The **keyindex** package[*]

Richard Zach

1 Introduction

The `keyindex` package provides functionality for producing an index without directly entering index entries into the text using the `\index` command, but instead by looking up short keys and printing a predefined string in the main text and adding a corresponding index entry. The standard use case is the production of an index of names: Rather than having to write in the text, e.g., “`Einstein\index{Einstein, Albert}`” every time, write “`\keyindex{Einstein}`,” which produces “Einstein” in the text and an index entry under “Einstein, Albert”. Of course, the correct index entries must be collected somewhere.

The package distribution also includes a bash script `makenameindex` which produces a key index file for use with `keyindex` from a text file with entries of the form

⟨*last name*⟩, ⟨*first name*⟩
⟨*key*⟩`|`⟨*last name*⟩, ⟨*first name*⟩
⟨*key*⟩`@`⟨*indexentry*⟩

2 Interactions

`keyindex` requires the `ifthen` package.

3 Usage

`\keyindex` `\keyindex[`⟨*index option*⟩`]{`⟨*key*⟩`}` prints the text associated with index key ⟨*key*⟩ (using the hooks `\keyindexformat` and `\missingkeyindexformat`) followed by an index entry for ⟨*key*⟩. The index entry is generated using the hook `\keyindexcommand` and the optional argument ⟨*index option*⟩ is added to the argument of `\keyindexcommand` after a `|`. For instance, `\keyindex[(]{Einstein}` might produce the same as

`Einstein\index{Einstein, Albert|(}`.

`\keyindexprint` `\keyindexprint{`⟨*key*⟩`}` prints only the text corresponding to ⟨*key*⟩, but doesn’t

`\keyindexonly`

---

[*] This document corresponds to keyindex 1.0, dated 2019/04/05.

add anything to the index. `\keyindexonly{`⟨*key*⟩`}[`⟨*index option*⟩`]` only adds the index entry but doesn't print anything in the text.

`\keyindexfile`

`\keyindexfile{`⟨*file*⟩`}` will load the index key definitions from ⟨*file*⟩`.kix`, which is assumed to contain only a list of lines of the form

`\keyindexentry {`⟨*key*⟩`}{`⟨*print text*⟩`}{`⟨*index text*⟩`}`

If it is not called, `keyindex` will use `\jobname.kix` as default. This must be used in the preamble.

`\keyindexformat`
`\missingkeyindexformat`
`\keyindexcommand`

These are hooks to carry out the formatting in the text of the text corresponding to an index entry, or adding the entry to the index. They can be redefined using `\renewcommand`. By default, `\keyindexformat{`⟨*text*⟩`}` just adds ⟨*text*⟩ to the document, `\missingkeyindexformat{`⟨*text*⟩`}` adds `\textbf{`⟨*text*⟩`}`, and `\keyindexcommand{`⟨*text*⟩`}` issues `\index{`⟨*text*⟩`}`.

4 Implementation

4.1 Setup

```
\RequirePackage{ifthen}
```

`\keyindexfile` sets `\@keyindexfile` to its argument; we set `\@keyindexfile` to default to `\jobname.kix`

```
\newcommand*{\keyindexfile}[1]{%
  \renewcommand*{\@keyindexfile}{#1}}
\newcommand*{\@keyindexfile}{\jobname.kix}
```

`\keyindexformat` formats index text; default: just prints argument.

```
\newcommand*{\keyindexformat}[1]{#1}
```

`\missingkeyindexformat` prints index key when not defined; by default, `\textbf`.

```
\newcommand*{\missingkeyindexformat}[1]{%
  \textbf{#1}}
```

`\keyindexcommand{`⟨*index text*⟩`}` adds `{`⟨*index text*⟩`}` to the index; by default, it does `\index{`⟨*index text*⟩`}` but can be redefined.

```
\newcommand*{\keyindexcommand}[1]{\index{#1}}
```

At the beginning of the document, we read the index key definition file

```
\AtBeginDocument{%
  \InputIfFileExists{\@keyindexfile}{%
    \PackageInfo{keyindex}{Using index key definition file
      \@keyindexfile.}}{%
    \PackageWarning{keyindex}{No index key definition file
      \@keyindexfile!}}}
```

`\keyindex[`⟨*index option*⟩`]{`⟨*key*⟩`}` prints text for key ⟨*key*⟩, and indexes it (with `|`⟨*index option*⟩, if present). Printing is done by `\keyindexprint`, indexing by `\keyindexonly`. If the key is undefined, we issue a warning and print the key (using `\missingkeyindexformat`), not its replacement text.

```
\newcommand*{\keyindexprint}[1]{\@ifundefined{kix@e@\detokenize{#1}}{%
    \PackageWarning{keyindex}{Index key \detokenize{#1}
```

```
      undefined}%
    \missingkeyindexformat{#1}}{%
    \keyindexformat{\@nameuse{kix@e@\detokenize{#1}}}}}
\newcommand*{\keyindexonly}[2][]{\@ifundefined{kix@e@\detokenize{#2}}{%
    \PackageWarning{keyindex}{Index key \detokenize{#2}
      undefined}}{%
    \ifthenelse{\equal{#1}{}}{%
      \keyindexcommand{\@nameuse{kix@i@\detokenize{#2}}}}{%
      \keyindexcommand{\@nameuse{kix@i@\detokenize{#2}}|#1}}}}
\newcommand{\keyindex}[2][]{\keyindexprint{#2}\keyindexonly[#1]{#2}}
```

The file `\@keyindexfile` must contain lines of the form `\keyindexentry{`⟨*key*⟩`}{`⟨*text*⟩`}{`⟨*index entry*⟩`}`. It is read at the beginning of the document and for each key, defines commands `\kix@e@`⟨*key*⟩ and `\kix@i@`⟨*key*⟩ which evaluate to ⟨*text*⟩ and ⟨*index entry*⟩, respectively. Duplicate definitions are ignored but generate a warning.

```
\newcommand*{\keyindexentry}[3]{%
  \@ifundefined{kix@e@\detokenize{#1}}{%
    \@namedef{kix@e@\detokenize{#1}}{#2}%
    \@namedef{kix@i@\detokenize{#1}}{#3}}{%
    \PackageWarning{keyindex}{Duplicate definition for keyindex key
      \detokenize{#1} ignored}}}
```