The zref-vario package*
User manual

Gustavo Barros†

2022-02-11

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

*This file describes v0.1.3-alpha, released 2022-02-11.

†https://github.com/gusbrs/zref-vario

1 Loading the package

As usual:

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

zref-vario will load varioref and zref-clever at `begindocument/before`, if they have not already been loaded. However, considering varioref has known load order interaction with other packages, prominently with hyperref, you may wish to load it explicitly to ensure its load order requirements are met.

2 Dependencies

zref-vario requires varioref and zref-clever.

3 User interface

`\zvref` `\zvref⟨*⟩[⟨options⟩]{⟨label⟩}`

`\zvpageref` `\zvpageref⟨*⟩[⟨options⟩]{⟨label⟩}`

`\zvrefrange` `\zvrefrange⟨*⟩[⟨options⟩]{⟨label⟩}{⟨label⟩}`

`\zvpagerefrange` `\zvpagerefrange⟨*⟩[⟨options⟩]{⟨label⟩}{⟨label⟩}`

`\zfullref` `\zfullref⟨*⟩[⟨options⟩]{⟨label⟩}`

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 `\zcref`'s options, and those are set for calls to `\zcref` 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 `\zcref`, 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 `\zcref` and then `\zvpageref`,

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

\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 \zcref.[2]

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 ⟨*language*⟩. ⟨*language*⟩ must be known to zref-clever. The ⟨*options*⟩ are familiar to users of varioref: reftextfaceafter, reftextfacebefore, reftextafter, reftextbefore, reftextcurrent, reftextfaraway, reftextpagerange, reftextlabelrange, 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 "reftext..." options. Language independent default values are provided for the "...format" options (equivalent to the ones from varioref, which are adequate for 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.

As far as zref-vario is concerned, varioref's options are (mostly) not taken into account: the language options are disregarded (settings made with \zvLanguageSetup are used instead), the nospace option is hard-coded (locally) for the "\z..." commands, and draft and final are typically given to \documentclass, though they do affect zref-vario's commands, just as they do with varioref's.

5 Integration with zref-check

When package zref-check is loaded, zref-vario provides one further option to its user commands: vcheck. The purpose of this option is to check the relative position of label and reference within the same page. It can receive two values: above and below, being those the names of the zref-check's checks which are meaningful for the use case at hand. Of course, these could also be performed with \zcref's check option, which is available for zref-vario's commands as well. The difference here is that the check specified in vcheck is only performed when the referenced label – or labels, in the case of a range

[2] Note 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 \zcref 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 \zcref/\zpageref done in the formats, but does not eliminate the limitations which arise from the underlying problem.

[3] For details, see varioref's documentation.

– fall on the same page as the reference itself.[4] In other words, when `vcurrent` would be used (if provided). For this reason, only the commands which support `vcurrent` offer vcheck. Also, `vcheck` cannot receive zref-check's options, as `check` is able to, but the latter can be used for the purpose of locally controlling the behavior of the checks performed by `vcheck`. Consult zref-check's documentation for details and limitations of these checks and envisaged workflows for their reliable use.

6 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`.

[4]There's another technical difference between them. zref-check's checks, and hence those of the check option, make sure the whole reference passes each check by setting labels both at the start and the end of the reference, and verifying if each one of them passes the checks. But, since varioref already has it's own mechanism to handle references which cross page boundaries, vcheck's checks set only one label, at the end of the reference, the same position varioref uses to check whether label and reference are on the same page.