

The `zref-vario` package^{*}

Code documentation

Gustavo Barros[†]

2022-07-08

EXPERIMENTAL

Contents

1	Initial setup	2
2	Options	2
3	Reference format	3
4	User interface	10
5	<code>zref-check</code> integration	14
6	Languages	15
	Index	17

^{*}This file describes v0.1.5, released 2022-07-08.

[†]<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} {2022-07-08} {0.1.5}
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 Options

`\zvsetup`

`\zvsetup` Provide `\zvsetup`.

```
\zvsetup{<options>}
```

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

(End definition for `\zvsetup`.)

`pageprop`

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

```

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

```

3 Reference format

Provide some functions to define options names in a standard form.

```

\__zrefvario_ref_opt_default_cs:nn      \__zrefvario_ref_opt_default_cs:nn {<option>} {<signature>}
49 \cs_new:Npn \__zrefvario_ref_opt_default_cs:nn #1#2
50 { __zrefvario_ref_opt_default_ #1 : #2 }

```

(End definition for __zrefvario_ref_opt_default_cs:nn.)

```

\__zrefvario_base_lang:n              \__zrefvario_base_lang:n {<language>}
51 \cs_new:Npn \__zrefvario_base_lang:n #1
52 {
53   \zrefclever_language_if_declared:nTF {#1}
54     { \tl_use:c { \zrefclever_language_varname:n {#1} } }
55     { zv_unknown_language }
56 }
57 \cs_generate_variant:Nn \__zrefvario_base_lang:n { e }

```

(End definition for __zrefvario_base_lang:n.)

\l_zrefvario_ref_language_tl An internal version of \l_zrefclever_ref_language_tl, since we need to retrieve the current value of the variable in multiple places.

```

58 \tl_new:N \l_zrefvario_ref_language_tl
59 \tl_set:Nn \l_zrefvario_ref_language_tl { \l_zrefclever_ref_language_tl }

```

(End definition for \l_zrefvario_ref_language_tl.)

```

\__zrefvario_ref_opt_name_lang_tl:nn  \__zrefvario_ref_opt_name_lang_tl:nn {<lang>} {<option>}
60 \cs_new:Npn \__zrefvario_ref_opt_name_lang_tl:nn #1#2
61 { g__zrefvario_ref_opt_lang_ \__zrefvario_base_lang:n {#1} _ #2 _tl }
62 \cs_generate_variant:Nn \__zrefvario_ref_opt_name_lang_tl:nn { en }

```

(End definition for `_zrefvario_ref_opt_name_lang_tl:nn`.)

```
\_zrefvario_ref_opt_name_lang_cs:nnn    \_zrefvario_ref_opt_name_lang_cs:nnn {<lang>} {<option>} {<signature>}
63 \cs_new:Npn \_zrefvario_ref_opt_name_lang_cs:nnn #1#2#3
64   { \_zrefvario_ref_opt_lang_ \_zrefvario_base_lang:n {#1} _ #2 : #3 }
65 \cs_generate_variant:Nn \_zrefvario_ref_opt_name_lang_cs:nnn { enn }
```

(End definition for `_zrefvario_ref_opt_name_lang_cs:nnn`.)

Retrieving options values.

```
\_zrefvario_ref_opt_tl_get:nnN    \_zrefvario_ref_opt_tl_get:nnN {<lang>} {<option>} {<tl var to set>}
\_zrefvario_ref_opt_cs_get:nnnN    \_zrefvario_ref_opt_cs_get:nnnN {<lang>} {<option>}
\_zrefvario_ref_opt_cs_get:nnnN    {<cs signature>} {<cs to set>}
66 \cs_new_protected:Npn \_zrefvario_ref_opt_tl_get:nnN #1#2#3
67   {
68     \tl_if_exist:cTF
69       { \_zrefvario_ref_opt_name_lang_tl:nn {#1} {#2} }
70       {
71         \tl_set_eq:Nc #3
72           { \_zrefvario_ref_opt_name_lang_tl:nn {#1} {#2} }
73       }
74       {
75         \tl_set_eq:Nc #3
76           { \_zrefvario_ref_opt_name_lang_tl:nn { zv_unknown_language } {#2} }
77       }
78   }
79 \cs_generate_variant:Nn \_zrefvario_ref_opt_tl_get:nnN { xnN }
80 \cs_new_protected:Npn \_zrefvario_ref_opt_cs_get:nnnN #1#2#3#4
81   {
82     \cs_if_exist:cTF
83       { \_zrefvario_ref_opt_name_lang_cs:nnn {#1} {#2} {#3} }
84       {
85         \cs_set_eq:Nc #4
86           { \_zrefvario_ref_opt_name_lang_cs:nnn {#1} {#2} {#3} }
87       }
88       {
89         \cs_if_exist:cTF
90           { \_zrefvario_ref_opt_default_cs:nn {#2} {#3} }
91           {
92             \cs_set_eq:Nc #4
93               { \_zrefvario_ref_opt_default_cs:nn {#2} {#3} }
94           }
95           {
96             \cs_set_eq:Nc #4
97               {
98                 \_zrefvario_ref_opt_name_lang_cs:nnn
99                   { zv_unknown_language } {#2} {#3}
100               }
101           }
102       }
103   }
104 \cs_generate_variant:Nn \_zrefvario_ref_opt_cs_get:nnnN { xnnN }
```

(End definition for `_zrefvario_ref_opt_tl_get:nnN` and `_zrefvario_ref_opt_cs_get:nnnN`.)

Handle the options set of the user commands, and add support for `varioref` optional arguments. All commands can receive the same set of options `\zcref` can, and these are processed the same way and “consumed” by any `\zcref` or `\zcpageref` calls. Besides those, `vother` is made available for `\zvpageref`, corresponding to its second optional argument, and `vcurrent` which is available to all commands except `\fullref`, corresponding to their first optional arguments. The distinction between `\l__zrefvario_reftextcurrent_tl` and `\l__zrefvario_vref_reftextcurrent_tl` is in their default/initial values. The default values for the optional arguments of `\vpageref`, `\vrefrange`, and `\vpagerefrange` are set to `\reftextcurrent`, while the default value of the optional argument of `\vref` is empty.

```

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

```

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

```

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

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

```

```

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

```

(End definition for \zvLanguageSetup.)

```

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

```

```

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

```

```

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

```

Default reference formats for all languages. These will be used by `__zrefvario_ref_opt_cs_get:nnnN`, unless a language specific setting for the same format has been set. `\zreftextfaraway` is defined so it can be called with different arguments in `\zvpageref` (where it is alone) and in `\zfullref` (where it is the second of the pair).

```

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

```



```

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

```

Warnings and values for missing strings / unknown languages.

```

307 \group_begin:
308 \tl_set:Nn \l__zrefvario_setup_language_tl { zv_unknown_language }
309 \AddToHook { begindocument/before }
310 {
311   \keys_set:nn { zref-vario/langsetup }
312   {
313     reftextfaceafter =
314     {
315       \msg_warning:nxxx { zref-vario } { missing-string }
316       { reftextfaceafter } { \l__zrefvario_ref_language_tl }
317       \zref@default
318     } ,
319     reftextfacebefore =
320     {
321       \msg_warning:nxxx { zref-vario } { missing-string }
322       { reftextfacebefore } { \l__zrefvario_ref_language_tl }
323       \zref@default
324     } ,
325     reftextafter =
326     {
327       \msg_warning:nxxx { zref-vario } { missing-string }
328       { reftextafter } { \l__zrefvario_ref_language_tl }
329       \zref@default
330     } ,
331     reftextbefore =
332     {
333       \msg_warning:nxxx { zref-vario } { missing-string }
334       { reftextbefore } { \l__zrefvario_ref_language_tl }
335       \zref@default
336     } ,
337     reftextcurrent =
338     {
339       \msg_warning:nxxx { zref-vario } { missing-string }
340       { reftextcurrent } { \l__zrefvario_ref_language_tl }
341       \zref@default
342     } ,
343     reftextfaraway =
344     {
345       \msg_warning:nxxx { zref-vario } { missing-string }
346       { reftextfaraway } { \l__zrefvario_ref_language_tl }
347       \zref@default

```

```

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

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

366 \AddToHook { begindocument/before }
367 {
368     \AddToHook { begindocument }
369     {
370         \seq_if_in:NxF \g__zrefvario_setup_languages_seq
371         { \__zrefvario_base_lang:e { \l__zrefvario_ref_language_tl } }
372         {
373             \msg_warning:nxx { zref-vario } { language-not-setup }
374             { \l__zrefvario_ref_language_tl }
375         }
376     }
377 }
378 \msg_new:nnn { zref-vario } { language-not-setup }
379 {
380     Language~'#1'~is-not-set-up-for~'zref-vario'.~
381     See~documentation-for~'\iow_char:N\zvLanguageSetup'.
382 }

```

4 User interface

These three small functions are responsible for “connecting” `varioref` with `zref`, instead of with the standard referencing system. They are meant to locally replace their `varioref` counterparts: `\vref@pagenum`, `\vref@pagenum`, and `\vref@label`.

```

383 \cs_new_protected:Npn \__zrefvario_vref_pagenum:Nn #1#2
384 {
385     \exp_args:NNNo \exp_args:NNo \cs_set_nopar:Npn #1
386     { \zref@extractdefault {#2} { \l__zrefvario_pageprop_tl } { ?? } }
387 }
388 \cs_new_protected:Npn \__zrefvario_vref_pagenum:Nn #1#2
389 {
390     \zref@refused {#2}
391     \__zrefvario_vref_pagenum:Nn #1 {#2}
392 }

```

```

393 \cs_new_protected:Npn \__zrefvario_vref_label:n #1
394   { \zref@labelbyprops {#1} { \l__zrefvario_pageprop_tl } }

```

(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.

```

395 \bool_new:N \l__zrefvario_varioref_setup_bool
396 \cs_new_protected:Npn \__zrefvario_varioref_setup:
397   {
398     \bool_if:NF \l__zrefvario_varioref_setup_bool
399     {
400       \bool_set_true:N \l__zrefvario_varioref_setup_bool
401       \@vrefhandlespacefalse
402       \cs_set_eq:NN \vref@pagenum \__zrefvario_vref_pagenum:Nn
403       \cs_set_eq:NN \vrefpagenum \__zrefvario_vrefpagenum:Nn
404       \cs_set_eq:NN \vref@label \__zrefvario_vref_label:n
405       \__zrefvario_ref_opt_tl_get:xnN
406         { \l__zrefvario_ref_language_tl }
407         { reftextfaceafter }
408         \reftextfaceafter
409       \__zrefvario_ref_opt_tl_get:xnN
410         { \l__zrefvario_ref_language_tl }
411         { reftextfacebefore }
412         \reftextfacebefore
413       \__zrefvario_ref_opt_tl_get:xnN
414         { \l__zrefvario_ref_language_tl }
415         { reftextafter }
416         \reftextafter
417       \__zrefvario_ref_opt_tl_get:xnN
418         { \l__zrefvario_ref_language_tl }
419         { reftextbefore }
420         \reftextbefore
421       \__zrefvario_ref_opt_tl_get:xnN
422         { \l__zrefvario_ref_language_tl }
423         { reftextcurrent }
424         \reftextcurrent
425       \__zrefvario_ref_opt_cs_get:xnnN
426         { \l__zrefvario_ref_language_tl }
427         { reftextfaraway } { n }
428         \reftextfaraway
429       \__zrefvario_ref_opt_cs_get:xnnN
430         { \l__zrefvario_ref_language_tl }
431         { reftextpagerange } { nn }
432         \reftextpagerange
433       \__zrefvario_ref_opt_cs_get:xnnN
434         { \l__zrefvario_ref_language_tl }
435         { reftextlabelrange } { nn }
436         \reftextlabelrange
437       \__zrefvario_ref_opt_cs_get:xnnN
438         { \l__zrefvario_ref_language_tl }
439         { vrefformat } { nn }

```

```

440     \vrefformat
441     \__zrefvario_ref_opt_cs_get:xnnN
442     { \l__zrefvario_ref_language_tl }
443     { vrefrangeformat } { nnn }
444     \vrefrangeformat
445     \__zrefvario_ref_opt_cs_get:xnnN
446     { \l__zrefvario_ref_language_tl }
447     { fullrefformat } { n }
448     \fullrefformat
449   }
450 }

```

(End definition for `__zrefvario_varioref_setup:`)

User commands.

```

\zvref      \zvref{*}[\options]{\label}
451 \NewDocumentCommand \zvref { s O { } m }
452 { \zref@wrapper@babel \__zrefvario_zvref:nnn {#3} {#1} {#2} }
453 \cs_new_protected:Npn \__zrefvario_zvref:nnn #1#2#3
454 {
455   \group_begin:
456   \keys_set:nn { zref-vario/ref-oneoptarg } {#3}
457   \bool_if:nT {#2}
458     { \keys_set:nn { zref-vario/ref } { nohyperref } }
459   \__zrefvario_varioref_setup:
460   \__zrefvario_vref:Vn \l__zrefvario_vref_reftextcurrent_tl {#1}
461   \group_end:
462 }

```

“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.

```

463 \cs_new_protected:Npn \__zrefvario_vref:nn #1#2
464 { \vrefformat {#1} {#2} }
465 \cs_generate_variant:Nn \__zrefvario_vref:nn { Vn }

```

(End definition for `\zvref`.)

```

\zvpageref      \zvpageref{*}[\options]{\label}
466 \NewDocumentCommand \zvpageref { s O { } m }
467 { \zref@wrapper@babel \__zrefvario_zvpageref:nnn {#3} {#1} {#2} }
468 \cs_new_protected:Npn \__zrefvario_zvpageref:nnn #1#2#3
469 {
470   \group_begin:
471   \keys_set:nn { zref-vario/ref-twooptargs } {#3}
472   \bool_if:nT {#2}
473     { \keys_set:nn { zref-vario/ref } { nohyperref } }
474   \__zrefvario_varioref_setup:
475   \__zrefvario_vpageref:VVn \l__zrefvario_reftextcurrent_tl
476     \l__zrefvario_reftextother_tl {#1}
477   \bool_lazy_and:nnT
478     { \g__zrefvario_zrefcheck_available_bool }
479     { \l__zrefvario_with_vcheck_bool }

```

```

480     {
481       \zrefcheck_zrefvario_label:
482       \vref@pagenum \l_tmpa_tl { \the\c@vrcnt @vr }
483       \vref@pagenum \l_tmpb_tl {#1}
484       \tl_if_eq:NNT \l_tmpa_tl \l_tmpb_tl
485       {
486         \zrefcheck_zrefvario_run_check_on_label:Vn
487         \l__zrefvario_vcheck_tl {#1}
488       }
489     }
490   \group_end:
491 }
492 \cs_new_protected:Npn \__zrefvario_vpageref:nnn #1#2#3
493 { \vpageref [{#1}] [#2] {#3} }
494 \cs_generate_variant:Nn \__zrefvario_vpageref:nnn { VVn }

```

(End definition for \zvpageref.)

```

\zvrefrange      \zvrefrange(*)[<options>]{<label>}{<label>}
495 \NewDocumentCommand \zvrefrange { s O { } m m }
496 {
497   \zref@wrapper@babel
498   { \zref@wrapper@babel \__zrefvario_zvrefrange:nnnn {#3} }
499   {#4} {#1} {#2}
500 }
501 \cs_new_protected:Npn \__zrefvario_zvrefrange:nnnn #1#2#3#4
502 {
503   \group_begin:
504   \keys_set:nn { zref-vario/ref-oneoptarg } {#4}
505   \bool_if:nT {#3}
506     { \keys_set:nn { zref-vario/ref } { nohyperref } }
507   \__zrefvario_varioref_setup:
508   \__zrefvario_vrefrange:Vnn \l__zrefvario_reftextcurrent_tl {#1} {#2}
509   \group_end:
510 }
511 \cs_new_protected:Npn \__zrefvario_vrefrange:nnn #1#2#3
512 { \vrefrange [{#1}] {#2} {#3} }
513 \cs_generate_variant:Nn \__zrefvario_vrefrange:nnn { Vnn }

```

(End definition for \zvrefrange.)

```

\zvpagerefrange  \zvpagerefrange(*)[<options>]{<label>}{<label>}
514 \NewDocumentCommand \zvpagerefrange { s O { } m m }
515 {
516   \zref@wrapper@babel
517   { \zref@wrapper@babel \__zrefvario_zvpagerefrange:nnnn {#3} }
518   {#4} {#1} {#2}
519 }
520 \cs_new_protected:Npn \__zrefvario_zvpagerefrange:nnnn #1#2#3#4
521 {
522   \group_begin:
523   \keys_set:nn { zref-vario/ref-oneoptarg } {#4}
524   \bool_if:nT {#3}
525     { \keys_set:nn { zref-vario/ref } { nohyperref } }

```

```

526 \__zrefvario_varioref_setup:
527 \__zrefvario_vpagerefrange:Vnn \l__zrefvario_reftextcurrent_tl {#1} {#2}
528 \bool_lazy_and:nnT
529 { \g__zrefvario_zrefcheck_available_bool }
530 { \l__zrefvario_with_vcheck_bool }
531 {
532   \zrefcheck_zrefvario_label:
533   \vrefpagemum \l_tmpa_tl {#1}
534   \vrefpagemum \l_tmpb_tl {#2}
535   \tl_if_eq:NNT \l_tmpa_tl \l_tmpb_tl
536   {
537     \vref@pagemum \l_tmpa_tl { \the\c@vrcnt @vr }
538     \tl_if_eq:NNT \l_tmpa_tl \l_tmpb_tl
539     {
540       \zrefcheck_zrefvario_run_check_on_label:Vn
541       \l__zrefvario_vcheck_tl {#1}
542       \zrefcheck_zrefvario_run_check_on_label:Vn
543       \l__zrefvario_vcheck_tl {#2}
544     }
545   }
546 }
547 \group_end:
548 }
549 \cs_new_protected:Npn \__zrefvario_vpagerefrange:nnn #1#2#3
550 { \vpagerefrange [{#1}] {#2} {#3} }
551 \cs_generate_variant:Nn \__zrefvario_vpagerefrange:nnn { Vnn }

```

(End definition for \zvpagerefrange.)

```

\zfullref \zfullref(*)[(options)]{<label>}
552 \NewDocumentCommand \zfullref { s O { } m }
553 { \zref@wrapper@babel \__zrefvario_zfullref:nnn {#3} {#1} {#2} }
554 \cs_new_protected:Npn \__zrefvario_zfullref:nnn #1#2#3
555 {
556   \group_begin:
557   \keys_set:nn { zref-vario/ref } {#3}
558   \bool_if:nT {#2}
559   { \keys_set:nn { zref-vario/ref } { nohyperref } }
560   \__zrefvario_varioref_setup:
561   \fullref {#1}
562   \group_end:
563 }

```

(End definition for \zfullref.)

5 zref-check integration

Integration with zref-check. `vcheck`, differently from `check`, receives only two values `above` and `below`, since these are the only two checks which make sense for zref-vario. This means that it also does not take `\zcheck`'s options, but these can always be passed to `check`. Besides this, `vcheck`'s check should only be done when label(s) and reference are in the same page, that is, when `vcurrent` is used. Furthermore, since `varioref` handles the case where the reference crosses page boundaries, we only set one label for the purpose,

and this is done at the end of the reference, because that's the point `varioref` itself uses to check if label and reference are in the same page.

```

564 \bool_new:N \g__zrefvario_zrefcheck_available_bool
565 \bool_new:N \l__zrefvario_with_vcheck_bool
566 \tl_new:N \l__zrefvario_vcheck_tl
567 \AddToHook { begindocument/before }
568 {
569   \IfPackageLoadedTF { zref-check }
570   {
571     \IfPackageAtLeastTF { zref-check } { 2022-02-08 }
572     {
573       \bool_gset_true:N \g__zrefvario_zrefcheck_available_bool
574       \keys_define:nn { zref-vario/current }
575       {
576         vcheck .choices:nn =
577           { above , below }
578           {
579             \bool_set_true:N \l__zrefvario_with_vcheck_bool
580             \tl_set:NV \l__zrefvario_vcheck_tl \l_keys_choice_tl
581           } ,
582       }
583     }
584   {
585     \bool_gset_false:N \g__zrefvario_zrefcheck_available_bool
586     \keys_define:nn { zref-vario/current }
587     {
588       vcheck .code:n =
589         { \msg_warning:nn { zref-vario } { zref-check-unavailable } } ,
590     }
591   }
592 }
593 {
594   \bool_gset_false:N \g__zrefvario_zrefcheck_available_bool
595   \keys_define:nn { zref-vario/current }
596   {
597     vcheck .code:n =
598       { \msg_warning:nn { zref-vario } { zref-check-unavailable } } ,
599   }
600 }
601 }
602 \msg_new:nnn { zref-vario } { zref-check-unavailable }
603 {
604   Option~'vcheck'~is~only~available~when~'zref-check'~is~loaded~
605   and~newer~than~'2022-02-08~v0.2.4'.
606 }

```

6 Languages

```

607 \zvLanguageSetup { english }
608 {
609   reftextfaceafter = {on-the~\reftextvario{facing}{next}~page} ,
610   reftextfacebefore = {on-the~\reftextvario{facing}{preceding}~page} ,
611   reftextafter = {on-the~\reftextvario{following}{next}~page} ,

```

```

612     reftextbefore = {on~the~\reftextvario{preceding}{previous}~page} ,
613     reftextcurrent = {on~\reftextvario{this}{the~current}~page} ,
614     reftextfaraway = {on~\zcpageref{#1}} ,
615     reftextpagerange = {on~\zcpageref[range]{#1,#2}} ,
616     reftextlabelrange = {\zcref[range]{#1,#2}} ,
617 }
618 \zvLanguageSetup { german }
619 {
620     reftextfaceafter = {auf~der~nächsten~Seite} ,
621     reftextfacebefore = {auf~der~vorherigen~Seite} ,
622     reftextafter = {auf~der~nächsten~Seite} ,
623     reftextbefore = {auf~der~vorherigen~Seite} ,
624     reftextcurrent = {auf~dieser~Seite} ,
625     reftextfaraway = {auf~\zcpageref[d=D,g=f]{#1}} ,
626     reftextpagerange = {auf~den~\zcpageref[range,d=D,g=f]{#1,#2}} ,
627     reftextlabelrange = {\zcref[range]{#1,#2}} ,
628 }
629 \zvLanguageSetup { french }
630 {
631     reftextfaceafter = {page~\reftextvario{ci-contre}{suivante}} ,
632     reftextfacebefore = {page~\reftextvario{ci-contre}{précédente}} ,
633     reftextafter = {page~suivante} ,
634     reftextbefore = {page~précédente} ,
635     reftextcurrent = {de~la~présente~page} ,
636     reftextfaraway = {\zcpageref[g=f]{#1}} ,
637     reftextpagerange = {\zcpageref[range,g=f]{#1,#2}} ,
638     reftextlabelrange = {\zcref[range]{#1,#2}} ,
639 }
640 \zvLanguageSetup { portuguese }
641 {
642     reftextfaceafter = {na~\reftextvario{página~oposta}{próxima~página}} ,
643     reftextfacebefore = {na~página~\reftextvario{oposta}{anterior}} ,
644     reftextafter = {na~\reftextvario{página~seguinte}{próxima~página}} ,
645     reftextbefore = {na~página~\reftextvario{anterior}{precedente}} ,
646     reftextcurrent = {nesta~página} ,
647     reftextfaraway = {na~\zcpageref[g=f]{#1}} ,
648     reftextpagerange = {nas~\zcpageref[range,g=f]{#1,#2}} ,
649     reftextlabelrange = {\zcref[range]{#1,#2}} ,
650 }
651 \zvLanguageSetup { spanish }
652 {
653     reftextfaceafter = {en~la~página~siguiente} ,
654     reftextfacebefore = {en~la~página~anterior} ,
655     reftextafter = {en~la~página~siguiente} ,
656     reftextbefore = {en~la~página~anterior} ,
657     reftextcurrent = {en~esta~página} ,
658     reftextfaraway = {en~la~\zcpageref[g=f]{#1}} ,
659     reftextpagerange = {en~las~\zcpageref[range,g=f]{#1,#2}} ,
660     reftextlabelrange = {\zcref[range]{#1,#2}} ,
661 }
662 \zvLanguageSetup { dutch }
663 {

```



```

664   retextfaceafter = {op-de-\reftextvario{rechter-pagina}{pagina-hiernaast}} ,
665   retextfacebefore = {op-de-\reftextvario{linker-pagina}{pagina-hiernaast}} ,
666   retextafter = {op-de-\reftextvario{volgende-pagina}{pagina-hierna}} ,
667   retextbefore = {op-de-\reftextvario{vorige-pagina}{pagina-hiervoor}} ,
668   retextcurrent = {op-deze-pagina} ,
669   retextfaraway = {op-\zcpageref[g=f]{#1}} ,
670   retextpagerange = {op-\zcpageref[range,g=f]{#1,#2}} ,
671   retextlabelrange = {\zcref[range]{#1,#2}} ,
672 }
673 \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, 26, 136, 309, 366, 368, 567	group commands:	
		\group_begin:	138, 299, 307, 455, 470, 503, 522, 556
		\group_end:	147, 305, 363, 461, 490, 509, 547, 562
	B		
bool commands:			I
\bool_gset_false:N	585, 594	\IfPackageAtLeastTF	571
\bool_gset_true:N	573	\IfPackageLoadedTF	569
\bool_if:NTF	398	iow commands:	
\bool_if:nTF	301, 457, 472, 505, 524, 558	\iow_char:N	381
\bool_lazy_and:nnTF	477, 528		
\bool_new:N	395, 564, 565		
\bool_set_true:N	400, 579		
	C		K
cs commands:		keys commands:	
\cs_generate_variant:Nn	57, 62, 65, 79, 104, 162, 465, 494, 513, 551	\l_keys_choice_tl	580
\cs_gset_nopar:Npn	212, 223, 234, 245, 256, 267	\keys_define:nn	11, 28, 109, 118, 123, 163, 574, 586, 595
\cs_if_exist:NTF	82, 89	\keys_set:nn	9, 142, 300, 302, 311, 456, 458, 471, 473, 504, 506, 523, 525, 557, 559
\cs_new:Npn	49, 51, 60, 63		
\cs_new_nopar:Npn	277, 283, 289		
\cs_new_protected:Npn	66, 80, 156, 297, 383, 388, 393, 396, 453, 463, 468, 492, 501, 511, 520, 549, 554		
\cs_set_eq:NN	85, 92, 96, 402, 403, 404		
\cs_set_nopar:Npn	385		
	E		M
exp commands:		msg commands:	
\exp_args:NNNo	385	\msg_line_context:	38, 41, 46, 152, 365
\exp_args:NNo	385	\msg_new:nnn	37, 39, 44, 150, 364, 378, 602
		\msg_warning:nn	589, 598
		\msg_warning:nnn	16, 20, 32, 146, 373
		\msg_warning:nnnn	315, 321, 327, 333, 339, 345, 351, 357
	F		
\fullref	5, 561		
\fullrefformat	448		
			N
		\NewDocumentCommand	8, 134, 295, 451, 466, 495, 514, 552
			P
		\ProvidesExplPackage	3

R		V	
<code>\reftextafter</code>	416	<code>\vpageref</code>	5, 493
<code>\reftextbefore</code>	420	<code>\vpagerefrange</code>	5, 550
<code>\reftextcurrent</code>	5, 106, 424	<code>\vref</code>	5, 12
<code>\reftextfaceafter</code>	408	<code>\vrefformat</code>	12, 440, 464
<code>\reftextfacebefore</code>	412	<code>\vrefpagenum</code>	10, 403, 483, 533, 534
<code>\reftextfaraway</code>	304, 428	<code>\vrefrange</code>	5, 512
<code>\reftextlabelrange</code>	286, 436	<code>\vrefrangeformat</code>	444
<code>\reftextpagerange</code>	432	Z	
<code>\reftextvario</code>	609, 610, 611, 612, 613, 631, 632, 642, 643, 644, 645, 664, 665, 666, 667	<code>\zcheck</code>	14
<code>\RequirePackage</code>	5, 7	<code>\zcpageref</code>	5, 614, 615, 625, 626, 636, 637, 647, 648, 658, 659, 669, 670
S		<code>\zcref</code>	5, 280, 292, 616, 627, 638, 649, 660, 671
seq commands:		<code>\zfullref</code>	8, 14, 552
<code>\seq_gput_right:Nn</code>	143	zrefcheck commands:	
<code>\seq_if_in:NnTF</code>	370	<code>\zrefcheck_zrefvario_label:</code>	481, 532
<code>\seq_new:N</code>	133	<code>\zrefcheck_zrefvario_run_check_</code> <code>on_label:nn</code>	486, 540, 542
T		zrefclever commands:	
\TeX and \LaTeX_{ϵ} commands:		<code>\zrefclever_language_if_declared:nTF</code>	53, 139
<code>\@onlypreamble</code>	155	<code>\zrefclever_language_varname:n</code> ..	54
<code>\@vrefhandlespace</code>	12	<code>\l_zrefclever_ref_language_tl</code> ..	3, 59
<code>\@vrefhandlespacefalse</code>	401	<code>\zreftextfaraway</code>	8, 293, 295
<code>\c@vrcnt</code>	482, 537	zrefvario internal commands:	
<code>\vr@f</code>	12	<code>__zrefvario_base_lang:n</code>	3, 51, 51, 57, 61, 64, 144, 371
<code>\vref@label</code>	10, 404	<code>__zrefvario_opt_tl_gset:Nn</code>	156, 162, 167, 176, 185, 194, 203
<code>\vref@pagenum</code>	10, 402, 482, 537	<code>\l__zrefvario_pageprop_tl</code>	10, 19, 386, 394
<code>\vref@star</code>	12	<code>\l__zrefvario_ref_language_tl</code> ..	58, 316, 322, 328, 334, 340, 346, 352, 358, 371, 374, 406, 410, 414, 418, 422, 426, 430, 434, 438, 442, 446
<code>\zref@default</code>	317, 323, 329, 335, 341, 347, 353, 359	<code>__zrefvario_ref_opt_cs_get:nnN</code>	4, 8, 66, 80, 104, 425, 429, 433, 437, 441, 445
<code>\zref@extractdefault</code>	386	<code>__zrefvario_ref_opt_default_</code> <code>cs:nn</code>	3, 49, 49, 90, 93, 278, 284, 290
<code>\zref@iflistcontainsprop</code>	18	<code>__zrefvario_ref_opt_name_lang_</code> <code>cs:nnn</code>	4, 63, 63, 65, 83, 86, 98, 214, 225, 236, 247, 258, 269
<code>\zref@ifpropundefined</code>	15	<code>__zrefvario_ref_opt_name_lang_</code> <code>tl:nn</code>	3, 60, 60, 62, 69, 72, 76, 169, 178, 187, 196, 205
<code>\zref@labelbyprops</code>	394	<code>__zrefvario_ref_opt_tl_get:nnN</code> ..	4, 66, 66, 79, 405, 409, 413, 417, 421
<code>\ZREF@mainlist</code>	18	<code>\l__zrefvario_reftextcurrent_tl</code> ..	5, 105, 106, 113, 475, 508, 527
<code>\zref@refused</code>	390	<code>\l__zrefvario_reftextother_tl</code> ..	108, 120, 476
<code>\zref@wrapper@babel</code>	296, 452, 467, 497, 498, 516, 517, 553		
<code>\the</code>	482, 537		
tl commands:			
<code>\tl_gset:Nn</code>	160		
<code>\tl_if_eq:NNTF</code>	484, 535, 538		
<code>\tl_if_exist:NTF</code>	68, 158		
<code>\tl_new:N</code>	10, 58, 105, 107, 108, 132, 159, 566		
<code>\tl_set:Nn</code>	19, 59, 106, 113, 114, 141, 308, 580		
<code>\tl_set_eq:NN</code>	71, 75		
<code>\tl_use:N</code>	54		
<code>\l_tmpa_tl</code>	482, 484, 533, 535, 537, 538		
<code>\l_tmpb_tl</code>	483, 484, 534, 535, 538		

<code>\l_zrefvario_setup_language_tl</code>	<code>_zrefvario_vrefpagenum:Nn</code>
. 132 , 141 , 170 , 179 , 188 , 197 , 383 , 388 , 403
206 , 215 , 226 , 237 , 248 , 259 , 270 , 308	<code>_zrefvario_vrefrange:nnn</code>
<code>\g_zrefvario_setup_languages_-seq</code> 508 , 511 , 513
. 133 , 143 , 370	<code>\l_zrefvario_with_vcheck_bool</code>
<code>_zrefvario_varioref_setup:</code> 479 , 530 , 565 , 579
. 303 , 395 , 396 , 459 , 474 , 507 , 526 , 560	<code>_zrefvario_zfullref:nmn</code> 553 , 554
<code>\l_zrefvario_varioref_setup_-bool</code>	<code>\g_zrefvario_zrefcheck_available_-bool</code>
. 11 , 395 , 398 , 400 478 , 529 , 564 , 573 , 585 , 594
<code>\l_zrefvario_vcheck_tl</code>	<code>_zrefvario_zreftextfaraway:nmn</code>
. 487 , 541 , 543 , 566 , 580 296 , 297
<code>_zrefvario_vpageref:nmn</code>	<code>_zrefvario_zvpageref:nmn</code> 467 , 468
. 475 , 492 , 494	<code>_zrefvario_zvpagerefrange:nmnn</code>
<code>_zrefvario_vpagerefrange:nmn</code> 517 , 520
. 527 , 549 , 551	<code>_zrefvario_zvref:nmn</code> 452 , 453
<code>_zrefvario_vref:nn</code> 460 , 463 , 465	<code>_zrefvario_zvrefrange:nmnn</code> 498 , 501
<code>_zrefvario_vref_label:n</code>	<code>\zvLanguageSetup</code>
. 383 , 393 , 404 5 , 132 , 607 , 618 , 629 , 640 , 651 , 662
<code>_zrefvario_vref_pagenum:Nn</code>	<code>\zvpageref</code> 5 , 8 , 12 , 281 , 466
. 383 , 383 , 391 , 402	<code>\zvpagerefrange</code> 13 , 287 , 514
<code>\l_zrefvario_vref_reftextcurrent_-tl</code>	<code>\zvref</code> 12 , 451
. 5 , 107 , 114 , 460	<code>\zvrefrange</code> 13 , 495
	<code>\zvsetup</code> 2 , 8