

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

Code documentation

Gustavo Barros[†]

2022-04-24

EXPERIMENTAL

Contents

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

^{*}This file describes v0.1.4, released 2022-04-24.

[†]<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} {2022-04-24} {0.1.4}
4   {Extended LaTeX page cross-references with varioref and zref-clever}

```

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

```

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

```

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}>
8 \cs_new:Npn \__zrefvario_ref_opt_default_cs:nn #1#2
9   { __zrefvario_ref_opt_default_ #1 : #2 }

```

(End definition for `__zrefvario_ref_opt_default_cs:nn`.)

```

\__zrefvario_base_lang:n               \__zrefvario_base_lang:n <{language}>
10 \cs_new:Npn \__zrefvario_base_lang:n #1
11   {
12     \zrefclever_language_if_declared:nTF {#1}
13       { \tl_use:c { \zrefclever_language_varname:n {#1} } }
14       { zv_unknown_language }
15   }
16 \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.

```

17 \tl_new:N \l_zrefvario_ref_language_tl
18 \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}>
19 \cs_new:Npn \__zrefvario_ref_opt_name_lang_tl:nn #1#2
20   { g__zrefvario_ref_opt_lang_ \__zrefvario_base_lang:n {#1} _ #2 _tl }
21 \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}}
22 \cs_new:Npn \__zrefvario_ref_opt_name_lang_cs:nnn #1#2#3
23   { \__zrefvario_ref_opt_lang_ \__zrefvario_base_lang:n {#1} _ #2 : #3 }
24 \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 \__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}
{\cs signature} {\cs to set}}
25 \cs_new_protected:Npn \__zrefvario_ref_opt_tl_get:nnN #1#2#3
26   {
27     \tl_if_exist:cTF
28       { \__zrefvario_ref_opt_name_lang_tl:nn {#1} {#2} }
29       {
30         \tl_set_eq:Nc #3
31         { \__zrefvario_ref_opt_name_lang_tl:nn {#1} {#2} }
32       }
33       {
34         \tl_set_eq:Nc #3
35         { \__zrefvario_ref_opt_name_lang_tl:nn { zv_unknown_language } {#2} }
36       }
37   }
38 \cs_generate_variant:Nn \__zrefvario_ref_opt_tl_get:nnN { xnN }
39 \cs_new_protected:Npn \__zrefvario_ref_opt_cs_get:nnnN #1#2#3#4
40   {
41     \cs_if_exist:cTF
42       { \__zrefvario_ref_opt_name_lang_cs:nnn {#1} {#2} {#3} }
43       {
44         \cs_set_eq:Nc #4
45         { \__zrefvario_ref_opt_name_lang_cs:nnn {#1} {#2} {#3} }
46       }
47       {
48         \cs_if_exist:cTF
49           { \__zrefvario_ref_opt_default_cs:nn {#2} {#3} }
50           {
51             \cs_set_eq:Nc #4
52             { \__zrefvario_ref_opt_default_cs:nn {#2} {#3} }
53           }
54           {
55             \cs_set_eq:Nc #4
56             {
57               \__zrefvario_ref_opt_name_lang_cs:nnn
58               { zv_unknown_language } {#2} {#3}
59             }
60           }
61       }
62   }
63 \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.

```

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

```

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

```

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

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

```

```

103         { \_zrefvario_base_lang:n {#1} }
104     }
105     { \msg_warning:nnn { zref-vario } { unknown-language-setup } {#1} }
106     \group_end:
107 }
108 }
109 \msg_new:nnn { zref-vario } { unknown-language-setup }
110 {
111     Language~'~#1'~is-unknown~\msg_line_context:~Can't~set~it~up.~
112     The~language~must~be~declared~for~'zref-clever',~see~its~documentation.
113 }
114 \@onlypreamble \zvLanguageSetup

```

(End definition for \zvLanguageSetup.)

```

115 \cs_new_protected:Npn \_zrefvario_opt_tl_gset:Nn #1#2
116 {
117     \tl_if_exist:NF #1
118     { \tl_new:N #1 }
119     \tl_gset:Nn #1 {#2}
120 }
121 \cs_generate_variant:Nn \_zrefvario_opt_tl_gset:Nn { cn }
122 \keys_define:nn { zref-vario/langsetup }
123 {
124     retextfaceafter .code:n =
125     {
126         \_zrefvario_opt_tl_gset:cn
127         {
128             \_zrefvario_ref_opt_name_lang_tl:en
129             { \l__zrefvario_setup_language_tl }
130             { retextfaceafter }
131         } {#1}
132     } ,
133     retextfacebefore .code:n =
134     {
135         \_zrefvario_opt_tl_gset:cn
136         {
137             \_zrefvario_ref_opt_name_lang_tl:en
138             { \l__zrefvario_setup_language_tl }
139             { retextfacebefore }
140         } {#1}
141     } ,
142     retextafter .code:n =
143     {
144         \_zrefvario_opt_tl_gset:cn
145         {
146             \_zrefvario_ref_opt_name_lang_tl:en
147             { \l__zrefvario_setup_language_tl }
148             { retextafter }
149         } {#1}
150     } ,
151     retextbefore .code:n =
152     {
153         \_zrefvario_opt_tl_gset:cn
154         {

```

```

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

```

```

209     }
210     ##1##2
211     {#1}
212   } ,
213   vrefrangeformat .code:n =
214   {
215     \cs_gset_nopar:cpn
216     {
217       \__zrefvario_ref_opt_name_lang_cs:enn
218       { \l__zrefvario_setup_language_tl }
219       { vrefrangeformat } { nnn }
220     }
221     ##1##2##3
222     {#1}
223   } ,
224   fullreformat .code:n =
225   {
226     \cs_gset_nopar:cpn
227     {
228       \__zrefvario_ref_opt_name_lang_cs:enn
229       { \l__zrefvario_setup_language_tl }
230       { fullreformat } { n }
231     }
232     ##1
233     {#1}
234   } ,
235 }

```

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).

```

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

```

```

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

```

Warnings and values for missing strings / unknown languages.

```

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

```



```

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

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

325 \AddToHook { begindocument/before }
326 {
327     \AddToHook { begindocument }
328     {
329         \seq_if_in:NxF \g__zrefvario_setup_languages_seq
330         { \__zrefvario_base_lang:e { \l__zrefvario_ref_language_tl } }
331         {
332             \msg_warning:nxx { zref-vario } { language-not-setup }
333             { \l__zrefvario_ref_language_tl }
334         }
335     }
336 }
337 \msg_new:nnn { zref-vario } { language-not-setup }
338 {
339     Language~'#1'~is-not-set-up~for~'zref-vario'~
340     See~documentation~for~'\iow_char:N\zvLanguageSetup'.
341 }

```

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`.

```

342 \cs_new_protected:Npn \__zrefvario_vref_pagenum:Nn #1#2
343 {
344     \exp_args:NNNo \exp_args:NNo \cs_set_nopar:Npn #1
345     { \zref@extractdefault {#2} { page } { ?? } }
346 }
347 \cs_new_protected:Npn \__zrefvario_vrefpagenum:Nn #1#2
348 {
349     \zref@refused {#2}
350     \__zrefvario_vref_pagenum:Nn #1 {#2}
351 }
352 \cs_new_protected:Npn \__zrefvario_vref_label:n #1
353 { \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.

```

354 \bool_new:N \l\_zrefvario_varioref_setup_bool
355 \cs_new_protected:Npn \_zrefvario_varioref_setup:
356 {
357   \bool_if:NF \l\_zrefvario_varioref_setup_bool
358   {
359     \bool_set_true:N \l\_zrefvario_varioref_setup_bool
360     \@vrefhandlespacefalse
361     \cs_set_eq:NN \vref@pagenum \_zrefvario_vref_pagenum:Nn
362     \cs_set_eq:NN \vrefpagenum \_zrefvario_vrefpagenum:Nn
363     \cs_set_eq:NN \vref@label \_zrefvario_vref_label:n
364     \_zrefvario_ref_opt_tl_get:xnN
365       { \l\_zrefvario_ref_language_tl }
366       { retextfaceafter }
367       \retextfaceafter
368     \_zrefvario_ref_opt_tl_get:xnN
369       { \l\_zrefvario_ref_language_tl }
370       { retextfacebefore }
371       \retextfacebefore
372     \_zrefvario_ref_opt_tl_get:xnN
373       { \l\_zrefvario_ref_language_tl }
374       { retextafter }
375       \retextafter
376     \_zrefvario_ref_opt_tl_get:xnN
377       { \l\_zrefvario_ref_language_tl }
378       { retextbefore }
379       \retextbefore
380     \_zrefvario_ref_opt_tl_get:xnN
381       { \l\_zrefvario_ref_language_tl }
382       { retextcurrent }
383       \retextcurrent
384     \_zrefvario_ref_opt_cs_get:xnnN
385       { \l\_zrefvario_ref_language_tl }
386       { retextfaraway } { n }
387       \retextfaraway
388     \_zrefvario_ref_opt_cs_get:xnnN
389       { \l\_zrefvario_ref_language_tl }
390       { retextpagerange } { nn }
391       \retextpagerange
392     \_zrefvario_ref_opt_cs_get:xnnN
393       { \l\_zrefvario_ref_language_tl }
394       { retextlabelrange } { nn }
395       \retextlabelrange
396     \_zrefvario_ref_opt_cs_get:xnnN
397       { \l\_zrefvario_ref_language_tl }
398       { vrefformat } { nn }
399       \vrefformat
400     \_zrefvario_ref_opt_cs_get:xnnN
401       { \l\_zrefvario_ref_language_tl }

```

```

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

```

(End definition for __zrefvario_varioref_setup.)
User commands.

```

\zvref      \zvref{<*>[<options>]}{<label>}
410 \NewDocumentCommand \zvref { s 0 { } m }
411 { \zref@wrapper@babel \__zrefvario_zvref:nnn {#3} {#1} {#2} }
412 \cs_new_protected:Npn \__zrefvario_zvref:nnn #1#2#3
413 {
414   \group_begin:
415   \keys_set:nn { zref-vario/ref-oneoptarg } {#3}
416   \bool_if:nT {#2}
417     { \keys_set:nn { zref-vario/ref } { nohyperref } }
418   \__zrefvario_varioref_setup:
419   \__zrefvario_vref:Vn \l__zrefvario_vref_reftextcurrent_tl {#1}
420   \group_end:
421 }

```

“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.

```

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

```

(End definition for \zvref.)

```

\zvpageref      \zvpageref{<*>[<options>]}{<label>}
425 \NewDocumentCommand \zvpageref { s 0 { } m }
426 { \zref@wrapper@babel \__zrefvario_zvpageref:nnn {#3} {#1} {#2} }
427 \cs_new_protected:Npn \__zrefvario_zvpageref:nnn #1#2#3
428 {
429   \group_begin:
430   \keys_set:nn { zref-vario/ref-twooptargs } {#3}
431   \bool_if:nT {#2}
432     { \keys_set:nn { zref-vario/ref } { nohyperref } }
433   \__zrefvario_varioref_setup:
434   \__zrefvario_vpageref:Vn \l__zrefvario_reftextcurrent_tl
435     \l__zrefvario_reftextother_tl {#1}
436   \bool_lazy_and:nnT
437     { \g__zrefvario_zrefcheck_available_bool }
438     { \l__zrefvario_with_vcheck_bool }
439     {
440       \zrefcheck_zrefvario_label:
441       \vref@pagenum \l_tmpa_tl { \the\c@vrcnt @vr }

```

```

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

```

(End definition for \zvpageref.)

```

\zvrefrange \zvrefrange{*}[\langle options \rangle]{\langle label \rangle}{\langle label \rangle}
454 \NewDocumentCommand \zvrefrange { s O { } m m }
455 {
456   \zref@wrapper@babel
457   { \zref@wrapper@babel \__zrefvario_zvrefrange:nnnn {#3} }
458   {#4} {#1} {#2}
459 }
460 \cs_new_protected:Npn \__zrefvario_zvrefrange:nnnn #1#2#3#4
461 {
462   \group_begin:
463   \keys_set:nn { zref-vario/ref-oneoptarg } {#4}
464   \bool_if:nT {#3}
465   { \keys_set:nn { zref-vario/ref } { nohyperref } }
466   \__zrefvario_varioref_setup:
467   \__zrefvario_vrefrange:Vnn \l__zrefvario_reftextcurrent_tl {#1} {#2}
468   \group_end:
469 }
470 \cs_new_protected:Npn \__zrefvario_vrefrange:nnn #1#2#3
471 { \vrefrange [{#1}] [{#2}] [{#3}] }
472 \cs_generate_variant:Nn \__zrefvario_vrefrange:nnn { Vnn }

```

(End definition for \zvrefrange.)

```

\zvpagerefrange \zvpagerefrange{*}[\langle options \rangle]{\langle label \rangle}{\langle label \rangle}
473 \NewDocumentCommand \zvpagerefrange { s O { } m m }
474 {
475   \zref@wrapper@babel
476   { \zref@wrapper@babel \__zrefvario_zvpagerefrange:nnnn {#3} }
477   {#4} {#1} {#2}
478 }
479 \cs_new_protected:Npn \__zrefvario_zvpagerefrange:nnnn #1#2#3#4
480 {
481   \group_begin:
482   \keys_set:nn { zref-vario/ref-oneoptarg } {#4}
483   \bool_if:nT {#3}
484   { \keys_set:nn { zref-vario/ref } { nohyperref } }
485   \__zrefvario_varioref_setup:
486   \__zrefvario_vpagerefrange:Vnn \l__zrefvario_reftextcurrent_tl {#1} {#2}
487   \bool_lazy_and:nnT

```

```

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

```

(End definition for \zvpagerefrange.)

```

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

```

(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.

```

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

```

5 Languages

```

566 \zvLanguageSetup { english }
567 {
568   retextfaceafter = {on~the~\retextvario{facing}{next}~page} ,
569   retextfacebefore = {on~the~\retextvario{facing}{preceding}~page} ,
570   retextafter = {on~the~\retextvario{following}{next}~page} ,
571   retextbefore = {on~the~\retextvario{preceding}{previous}~page} ,
572   retextcurrent = {on~\retextvario{this}{the~current}~page} ,

```

```

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

```

```

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

```

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

A		G	
\AddToHook	6, 95, 268, 325, 327, 526	group commands:	
B		\group_begin:	97, 258, 266, 414, 429, 462, 481, 515
bool commands:		\group_end:	106, 264, 322, 420, 449, 468, 506, 521
\bool_gset_false:N	544, 553	I	
\bool_gset_true:N	532	\IfPackageAtLeastTF	530
\bool_if:NTF	357	\IfPackageLoadedTF	528
\bool_if:nTF	260, 416, 431, 464, 483, 517	iow commands:	
\bool_lazy_and:nnTF	436, 487	\iow_char:N	340
\bool_new:N	354, 523, 524	K	
\bool_set_true:N	359, 538	keys commands:	
C		\l_keys_choice_tl	539
cs commands:		\keys_define:nn	68, 77, 82, 122, 533, 545, 554
\cs_generate_variant:Nn	16, 21, 24, 38, 63, 121, 424, 453, 472, 510	\keys_set:nn	101, 259, 261, 270, 415, 417, 430, 432, 463, 465, 482, 484, 516, 518
\cs_gset_nopar:Npn	171, 182, 193, 204, 215, 226	M	
\cs_if_exist:NTF	41, 48	msg commands:	
\cs_new:Npn	8, 10, 19, 22	\msg_line_context:	111, 324
\cs_new_nopar:Npn	236, 242, 248	\msg_new:nnn	109, 323, 337, 561
\cs_new_protected:Npn	25, 39, 115, 256, 342, 347, 352, 355, 412, 422, 427, 451, 460, 470, 479, 508, 513	\msg_warning:nn	548, 557
\cs_set_eq:NN	44, 51, 55, 361, 362, 363	\msg_warning:nnn	105, 332
\cs_set_nopar:Npn	344	\msg_warning:nnnn	274, 280, 286, 292, 298, 304, 310, 316
E		N	
exp commands:		\NewDocumentCommand	93, 254, 410, 425, 454, 473, 511
\exp_args:NNNo	344	P	
\exp_args:NNo	344	\ProvidesExplPackage	3
F			
\fullref	4, 520		
\fullrefformat	407		

R		<code>\vrefpagenum</code> 9, 362, 442, 492, 493
<code>\reftextafter</code>	375	<code>\vrefrange</code> 4, 471
<code>\reftextbefore</code>	379	<code>\vrefrangeformat</code> 403
<code>\reftextcurrent</code>	4, 65, 383	
<code>\reftextfaceafter</code>	367	Z
<code>\reftextfacebefore</code>	371	<code>\zcheck</code> 13
<code>\reftextfaraway</code>	263, 387	<code>\zcpageref</code> 4, 573, 574, 584, 585,
<code>\reftextlabelrange</code>	245, 395	595, 596, 606, 607, 617, 618, 628, 629
<code>\reftextpagerange</code>	391	<code>\zceref</code> 3, 4,
<code>\reftextvario</code>	568, 569, 570, 571, 572, 590, 591,	239, 251, 575, 586, 597, 608, 619, 630
	601, 602, 603, 604, 623, 624, 625, 626	<code>\zfullref</code> 7, 13, 511
<code>\RequirePackage</code>	5, 7	zrefcheck commands:
S		<code>\zrefcheck_zrefvario_label:</code> 440, 491
seq commands:		<code>\zrefcheck_zrefvario_run_check_-</code>
<code>\seq_gput_right:Nn</code>	102	<code>on_label:nn</code> 445, 499, 501
<code>\seq_if_in:NnTF</code>	329	zrefclever commands:
<code>\seq_new:N</code>	92	<code>\zrefclever_language_if_declared:nTF</code>
T	 12, 98
$\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}_{2\epsilon}$ commands:		<code>\zrefclever_language_varname:n</code> . . 13
<code>\@onlypreamble</code>	114	<code>\l_zrefclever_ref_language_tl</code> . 2, 18
<code>\@vrefhandlespace</code>	11	<code>\zreftextfaraway</code> 7, 252, 254
<code>\@vrefhandlespacefalse</code>	360	zrefvario internal commands:
<code>\c@vrcnt</code>	441, 496	<code>__zrefvario_base_lang:n</code>
<code>\vr@f</code>	11 2, 10, 20, 23, 103, 330
<code>\vref@label</code>	9, 363	<code>__zrefvario_opt_tl_gset:Nn</code>
<code>\vref@pagenum</code>	9, 361, 441, 496 115, 121, 126, 135, 144, 153, 162
<code>\vref@star</code>	11	<code>\l__zrefvario_ref_language_tl</code>
<code>\zref@default</code>	276, 282, 288, 294, 300, 306, 312, 318	. 17, 275, 281, 287, 293, 299, 305,
<code>\zref@extractdefault</code>	345	311, 317, 330, 333, 365, 369, 373,
<code>\zref@labelbyprops</code>	353	377, 381, 385, 389, 393, 397, 401, 405
<code>\zref@refused</code>	349	<code>__zrefvario_ref_opt_cs_get:nnnN</code>
<code>\zref@wrapper@babel</code>	255, 411, 426, 456, 457, 475, 476, 512	. 3, 7, 25, 384, 388, 392, 396, 400, 404
<code>\the</code>	441, 496	<code>__zrefvario_ref_opt_default_-</code>
tl commands:		<code>cs:nn</code> 2, 8, 49, 52, 237, 243, 249
<code>\tl_gset:Nn</code>	119	<code>__zrefvario_ref_opt_name_lang_-</code>
<code>\tl_if_eq:NNTF</code>	443, 494, 497	<code>cs:nnn</code> 3, 22,
<code>\tl_if_exist:NNTF</code>	27, 117	42, 45, 57, 173, 184, 195, 206, 217, 228
<code>\tl_new:N</code> 17, 64, 66, 67, 91, 118, 525		<code>__zrefvario_ref_opt_name_lang_-</code>
<code>\tl_set:Nn</code> 18, 65, 72, 73, 100, 267, 539		<code>tl:nn</code> 2,
<code>\tl_set_eq:NN</code>	30, 34	19, 28, 31, 35, 128, 137, 146, 155, 164
<code>\tl_use:N</code>	13	<code>__zrefvario_ref_opt_tl_get:nnN</code>
<code>\l_tmpa_tl</code>	441, 443, 492, 494, 496, 497 3, 25, 364, 368, 372, 376, 380
<code>\l_tmpb_tl</code>	442, 443, 493, 494, 497	<code>\l__zrefvario_reftextcurrent_tl</code>
V	 4, 64, 65, 72, 434, 467, 486
<code>\vpageref</code>	4, 452	<code>\l__zrefvario_reftextother_tl</code>
<code>\vpagerefrange</code>	4, 509 67, 79, 435
<code>\vref</code>	4, 11	<code>\l__zrefvario_setup_language_tl</code>
<code>\vrefformat</code>	11, 399, 423 91, 100, 129, 138, 147, 156,
		165, 174, 185, 196, 207, 218, 229, 267
		<code>\g__zrefvario_setup_languages_-</code>
		<code>seq</code> 92, 102, 329
		<code>__zrefvario_varioref_setup:</code>
	 262, 354, 418, 433, 466, 485, 519

\l_zrefvario_varioref_setup_- 438, 489, 524, 538
bool	10, 354, 357, 359
\l_zrefvario_vcheck_tl
.....	446, 500, 502, 525, 539
_zrefvario_vpageref:nnn
.....	434, 451, 453
_zrefvario_vpagerefrange:nnn
.....	486, 508, 510
_zrefvario_vref:nn ..	419, 422, 424
_zrefvario_vref_label:n ..	342, 363
_zrefvario_vref_pagenum:Nn	342, 361
\l_zrefvario_vref_reftextcurrent_-
tl	4, 66, 73, 419
_zrefvario_vrefpagenum:Nn	342, 362
_zrefvario_vrefrange:nnn
.....	467, 470, 472
\l_zrefvario_with_vcheck_bool
_zrefvario_zfullref:nnn ..	512, 513
\g_zrefvario_zrefcheck_available_-
bool	437, 488, 523, 532, 544, 553
_zrefvario_zreftextfaraway:nnn
.....	255, 256
_zrefvario_zvpageref:nnn ..	426, 427
_zrefvario_zvpagerefrange:nnnn
.....	476, 479
_zrefvario_zvref:nnn	411, 412
_zrefvario_zvrefrange:nnnn	457, 460
\zvLanguageSetup
... 4, 91, 566, 577, 588, 599, 610, 621	
\zvpageref	4, 7, 11, 240, 425
\zvpagerefrange	12, 246, 473
\zvref	11, 410
\zvrefrange	12, 454