

The zref-vario package^{*}

User manual

Gustavo Barros[†]

2022-01-31

Abstract

zref-vario offers a compatibility layer for varioref to be used alongside zref-clever. It provides “\z...” counterparts to varioref’s main reference commands, each of which essentially does some (scoped) setup for varioref, then calls the original one.

EXPERIMENTAL

Contents

1	Loading the package	2
2	Dependencies	2
3	User interface	2
4	Customization	3
5	Change history	3

^{*}This file describes v0.1.0-alpha, released 2022-01-31.

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

1 Loading the package

As usual:

```
\usepackage{zref-vario}
```

2 Dependencies

zref-vario requires varioref and zref-clever.

3 User interface

<hr/> <hr/>	<code>\zvref</code>	<code>\zvref{*}[\langle options \rangle]{\langle label \rangle}</code>
<hr/> <hr/>	<code>\zvpageref</code>	<code>\zvpageref{*}[\langle options \rangle]{\langle label \rangle}</code>
<hr/> <hr/>	<code>\zvrefrange</code>	<code>\zvrefrange{*}[\langle options \rangle]{\langle label \rangle}{\langle label \rangle}</code>
<hr/> <hr/>	<code>\zvpagerefrange</code>	<code>\zvpagerefrange{*}[\langle options \rangle]{\langle label \rangle}{\langle label \rangle}</code>
<hr/> <hr/>	<code>\zfullref</code>	<code>\zfullref{*}[\langle options \rangle]{\langle label \rangle}</code>

These user commands work pretty much as their `varioref` counterparts.¹ Indeed, they are just wrappers around them. As such, differently from `zref-clever`'s commands, these can receive single labels as arguments, not lists of them. In all of them the starred version inhibits hyperlinking, and they (locally) set `varioref`'s `nospace` option, so that the this syntax can be used unambiguously.

All of them have a single optional argument, which can receive any of `\zceref`'s options, and those are set for calls to `\zceref` or `\zpageref` which are part of the building blocks of the reference formats and strings set for `varioref`. However, there is indeed some potential friction and caveats stemming from the use of these options, which were designed to work for single calls of `\zceref`, to apply to *pairs* of them. Hence, whether each and every of these options is meaningful, useful, or potentially harmful in this context depends on the case, and it is up to the user to make proper use of them. Of course, one can always split the pair using `\zceref` and then `\zvpageref`, `\zvpagerefrange`, or `\zcpageref`, and have more control over each part. Anyway, the package does what it says: passes the options along to the underlying call(s) to `\zceref`.²

¹If you are not acquainted with them, see `varioref`'s documentation.

²Note that the options given to each call of these user macros are set *before* the call to the original `varioref` counterpart. This means that options given to `\zceref` or `\zpageref` in the strings and formats in `\zvLanguageSetup` will have precedence over the former. Which is useful, so that we can distinguish to some degree the first from the second call of `\zceref`/`\zpageref` done in the formats, but does not eliminate the limitations which arise from the underlying problem.

Besides these options, two other are provided corresponding to `varioref`'s commands' optional arguments: `vcurrent` (the first optional argument) and `vother` (the second optional argument). Of course, these are only available when the underlying `varioref` command supports them.

4 Customization

`\zvLanguageSetup` `\zvLanguageSetup{⟨language⟩}{⟨options⟩}`

User interface for customization of “strings” and “formats” for $\langle language \rangle$. $\langle language \rangle$ must be known to `zref-clever`. The $\langle options \rangle$ are familiar to users of `varioref`: `retextfaceafter`, `retextfacebefore`, `retextafter`, `retextbefore`, `retextcurrent`, `retextfaraway`, `retextpagerange`, `retextlabelrange`, `vrefformat`, `vrefrangeformat`, and `fullrefformat`.³ Their meaning is the same as the ones they have in `varioref` and, indeed, they work by setting those `varioref` macros to the values given to the corresponding options in `\zvLanguageSetup`. If you are setting up a language which has no built-in support, you should set at least the whole set of “`retext...`” options. Language independent default values are provided for the “`...format`” options (equivalent to the ones from `varioref`, which are adequate to most use cases), so you may omit them. But, if you need to adjust them, the default values can be overridden by setting the corresponding options in `\zvLanguageSetup`.

5 Change history

A change log with relevant changes for each version, eventual upgrade instructions, and upcoming changes, is maintained in the package's repository, at <https://github.com/gusbrs/zref-vario/blob/main/CHANGELOG.md>. An archive of historical versions of the package is also kept at <https://github.com/gusbrs/zref-vario/releases>.

³For details, see `varioref`'s documentation.