway [ S=false, typeset=both ] {#1}
248 }
249 \NewDocumentCommand \zreftextfaraway { s O { } m }
250 { \zref@wrapper@babel \__zrefvario_zreftextfaraway:nnn {#3} {#1} {#2} }
251 \cs_new_protected:Npn \__zrefvario_zreftextfaraway:nnn #1#2#3
252 {
253     \group_begin:
254     \keys_set:nn { zref-vario/ref } {#3}
255     \bool_if:nT {#2}
256     { \keys_set:nn { zref-vario/ref } { nohyperref } }
257     \__zrefvario_varioref_setup:
258     \reftextfaraway {#1}
259     \group_end:

```


260 }

Warnings and values for missing strings / unknown languages.

```
261 \group_begin:
262 \tl_set:Nn \l__zrefvario_setup_language_tl { zv_unknown_language }
263 \AddToHook { begindocument/before }
264 {
265   \keys_set:nn { zref-vario/langsetup }
266   {
267     retextfaceafter =
268     {
269       \msg_warning:nxxx { zref-vario } { missing-string }
270       { retextfaceafter } { \l__zrefvario_ref_language_tl }
271       \zref@default
272     } ,
273     retextfacebefore =
274     {
275       \msg_warning:nxxx { zref-vario } { missing-string }
276       { retextfacebefore } { \l__zrefvario_ref_language_tl }
277       \zref@default
278     } ,
279     retextafter =
280     {
281       \msg_warning:nxxx { zref-vario } { missing-string }
282       { retextafter } { \l__zrefvario_ref_language_tl }
283       \zref@default
284     } ,
285     retextbefore =
286     {
287       \msg_warning:nxxx { zref-vario } { missing-string }
288       { retextbefore } { \l__zrefvario_ref_language_tl }
289       \zref@default
290     } ,
291     retextcurrent =
292     {
293       \msg_warning:nxxx { zref-vario } { missing-string }
294       { retextcurrent } { \l__zrefvario_ref_language_tl }
295       \zref@default
296     } ,
297     retextfaraway =
298     {
299       \msg_warning:nxxx { zref-vario } { missing-string }
300       { retextfaraway } { \l__zrefvario_ref_language_tl }
301       \zref@default
302     } ,
303     retextpagerange =
304     {
305       \msg_warning:nxxx { zref-vario } { missing-string }
306       { retextpagerange } { \l__zrefvario_ref_language_tl }
307       \zref@default
308     } ,
309     retextlabelrange =
310     {
311       \msg_warning:nxxx { zref-vario } { missing-string }
```



```

312         { reftextlabelrange } { \l__zrefvario_ref_language_tl }
313         \zref@default
314     } ,
315 }
316 }
317 \group_end:
318 \msg_new:nnn { zref-vario } { missing-string }
319 { Missing~'#1'~string~for~language~'#2'~\msg_line_context:. }

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

320 \AddToHook { begindocument/before }
321 {
322     \AddToHook { begindocument }
323     {
324         \seq_if_in:NxF \g__zrefvario_setup_languages_seq
325         { \__zrefvario_base_lang:e { \l__zrefvario_ref_language_tl } }
326         {
327             \msg_warning:nnx { zref-vario } { language-not-setup }
328             { \l__zrefvario_ref_language_tl }
329         }
330     }
331 }
332 \msg_new:nnn { zref-vario } { language-not-setup }
333 {
334     Language~'#1'~is~not~set~up~for~'zref-vario'.~
335     See~documentation~for~'\iow_char:N\zvLanguageSetup'.
336 }

```

3 User interface

`__zrefvario_vref_pagenum:Nn` 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`.

```

337 \cs_new_protected:Npn \__zrefvario_vref_pagenum:Nn #1#2
338 {
339     \exp_args:NNNo \exp_args:NNo \tl_set:Nn #1
340     { \zref@extractdefault {#2} { page } { ?? } }
341 }
342 \cs_new_protected:Npn \__zrefvario_vrefpagenum:Nn #1#2
343 {
344     \zref@refused {#2}
345     \__zrefvario_vref_pagenum:Nn #1 {#2}
346 }
347 \cs_new_protected:Npn \__zrefvario_vref_label:n #1
348 { \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.

```

349 \bool_new:N \l__zrefvario_varioref_setup_bool

```



```

350 \cs_new_protected:Npn \__zrefvario_varioref_setup:
351 {
352   \bool_if:NF \l__zrefvario_varioref_setup_bool
353   {
354     \bool_set_true:N \l__zrefvario_varioref_setup_bool
355     \@vrefhandlespacefalse
356     \cs_set_eq:NN \vref@pagenum \__zrefvario_vref_pagenum:Nn
357     \cs_set_eq:NN \vrefpagenum \__zrefvario_vrefpagenum:Nn
358     \cs_set_eq:NN \vref@label \__zrefvario_vref_label:n
359     \__zrefvario_ref_opt_tl_get:xnN
360     { \l__zrefvario_ref_language_tl }
361     { retextfaceafter }
362     \retextfaceafter
363     \__zrefvario_ref_opt_tl_get:xnN
364     { \l__zrefvario_ref_language_tl }
365     { retextfacebefore }
366     \retextfacebefore
367     \__zrefvario_ref_opt_tl_get:xnN
368     { \l__zrefvario_ref_language_tl }
369     { retextafter }
370     \retextafter
371     \__zrefvario_ref_opt_tl_get:xnN
372     { \l__zrefvario_ref_language_tl }
373     { retextbefore }
374     \retextbefore
375     \__zrefvario_ref_opt_tl_get:xnN
376     { \l__zrefvario_ref_language_tl }
377     { retextcurrent }
378     \retextcurrent
379     \__zrefvario_ref_opt_cs_get:xnnN
380     { \l__zrefvario_ref_language_tl }
381     { retextfaraway } { n }
382     \retextfaraway
383     \__zrefvario_ref_opt_cs_get:xnnN
384     { \l__zrefvario_ref_language_tl }
385     { retextpagerange } { nn }
386     \retextpagerange
387     \__zrefvario_ref_opt_cs_get:xnnN
388     { \l__zrefvario_ref_language_tl }
389     { retextlabelrange } { nn }
390     \retextlabelrange
391     \__zrefvario_ref_opt_cs_get:xnnN
392     { \l__zrefvario_ref_language_tl }
393     { vrefformat } { nn }
394     \vrefformat
395     \__zrefvario_ref_opt_cs_get:xnnN
396     { \l__zrefvario_ref_language_tl }
397     { vrefrangeformat } { nnn }
398     \vrefrangeformat
399     \__zrefvario_ref_opt_cs_get:xnnN
400     { \l__zrefvario_ref_language_tl }
401     { fullrefformat } { n }
402     \fullrefformat
403   }

```


404 }

(End definition for _zrefvario_varioref_setup:.)

User commands.

```
\zvref      \zvref<*>[<options>]{<label>}
405 \NewDocumentCommand \zvref { s O { } m }
406 { \zref@wrapper@babel \_zrefvario_zvref:nnn {#3} {#1} {#2} }
407 \cs_new_protected:Npn \_zrefvario_zvref:nnn #1#2#3
408 {
409   \group_begin:
410   \keys_set:nn { zref-vario/ref-oneoptarg } {#3}
411   \bool_if:nT {#2}
412     { \keys_set:nn { zref-vario/ref } { nohyperref } }
413   \_zrefvario_varioref_setup:
414   \_zrefvario_vref:Vn \l__zrefvario_vref_reftextcurrent_tl {#1}
415   \group_end:
416 }
```

“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.

```
417 \cs_new_protected:Npn \_zrefvario_vref:nn #1#2
418 { \vrefformat {#1} {#2} }
419 \cs_generate_variant:Nn \_zrefvario_vref:nn { Vn }
```

(End definition for \zvref.)

```
\zvpageref      \zvpageref<*>[<options>]{<label>}
420 \NewDocumentCommand \zvpageref { s O { } m }
421 { \zref@wrapper@babel \_zrefvario_zvpageref:nnn {#3} {#1} {#2} }
422 \cs_new_protected:Npn \_zrefvario_zvpageref:nnn #1#2#3
423 {
424   \group_begin:
425   \keys_set:nn { zref-vario/ref-twooptargs } {#3}
426   \bool_if:nT {#2}
427     { \keys_set:nn { zref-vario/ref } { nohyperref } }
428   \_zrefvario_varioref_setup:
429   \_zrefvario_vpageref:VVn \l__zrefvario_reftextcurrent_tl
430     \l__zrefvario_reftextother_tl {#1}
431   \bool_lazy_and:nnT
432     { \g__zrefvario_zrefcheck_available_bool }
433     { \l__zrefvario_with_vcheck_bool }
434     {
435       \zrefcheck_zrefvario_label:
436       \vref@pagenum \l_tmpa_tl { \the\c@vrcnt @vr }
437       \vrefpagenum \l_tmpb_tl {#1}
438       \tl_if_eq:NNT \l_tmpa_tl \l_tmpb_tl
439       {
440         \zrefcheck_zrefvario_run_check_on_label:Vn
441         \l__zrefvario_vcheck_tl {#1}
442       }
443     }
444 }
```



```

444 \group_end:
445 }
446 \cs_new_protected:Npn \__zrefvario_vpageref:nnn #1#2#3
447 { \vpageref [{#1}] [{#2}] [{#3}] }
448 \cs_generate_variant:Nn \__zrefvario_vpageref:nnn { VVn }
(End definition for \zvpageref.)

```

```

\zvrefrange \zvrefrange<*>[<options>]{<label>}{<label>}
449 \NewDocumentCommand \zvrefrange { s O { } m m }
450 {
451 \zref@wrapper@babel
452 { \zref@wrapper@babel \__zrefvario_zvrefrange:nnnn {#3} }
453 {#4} {#1} {#2}
454 }
455 \cs_new_protected:Npn \__zrefvario_zvrefrange:nnnn #1#2#3#4
456 {
457 \group_begin:
458 \keys_set:nn { zref-vario/ref-oneoptarg } {#4}
459 \bool_if:nT {#3}
460 { \keys_set:nn { zref-vario/ref } { nohyperref } }
461 \__zrefvario_varioref_setup:
462 \__zrefvario_vrefrange:Vnn \l__zrefvario_reftextcurrent_tl {#1} {#2}
463 \group_end:
464 }
465 \cs_new_protected:Npn \__zrefvario_vrefrange:nnn #1#2#3
466 { \vrefrange [{#1}] [{#2}] [{#3}] }
467 \cs_generate_variant:Nn \__zrefvario_vrefrange:nnn { Vnn }
(End definition for \zvrefrange.)

```

```

\zvpagerefrange \zvpagerefrange<*>[<options>]{<label>}{<label>}
468 \NewDocumentCommand \zvpagerefrange { s O { } m m }
469 {
470 \zref@wrapper@babel
471 { \zref@wrapper@babel \__zrefvario_zvpagerefrange:nnnn {#3} }
472 {#4} {#1} {#2}
473 }
474 \cs_new_protected:Npn \__zrefvario_zvpagerefrange:nnnn #1#2#3#4
475 {
476 \group_begin:
477 \keys_set:nn { zref-vario/ref-oneoptarg } {#4}
478 \bool_if:nT {#3}
479 { \keys_set:nn { zref-vario/ref } { nohyperref } }
480 \__zrefvario_varioref_setup:
481 \__zrefvario_vpagerefrange:Vnn \l__zrefvario_reftextcurrent_tl {#1} {#2}
482 \bool_lazy_and:nnT
483 { \g__zrefvario_zrefcheck_available_bool }
484 { \l__zrefvario_with_vcheck_bool }
485 {
486 \zrefcheck_zrefvario_label:
487 \vrefpagenum \l_tmpa_tl {#1}
488 \vrefpagenum \l_tmpb_tl {#2}
489 \tl_if_eq:NNT \l_tmpa_tl \l_tmpb_tl

```



```

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

```

(End definition for \zvpagerefrange.)

```

\zfullref      \zfullref{<*>[<options>]}{<label>}}
506 \NewDocumentCommand \zfullref { s O { } m }
507 { \zref@wrapper@babel \__zrefvario_zfullref:nnn {#3} {#1} {#2} }
508 \cs_new_protected:Npn \__zrefvario_zfullref:nnn #1#2#3
509 {
510   \group_begin:
511   \keys_set:nn { zref-vario/ref } {#3}
512   \bool_if:nT {#2}
513   { \keys_set:nn { zref-vario/ref } { nohyperref } }
514   \__zrefvario_varioref_setup:
515   \fullref {#1}
516   \group_end:
517 }

```

(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.

```

518 \bool_new:N \g__zrefvario_zrefcheck_available_bool
519 \bool_new:N \l__zrefvario_with_vcheck_bool
520 \tl_new:N \l__zrefvario_vcheck_tl
521 \AddToHook { begindocument/before }
522 {
523   \IfPackageLoadedTF { zref-check }
524   {

```



```

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

```

5 Languages

```

561 \zvLanguageSetup { english }
562 {
563   retextfaceafter = {on~the~\reftextvario{facing}{next}~page} ,
564   retextfacebefore = {on~the~\reftextvario{facing}{preceding}~page} ,
565   retextafter = {on~the~\reftextvario{following}{next}~page} ,
566   retextbefore = {on~the~\reftextvario{preceding}{previous}~page} ,
567   retextcurrent = {on~\reftextvario{this}{the~current}~page} ,
568   retextfaraway = {on~\zcpageref{#1}} ,
569   retextpagerange = {on~\zcpageref[range]{#1,#2}} ,
570   retextlabelrange = {\zceref[range]{#1,#2}} ,
571 }
572 \zvLanguageSetup { german }
573 {
574   retextfaceafter = {auf~der~nächsten~Seite} ,

```



```

575     retextfacebefore = {auf-der~vorherigen-Seite} ,
576     retextafter = {auf-der~nächsten-Seite} ,
577     retextbefore = {auf-der~vorherigen-Seite} ,
578     retextcurrent = {auf~dieser~Seite} ,
579     retextfaraway = {auf~\zcpageref[d=D,g=f]{#1}} ,
580     retextpagerange = {auf~den~\zcpageref[range,d=D,g=f]{#1,#2}} ,
581     retextlabelrange = {\zcref[range]{#1,#2}} ,
582 }

583 \zvLanguageSetup { french }
584 {
585     retextfaceafter = {page~\retextvario{ci-contre}{suivante}} ,
586     retextfacebefore = {page~\retextvario{ci-contre}{précédente}} ,
587     retextafter = {page~suivante} ,
588     retextbefore = {page~précédente} ,
589     retextcurrent = {de-la~présente~page} ,
590     retextfaraway = {\zcpageref[g=f]{#1}} ,
591     retextpagerange = {\zcpageref[range,g=f]{#1,#2}} ,
592     retextlabelrange = {\zcref[range]{#1,#2}} ,
593 }

594 \zvLanguageSetup { portuguese }
595 {
596     retextfaceafter = {na~\retextvario{página-oposta}{próxima-página}} ,
597     retextfacebefore = {na~página~\retextvario{oposta}{anterior}} ,
598     retextafter = {na~\retextvario{página-seguinte}{próxima-página}} ,
599     retextbefore = {na~página~\retextvario{anterior}{precedente}} ,
600     retextcurrent = {nesta~página} ,
601     retextfaraway = {na~\zcpageref[g=f]{#1}} ,
602     retextpagerange = {nas~\zcpageref[range,g=f]{#1,#2}} ,
603     retextlabelrange = {\zcref[range]{#1,#2}} ,
604 }

605 \zvLanguageSetup { spanish }
606 {
607     retextfaceafter = {en-la~página-siguiente} ,
608     retextfacebefore = {en-la~página-anterior} ,
609     retextafter = {en-la~página-siguiente} ,
610     retextbefore = {en-la~página-anterior} ,
611     retextcurrent = {en-esta~página} ,
612     retextfaraway = {en-la~\zcpageref[g=f]{#1}} ,
613     retextpagerange = {en-las~\zcpageref[range,g=f]{#1,#2}} ,
614     retextlabelrange = {\zcref[range]{#1,#2}} ,
615 }

616 \zvLanguageSetup { dutch }
617 {
618     retextfaceafter = {op-de~\retextvario{rechter~pagina}{pagina~hiernaast}} ,
619     retextfacebefore = {op-de~\retextvario{linker~pagina}{pagina~hiernaast}} ,
620     retextafter = {op-de~\retextvario{volgende~pagina}{pagina~hierna}} ,
621     retextbefore = {op-de~\retextvario{vorige~pagina}{pagina~hiervoor}} ,
622     retextcurrent = {op-deze~pagina} ,
623     retextfaraway = {op~\zcpageref[g=f]{#1}} ,
624     retextpagerange = {op~\zcpageref[range,g=f]{#1,#2}} ,
625     retextlabelrange = {\zcref[range]{#1,#2}} ,
626 }

```


627 </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>\AddToHook</code>	<u>5</u> , <u>97</u> , <u>263</u> , <u>320</u> , <u>322</u> , <u>521</u>
B	
bool commands:	
<code>\bool_gset_false:N</code>	<u>539</u> , <u>548</u>
<code>\bool_gset_true:N</code>	<u>527</u>
<code>\bool_if:NTF</code>	<u>352</u>
<code>\bool_if:nTF</code> <u>255</u> , <u>411</u> , <u>426</u> , <u>459</u> , <u>478</u> , <u>512</u>	
<code>\bool_lazy_and:nnTF</code>	<u>431</u> , <u>482</u>
<code>\bool_new:N</code>	<u>349</u> , <u>518</u> , <u>519</u>
<code>\bool_set_true:N</code>	<u>354</u> , <u>533</u>
C	
cs commands:	
<code>\cs_generate_variant:Nn</code>	<u>18</u> , <u>23</u> , <u>26</u> , <u>40</u> , <u>65</u> , <u>419</u> , <u>448</u> , <u>467</u> , <u>505</u>
<code>\cs_gset_nopar:Npn</code>	<u>166</u> , <u>177</u> , <u>188</u> , <u>199</u> , <u>210</u> , <u>221</u>
<code>\cs_if_exist:NTF</code>	<u>43</u> , <u>50</u>
<code>\cs_new:Npn</code>	<u>10</u> , <u>12</u> , <u>21</u> , <u>24</u>
<code>\cs_new_nopar:Npn</code>	<u>231</u> , <u>237</u> , <u>243</u>
<code>\cs_new_protected:Npn</code>	<u>27</u> , <u>41</u> , <u>251</u> , <u>337</u> , <u>342</u> , <u>347</u> , <u>350</u> , <u>407</u> , <u>417</u> , <u>422</u> , <u>446</u> , <u>455</u> , <u>465</u> , <u>474</u> , <u>503</u> , <u>508</u>
<code>\cs_set_eq:NN</code> <u>46</u> , <u>53</u> , <u>57</u> , <u>356</u> , <u>357</u> , <u>358</u>	
E	
exp commands:	
<code>\exp_args:NNNo</code>	<u>339</u>
<code>\exp_args:NNo</code>	<u>339</u>
F	
<code>\fullref</code>	<u>4</u> , <u>515</u>
<code>\fullreformat</code>	<u>402</u>
G	
group commands:	
<code>\group_begin:</code>	<u>99</u> , <u>253</u> , <u>261</u> , <u>409</u> , <u>424</u> , <u>457</u> , <u>476</u> , <u>510</u>
<code>\group_end:</code>	<u>108</u> , <u>259</u> , <u>317</u> , <u>415</u> , <u>444</u> , <u>463</u> , <u>501</u> , <u>516</u>
I	
<code>\IfPackageAtLeastTF</code>	<u>525</u>
<code>\IfPackageLoadedTF</code>	<u>523</u>
iow commands:	
<code>\iow_char:N</code>	<u>335</u>
K	
keys commands:	
<code>\l_keys_choice_tl</code>	<u>534</u>
<code>\keys_define:nn</code>	<u>70</u> , <u>79</u> , <u>84</u> , <u>117</u> , <u>528</u> , <u>540</u> , <u>549</u>
<code>\keys_set:nn</code>	<u>103</u> , <u>254</u> , <u>256</u> , <u>265</u> , <u>410</u> , <u>412</u> , <u>425</u> , <u>427</u> , <u>458</u> , <u>460</u> , <u>477</u> , <u>479</u> , <u>511</u> , <u>513</u>
M	
msg commands:	
<code>\msg_line_context:</code>	<u>113</u> , <u>319</u>
<code>\msg_new:nnn</code>	<u>111</u> , <u>318</u> , <u>332</u> , <u>556</u>
<code>\msg_warning:nn</code>	<u>543</u> , <u>552</u>
<code>\msg_warning:nnn</code>	<u>107</u> , <u>327</u>
<code>\msg_warning:nnnn</code>	<u>269</u> , <u>275</u> , <u>281</u> , <u>287</u> , <u>293</u> , <u>299</u> , <u>305</u> , <u>311</u>
N	
<code>\NewDocumentCommand</code>	<u>95</u> , <u>249</u> , <u>405</u> , <u>420</u> , <u>449</u> , <u>468</u> , <u>506</u>
P	
<code>\ProvidesExplPackage</code>	<u>3</u>
R	
<code>\reftextafter</code>	<u>370</u>
<code>\reftextbefore</code>	<u>374</u>
<code>\reftextcurrent</code>	<u>4</u> , <u>67</u> , <u>378</u>
<code>\reftextfaceafter</code>	<u>362</u>
<code>\reftextfacebefore</code>	<u>366</u>
<code>\reftextfaraway</code>	<u>258</u> , <u>382</u>
<code>\reftextlabelrange</code>	<u>240</u> , <u>390</u>
<code>\reftextpagerange</code>	<u>386</u>
<code>\reftextvario</code>	<u>563</u> , <u>564</u> , <u>565</u> , <u>566</u> , <u>567</u> , <u>585</u> , <u>586</u> , <u>596</u> , <u>597</u> , <u>598</u> , <u>599</u> , <u>618</u> , <u>619</u> , <u>620</u> , <u>621</u>
<code>\RequirePackage</code>	<u>7</u> , <u>8</u>
S	
seq commands:	
<code>\seq_gput_right:Nn</code>	<u>104</u>
<code>\seq_if_in:NnTF</code>	<u>324</u>
<code>\seq_new:N</code>	<u>94</u>
T	
T _E X and L ^A T _E X 2 _ε commands:	
<code>\@onlypreamble</code>	<u>116</u>
<code>\@vrefhandlespace</code>	<u>11</u>
<code>\@vrefhandlespacefalse</code>	<u>355</u>

<code>\c@vrcnt</code>	436, 491	<code>\l__zrefvario_ref_language_tl</code> ...	
<code>\vr@f</code>	11	. 19, 270, 276, 282, 288, 294, 300,	
<code>\vref@label</code>	9, 358	306, 312, 325, 328, 360, 364, 368,	
<code>\vref@pagenum</code>	9, 356, 436, 491	372, 376, 380, 384, 388, 392, 396, 400	
<code>\vref@star</code>	11	<code>__zrefvario_ref_opt_cs_get:nnnN</code>	
<code>\zref@default</code> 3, 7, 27, 379, 383, 387, 391, 395, 399	
271, 277, 283, 289, 295, 301, 307, 313		<code>__zrefvario_ref_opt_default_-</code>	
<code>\zref@extractdefault</code>	340	cs:nn ... 2, 10, 51, 54, 232, 238, 244	
<code>\zref@labelbyprops</code>	348	<code>__zrefvario_ref_opt_name_lang_-</code>	
<code>\zref@refused</code>	344	cs:nnn	3, 24,
<code>\zref@wrapper@babel</code>		44, 47, 59, 168, 179, 190, 201, 212, 223	
250, 406, 421, 451, 452, 470, 471, 507		<code>__zrefvario_ref_opt_name_lang_-</code>	
<code>\the</code>	436, 491	tl:nn	2,
tl commands:		21, 30, 33, 37, 123, 132, 141, 150, 159	
<code>\tl_gset:Nn</code> ...	121, 130, 139, 148, 157	<code>__zrefvario_ref_opt_tl_get:nnN</code> .	
<code>\tl_if_eq:NNTF</code>	438, 489, 492	3, 27, 359, 363, 367, 371, 375
<code>\tl_if_exist:NTF</code>	29	<code>\l__zrefvario_ref_textcurrent_tl</code> .	
<code>\tl_new:N</code>	19, 66, 68, 69, 93, 520	4, 66, 67, 74, 429, 462, 481
<code>\tl_set:Nn</code>		<code>\l__zrefvario_ref_textother_tl</code> ...	
.....	20, 67, 74, 75, 102, 262, 339, 534	69, 81, 430
<code>\tl_set_eq:NN</code>	32, 36	<code>\l__zrefvario_setup_language_tl</code> .	
<code>\tl_use:N</code>	15	93, 102, 124, 133, 142, 151,
<code>\l_tmpa_tl</code> .	436, 438, 487, 489, 491, 492	160, 169, 180, 191, 202, 213, 224, 262	
<code>\l_tmpb_tl</code>	437, 438, 488, 489, 492	<code>\g__zrefvario_setup_languages_-</code>	
		seq	94, 104, 324
		<code>__zrefvario_varioref_setup:</code> ...	
		257, 349, 413, 428, 461, 480, 514
		<code>\l__zrefvario_varioref_setup_-</code>	
		bool	9, 349, 352, 354
		<code>\l__zrefvario_vcheck_tl</code>	
		441, 495, 497, 520, 534
		<code>__zrefvario_vpageref:nnn</code>	
		429, 446, 448
		<code>__zrefvario_vpagerefrange:nnn</code> ..	
		481, 503, 505
		<code>__zrefvario_vref:nn</code> ..	414, 417, 419
		<code>__zrefvario_vref_label:n</code> ..	337, 358
		<code>__zrefvario_vref_pagenum:Nn</code>	337, 356
		<code>\l__zrefvario_vref_ref_textcurrent_-</code>	
		tl	4, 68, 75, 414
		<code>__zrefvario_vref_pagenum:Nn</code>	337, 357
		<code>__zrefvario_vrefrange:nnn</code>	
		462, 465, 467
		<code>\l__zrefvario_with_vcheck_bool</code> ..	
		433, 484, 519, 533
		<code>__zrefvario_zfullref:nnn</code> ..	507, 508
		<code>\g__zrefvario_zrefcheck_available_-</code>	
		bool ...	432, 483, 518, 527, 539, 548
		<code>__zrefvario_zref_textfaraway:nnn</code>	
		250, 251
		<code>__zrefvario_zvpageref:nnn</code> .	421, 422
		<code>__zrefvario_zvpagerefrange:nnnn</code>	
		471, 474
		<code>__zrefvario_zvref:nnn</code>	406, 407

V

<code>\vpageref</code>	4, 447
<code>\vpagerefrange</code>	4, 504
<code>\vref</code>	4, 11
<code>\vrefformat</code>	11, 394, 418
<code>\vrefpagenum</code>	9, 357, 437, 487, 488
<code>\vrefrange</code>	4, 466
<code>\vrefrangeformat</code>	398

Z

<code>\zcheck</code>	13
<code>\zcpageref</code>	4, 568, 569, 579, 580,
590, 591, 601, 602, 612, 613, 623, 624	
<code>\zcref</code>	3, 4,
234, 246, 570, 581, 592, 603, 614, 625	
<code>\zfullref</code>	7, 13, 506
zrefcheck commands:	
<code>\zrefcheck_zrefvario_label:</code>	435, 486
<code>\zrefcheck_zrefvario_run_check_-</code>	
on_label:nn	440, 494, 496
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>\zref_textfaraway</code>	7, 247, 249
zrefvario internal commands:	
<code>__zrefvario_base_lang:n</code>	
.....	2, 12, 22, 25, 105, 325

_zrefvarto_zvrefrange:nnnn	452, 455	\zvpagerefrange	12, 241, 468
\zvLanguageSetup	...	\zvref	11, 405
...	4, 93, 561, 572, 583, 594, 605, 616	\zvrefrange	12, 449
\zvpageref	4, 7, 11, 235, 420		