

# hvindex, v. 0.03 simplifying indexing ...

Herbert Voß

2009/10/24

## Inhaltsverzeichnis

1	Package options	1
2	Introduction	1
3	The macros	2
	Index	4

## 1 Package options

With the option `makeidx` the also called package is loaded and the macro `\makeindex` is executed.

The package should be loaded *after* package `babel`.

## 2 Introduction

The package `hvindex` defines some macros as a replacement for `\index` which makes `live` easier. You have to type every word only once to get it into the index and into the text .

The `\Index{package} \ttIndex{hvindex}` defines some `\Index{macros}` as a `\Index{macro!replacement}` for `\Index{macros!index@\texttt{\textbackslash index}}` which makes `\Index{live}` easier `\index{easy}`. You have to type every `\Index{word}` only `\Index{once}` to get it into the `\Index{index}` and into the `\Index{text}`.

### 3 The macros

<i>code</i>	<i>Text</i>	<i>index</i>
<code>\Index{foo}</code>	⇒ foo	foo
<code>\Index{foo!bar}</code>	⇒ bar	foo bar
<code>\Index{foo!bar!baz}</code>	⇒ baz	foo bar baz
<code>\Index{foo@\textsc{foo}}</code>	⇒ FOO	FOO
<code>\Index{foo!bar@\textsf{baz}}</code>	⇒ bar	foo bar
<code>\Index{foo!bar!baz@\texttt{baz}}</code>	⇒ baz	foo bar baz
<code>\Index{foo@\textsc{foo}!bar}</code>	⇒ bar	FOO bar
<code>\Index{foo!bar@\textsc{bar}!baz}</code>	⇒ baz	foo BAR baz
<code>\Index{foo@\textsc{foo}}</code>	⇒ FOO	FOO
<code>\Index{foo!bar@\textsf{baz}}</code>	⇒ bar	foo bar
<code>\Index{foo!bar!baz@\texttt{baz}}</code>	⇒ baz	foo bar baz
<code>\ttIndex{foo}</code>	⇒ foo	foo
<code>\bfIndex{foo}</code>	⇒ <b>foo</b>	<b>foo</b>
<code>\sfIndex{foo}</code>	⇒ foo	foo
<code>\scIndex{foo}</code>	⇒ FOO	FOO
<code>\itIndex{foo}</code>	⇒ <i>foo</i>	<i>foo</i>
<code>\ttIndex{foo!bar}</code>	⇒ bar	foo bar
<code>\bfIndex{foo!bar}</code>	⇒ <b>bar</b>	foo <b>bar</b>
<code>\sfIndex{foo!bar}</code>	⇒ bar	foo bar
<code>\scIndex{foo!bar}</code>	⇒ BAR	foo BAR
<code>\itIndex{foo!bar}</code>	⇒ <i>bar</i>	foo <i>bar</i>

continued on next page ...

... continued from previous page

<i>code</i>	<i>Text</i>	<i>index</i>
<code>\ttIndex{foo!bar!baz}</code>	$\Rightarrow$ baz	foo bar baz
<code>\bfIndex{foo!bar!baz}</code>	$\Rightarrow$ <b>baz</b>	foo bar <b>baz</b>
<code>\sfIndex{foo!bar!baz}</code>	$\Rightarrow$ baz	foo bar baz
<code>\scIndex{foo!bar!baz}</code>	$\Rightarrow$ BAZ	foo bar BAZ
<code>\itIndex{foo!bar!baz}</code>	$\Rightarrow$ <i>baz</i>	foo bar <i>baz</i>
<code>\sIndex{foo bar}</code>	$\Rightarrow$ foo	foo, see bar
<code>\saIndex{baz bar}</code>	$\Rightarrow$ baz	baz, see also bar

It is not easy to get the braces into an index, because `\index` reads its argument verbatim. The package defines the following macros:

`\iBraceL`, which writes the { into the index, same for `\iBraceR` for the right }. With the upper case variant it is written into the index and also printed into the text, like this one: { and } (`\IBraceL` and `\IBraceR`). The advantage of these macros is, that the braces are sorted correct.

## Index

{, 3  
|, 3  
}, 3

babel, 1  
bar, 2, 3  
baz, 2, *siehe auch* bar  
braces, 3

easy, 1

foo, 2, *siehe* bar  
  bar  
    **baz**, 3  
    *baz*, 3  
    BAZ, 3  
    baz, 3  
    baz, 2, 3  
  **bar**, 2  
  *bar*, 2  
  BAR, 2  
  bar, 2  
  bar, 2

**foo**, 2  
*foo*, 2  
FOO, 2  
foo, 2  
foo, 2

hvindex, 1

index, 1, 3

live, 1

macros, 1  
  \index, 1  
makeidx, 1

once, 1  
option, 1

package, 1

replacement, 1

text, 1

verbatim, 3

word, 1