

hvindex, v. 0.04a  
simplifying indexing ...

Herbert Voß

2025/01/24

**Contents**

<b>1</b>	<b>Package options</b>	<b>2</b>
<b>2</b>	<b>Introduction</b>	<b>2</b>
<b>3</b>	<b>The macros</b>	<b>3</b>
	<b>Index</b>	<b>5</b>

Thanks to:  
Denis Bitouzé

## 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 `hindex` 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{hindex}` 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>	baz	foo bar baz
<code>\bfIndex{foo!bar!baz}</code>	<b>baz</b>	foo bar <b>baz</b>
<code>\sfIndex{foo!bar!baz}</code>	baz	foo bar baz
<code>\scIndex{foo!bar!baz}</code>	BAZ	foo bar BAZ
<code>\itIndex{foo!bar!baz}</code>	<i>baz</i>	foo bar <i>baz</i>
<code>\sIndex{foo bar}</code>	foo	foo, see bar
<code>\saIndex{baz bar}</code>	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, the character `|` (written as `\lVert`) must appear between them and it does ...

# Index

## Symbols

{, 3  
|, 3  
}, 3

## B

babel, 1  
bar, 2f  
baz, 2, *see also* bar  
braces, 3

## E

easy, 1

## F

foo  
-bar  
- **baz**, 3  
- *baz*, 3  
- BAZ, 3  
- baz, 3  
- baz, 2f  
- **bar**, 2  
- *bar*, 2  
- BAR, 2  
- bar, 2  
- bar, 2

**foo**, 2  
*foo*, 2  
FOO, 2  
foo, 2  
foo, 2  
foo, 2, *see* bar

## H

hvindex, 1

## I

index, 1, 3

## L

live, 1

## M

macros

- \index, 1

macros, 1  
makeidx, 1

## O

once, 1  
option, 1

## P

package, 1

## R

replacement, 1

## T

text, 1

## V

verbatim, 3

## W

word, 1