

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

Code documentation

Gustavo Barros[†]

2022-12-29

EXPERIMENTAL

Contents

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

^{*}This file describes v0.1.6, released 2022-12-29.

[†]<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} {2022-12-29} {0.1.6}
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.

Note that `zref-clever v0.2.0-alpha`, from 2022-01-28, requires a kernel 2021-11-15 or newer.

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

Also note that, contrary to `zref-clever`, `zref-vario` is incompatible with `cleveref`, because the latter patches `varioref` in ways that break `zref-vario`.

2 Options

`\zvsetup`

`\zvsetup` Provide `\zvsetup`.

```
\zvsetup{<options>}

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

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

```

Warnings and values for missing strings / unknown languages.

```

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

```

```

344         { reftextcurrent } { \l__zrefvario_ref_language_tl }
345         \zref@default
346     } ,
347     reftextfaraway =
348     {
349         \msg_warning:nxxx { zref-vario } { missing-string }
350         { reftextfaraway } { \l__zrefvario_ref_language_tl }
351         \zref@default
352     } ,
353     reftextpagerange =
354     {
355         \msg_warning:nxxx { zref-vario } { missing-string }
356         { reftextpagerange } { \l__zrefvario_ref_language_tl }
357         \zref@default
358     } ,
359     reftextlabelrange =
360     {
361         \msg_warning:nxxx { zref-vario } { missing-string }
362         { reftextlabelrange } { \l__zrefvario_ref_language_tl }
363         \zref@default
364     } ,
365 }
366 }
367 \group_end:
368 \msg_new:nnn { zref-vario } { missing-string }
369 { Missing~'#1'~string~for~language~'#2'~\msg_line_context:. }

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

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

```

Provide proper hyperlinking for the “nearby” page references. `hyperref` makes `\vref` a *single* link, including both the standard and the page reference (see Ulrike Fischer’s answer and discussion in the comments at <https://tex.stackexchange.com/a/655106>). However, it does not provide the same treatment to `\vpageref`, where there’s no linking at all for nearby pages, to `\vrefrange`, `\vpagerefrange`, or `\vnameref`. In my view, this makes `hyperref` inconsistent. `cleveref` indeed discusses the issue in the “Known Bugs and Work-Arounds” section, mentioning two problems. The first, that the page reference does not honor the `nameinlink` option, which it is actually the same behavior we get

from `\vnameref`. The second, that nearby page references are not included in the links or hyperlinked when by themselves. This latter problem may be considered undesirable, and even a “bug”, but at least `cleveref`’s behavior in this regard is consistent.

This is actually tricky and I think the best alternative here is to not even try to make a single link of both references, as `hyperref` does for `\vref`. But rather to offer ways to hyperlink the “nearby” page references separately, which is something that can work for paired references and for standalone page references just as well. `\zvhyperlink` is provided for such purpose, and is meant to be used either in the reference format settings of `\zvLanguageSetup`, or in the `vcurrent` and `vother` options of the reference commands. It only makes a hyperlink if used in one such place (technically, if `\l__zrefvario_linklabel_tl` is not empty), if `\l__zrefvario_hyperlink_bool` is true (to control for starred variants) and if `hyperref` and `zref-hyperref` are loaded, otherwise it passes on its argument untouched.

```

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

\zvhyperlink      \zvhyperlink{<text>}

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

(End definition for \zvhyperlink.)

```

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`, `\vrefpagemum`, and `\vref@label`.

```

415 \cs_new_protected:Npn \__zrefvario_vref_pagenum:Nn #1#2
416 {
417   \exp_args:NNNo \exp_args:NNo \cs_set_nopar:Npn #1
418   { \zref@extractdefault {#2} { \l__zrefvario_pageprop_tl } { ?? } }
419 }
420 \cs_new_protected:Npn \__zrefvario_vrefpagenum:Nn #1#2
421 {
422   \zref@refused {#2}
423   \__zrefvario_vref_pagenum:Nn #1 {#2}
424 }
425 \cs_new_protected:Npn \__zrefvario_vref_label:n #1
426 { \zref@labelbyprops {#1} { \l__zrefvario_pageprop_tl } }

```

(End definition for __zrefvario_vref_pagenum:Nn, __zrefvario_vrefpagenum:Nn, and __zrefvario_vref_label:n.)

__zrefvario_varioref_setup: Sets things up for varioref to work with zref/zref-clever. It is pretty straightforward, but it is the core of the package. \l__zrefvario_varioref_setup_bool ensures this setup runs only once in case of nested calls to the package's commands.

```

427 \bool_new:N \l__zrefvario_varioref_setup_bool
428 \cs_new_protected:Npn \__zrefvario_varioref_setup:
429 {
430   \bool_if:NF \l__zrefvario_varioref_setup_bool
431   {
432     \bool_set_true:N \l__zrefvario_varioref_setup_bool
433     \@vrefhandlespacefalse
434     \cs_set_eq:NN \vref@pagenum \__zrefvario_vref_pagenum:Nn
435     \cs_set_eq:NN \vrefpagenum \__zrefvario_vrefpagenum:Nn
436     \cs_set_eq:NN \vref@label \__zrefvario_vref_label:n
437     \__zrefvario_ref_opt_tl_get:xnN
438     { \l__zrefvario_ref_language_tl }
439     { retextfaceafter }
440     \retextfaceafter
441     \__zrefvario_ref_opt_tl_get:xnN
442     { \l__zrefvario_ref_language_tl }
443     { retextfacebefore }
444     \retextfacebefore
445     \__zrefvario_ref_opt_tl_get:xnN
446     { \l__zrefvario_ref_language_tl }
447     { retextafter }
448     \retextafter
449     \__zrefvario_ref_opt_tl_get:xnN
450     { \l__zrefvario_ref_language_tl }
451     { retextbefore }
452     \retextbefore
453     \__zrefvario_ref_opt_tl_get:xnN
454     { \l__zrefvario_ref_language_tl }
455     { retextcurrent }
456     \retextcurrent
457     \__zrefvario_ref_opt_cs_get:xnnN
458     { \l__zrefvario_ref_language_tl }
459     { retextfaraway } { n }
460     \retextfaraway
461     \__zrefvario_ref_opt_cs_get:xnnN

```

```

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

```

(End definition for __zrefvario_varioref_setup:.)

User commands.

```

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

```

“Ideally”, we’d be using \vref here, but `hyperref` redefines \vref@star and \vr@f, hard-coding what was supposed to be configurable..., so we use \vrefformat instead, which is the original definition of \vref@star, and \vr@f also boils down to \vref@star when \@vrefhandlespace is false, which is our case.

```

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

```

(End definition for \zvref.)

```

\zvpageref          \zvpageref{<*>[<options>]}{<label>}}
502 \NewDocumentCommand \zvpageref { s O { } m }
503 { \zref@wrapper@babel \__zrefvario_zvpageref:nnn {#3} {#1} {#2} }
504 \cs_new_protected:Npn \__zrefvario_zvpageref:nnn #1#2#3
505 {
506   \group_begin:
507   \keys_set:nn { zref-vario/ref-twooptargs } {#3}
508   \bool_if:nT {#2}
509   {
510     \keys_set:nn { zref-vario/ref } { nohyperref }
511     \bool_set_false:N \l__zrefvario_hyperlink_bool
512   }
513   \tl_set:Nn \l__zrefvario_linklabel_tl {#1}
514   \__zrefvario_varioref_setup:
515   \__zrefvario_vpageref:Vn \l__zrefvario_reftextcurrent_tl
516   \l__zrefvario_reftextother_tl {#1}
517   \bool_lazy_and:nnT
518   { \g__zrefvario_zrefcheck_available_bool }
519   { \l__zrefvario_with_vcheck_bool }
520   {
521     \zrefcheck_zrefvario_label:
522     \vref@pagenum \l_tmpa_tl { \the\c@vrcnt @vr }
523     \vref@pagenum \l_tmpb_tl {#1}
524     \tl_if_eq:NNT \l_tmpa_tl \l_tmpb_tl
525     {
526       \zrefcheck_zrefvario_run_check_on_label:Vn
527       \l__zrefvario_vcheck_tl {#1}
528     }
529   }
530   \group_end:
531 }
532 \cs_new_protected:Npn \__zrefvario_vpageref:nnn #1#2#3
533 { \vpageref [{#1}] [{#2}] {#3} }
534 \cs_generate_variant:Nn \__zrefvario_vpageref:nnn { VVn }
(End definition for \zvpageref.)

```

```

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

```

```

551 \vpagerefcompare {#1} {#2}
552 { \tl_set:Nn \l__zrefvario_linklabel_tl {#1} } { }
553 \__zrefvario_vrefrange:Vnn \l__zrefvario_reftextcurrent_tl {#1} {#2}
554 \group_end:
555 }
556 \cs_new_protected:Npn \__zrefvario_vrefrange:nnn #1#2#3
557 { \vrefrange [{#1}] {#2} {#3} }
558 \cs_generate_variant:Nn \__zrefvario_vrefrange:nnn { Vnn }

```

(End definition for \zvrefrange.)

```

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

```

```

601 \cs_generate_variant:Nn \__zrefvario_vpagerefrange:nnn { Vnn }
(End definition for \zvpagerefrange.)

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

```

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

```



```

636     }
637   }
638   {
639     \bool_gset_false:N \g__zrefvario_zrefcheck_available_bool
640     \keys_define:nn { zref-vario/current }
641     {
642       vcheck .code:n =
643         { \msg_warning:nn { zref-vario } { zref-check-unavailable } } ,
644     }
645   }
646 }
647 {
648   \bool_gset_false:N \g__zrefvario_zrefcheck_available_bool
649   \keys_define:nn { zref-vario/current }
650   {
651     vcheck .code:n =
652       { \msg_warning:nn { zref-vario } { zref-check-unavailable } } ,
653   }
654 }
655 }
656 \msg_new:nnn { zref-vario } { zref-check-unavailable }
657 {
658   Option~'vcheck'~is~only~available~when~'zref-check'~is~loaded~
659   and~newer~than~'2022-02-08~v0.2.4'.
660 }

```

6 Languages

Localization for the languages provided by the author, except where otherwise stated. However, I could only do so thanks to `varioref`, in particular for languages I don't really know. So, credits go to the contributors of `varioref`. Of course, if any native speaker would like to review these, or contribute new ones, it is much welcome. Note, however, that since `zref-vario` depends on `zref-clever` and piggybacks on its language infrastructure, it only makes sense to provide localization here for languages already supported by `zref-clever`.

```

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

```

```

678 {
679     retextfaceafter = {auf~der~\zvhyperlink{nächsten~Seite}} ,
680     retextfacebefore = {auf~der~\zvhyperlink{vorherigen~Seite}} ,
681     retextafter = {auf~der~\zvhyperlink{nächsten~Seite}} ,
682     retextbefore = {auf~der~\zvhyperlink{vorherigen~Seite}} ,
683     retextcurrent = {auf~\zvhyperlink{dieser~Seite}} ,
684     retextfaraway = {auf~\zcpageref[d=D,g=f]{#1}} ,
685     retextpagerange = {auf~den~\zcpageref[range,d=D,g=f]{#1,#2}} ,
686     retextlabelrange = {\zcref[range]{#1,#2}} ,
687 }
688 \zvLanguageSetup { french }
689 {
690     retextfaceafter =
691         {\zvhyperlink{page~\retextvario{ci-contre}{suivante}}} ,
692     retextfacebefore =
693         {\zvhyperlink{page~\retextvario{ci-contre}{précédente}}} ,
694     retextafter = {\zvhyperlink{page~suivante}} ,
695     retextbefore = {\zvhyperlink{page~précédente}} ,
696     retextcurrent = {de~la~\zvhyperlink{présente~page}} ,
697     retextfaraway = {\zcpageref[g=f]{#1}} ,
698     retextpagerange = {\zcpageref[range,g=f]{#1,#2}} ,
699     retextlabelrange = {\zcref[range]{#1,#2}} ,
700 }
701 \zvLanguageSetup { portuguese }
702 {
703     retextfaceafter =
704         {na~\zvhyperlink{\retextvario{página-oposta}{próxima-página}}} ,
705     retextfacebefore =
706         {na~\zvhyperlink{página~\retextvario{oposta}{anterior}}} ,
707     retextafter =
708         {na~\zvhyperlink{\retextvario{página-seguinte}{próxima-página}}} ,
709     retextbefore =
710         {na~\zvhyperlink{página~\retextvario{anterior}{precedente}}} ,
711     retextcurrent = {\zvhyperlink{nesta~página}} ,
712     retextfaraway = {na~\zcpageref[g=f]{#1}} ,
713     retextpagerange = {nas~\zcpageref[range,g=f]{#1,#2}} ,
714     retextlabelrange = {\zcref[range]{#1,#2}} ,
715 }
716 \zvLanguageSetup { spanish }
717 {
718     retextfaceafter = {en~la~\zvhyperlink{página-siguiente}} ,
719     retextfacebefore = {en~la~\zvhyperlink{página-anterior}} ,
720     retextafter = {en~la~\zvhyperlink{página-siguiente}} ,
721     retextbefore = {en~la~\zvhyperlink{página-anterior}} ,
722     retextcurrent = {en~\zvhyperlink{esta~página}} ,
723     retextfaraway = {en~la~\zcpageref[g=f]{#1}} ,
724     retextpagerange = {en~las~\zcpageref[range,g=f]{#1,#2}} ,
725     retextlabelrange = {\zcref[range]{#1,#2}} ,
726 }
727 \zvLanguageSetup { dutch }
728 {
729     retextfaceafter =
730         {op~de~\zvhyperlink{\retextvario{rechter~pagina}{pagina-hiernaast}}} ,

```

```

731 retextfacebefore =
732     {op~de~\zvhyperlink{\retextvario{linker~pagina}{pagina~hiernaast}}} ,
733 retextafter =
734     {op~de~\zvhyperlink{\retextvario{volgende~pagina}{pagina~hierna}}} ,
735 retextbefore =
736     {op~de~\zvhyperlink{\retextvario{vorige~pagina}{pagina~hiervoor}}} ,
737 retextcurrent = {op~\zvhyperlink{deze~pagina}} ,
738 retextfaraway = {op~\zcpageref[g=f]{#1}} ,
739 retextpagerange = {op~\zcpageref[range,g=f]{#1,#2}} ,
740 retextlabelrange = {\zceref[range]{#1,#2}} ,
741 }
742 \zvLanguageSetup { italian }
743 {
744     retextfaceafter =
745         {\retextvario{}}{nella~}\zvhyperlink{\retextvario{a~fronte}{pagina~successiva}}} ,
746     retextfacebefore =
747         {\retextvario{}}{nella~}\zvhyperlink{\retextvario{a~fronte}{pagina~precedente}}} ,
748     retextafter =
749         {nella~\zvhyperlink{pagina~\retextvario{seguente}{successiva}}} ,
750     retextbefore = {nella~\zvhyperlink{pagina~precedente}} ,
751     retextcurrent = {in~\zvhyperlink{questa~pagina}} ,
752     retextfaraway = {a~\zcpageref[g=f]{#1}} ,
753     retextpagerange = {nelle~\zcpageref[range,g=f]{#1,#2}} ,
754     retextlabelrange = {\zceref[range]{#1,#2}} ,
755 }
756 \package

```

Index

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

A		62, 65, 79, 104, 162, 501, 534, 558, 601
\AddToHook	6, 26, 136, 313, 370, 372, 389, 391, 621	\cs_gset_nopar:Npn 212, 223, 234, 245, 256, 267
B		\cs_if_exist:NTF 82, 89
bool commands:		\cs_new:Npn 49, 51, 60, 63
\bool_gset_false:N	639, 648	\cs_new_nopar:Npn 277, 283, 289
\bool_gset_true:N	393, 627	\cs_new_protected:Npn 66, 80, 156, 297, 400, 415, 420, 425, 428, 485, 499, 504, 532, 541, 556, 565, 599, 604
\bool_if:NTF	430	\cs_set_eq:NN 85, 92, 96, 434, 435, 436
\bool_if:nTF	301, 489, 508, 545, 569, 608	\cs_set_nopar:Npn 417
\bool_lazy_all:nTF	402	
\bool_lazy_and:nnTF	517, 578	E
\bool_new:N	387, 388, 427, 618, 619	exp commands:
\bool_set_false:N	304, 492, 511, 548, 572, 611	\exp_args:NNNo 417
\bool_set_true:N	394, 432, 633	\exp_args:NNo 417
C		
cs commands:		F
\cs_generate_variant:Nn	57,	\fullref 5, 615
		\fullrefformat 480

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

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