Standard LaTeX 2_ε packages `makeidx` and `showidx`

Johannes Braams David Carlisle Alan Jeffrey
Leslie Lamport Frank Mittelbach Chris Rowley
Rainer Schöpf

2014/09/29

This file is maintained by the LaTeX Project team.
Bug reports can be opened (category `latex`) at
https://latex-project.org/bugs.html.

1 Description

1.1 Makeidx

The package `makeidx` provides two new commands, `\see` and `\printindex`.

`\see` The command `\see` can used in the index to cross reference to other items.

`\printindex` This command can be used to include the sorted and formatted index in the document.

1.2 Showidx

The package `showidx` changes a number of internal LaTeX 2_ε commands in such a way that each index entry is shown in the margin on the page where the entry appears. This package was originally meant to be used with the `report` and `book` document classes, but works with other classes as well. It makes `\flushbottom` the default.

2 The DOCSTRIP modules

The following modules are used in the implementation to direct DOCSTRIP in generating the external files:

makeidx	produce makeidx.sty
showidx	produce showidx.sty
driver	produce a documentation driver file

3 The documentation driver file

The next bit of code contains the documentation driver file for TeX, i.e., the file that will produce the documentation you are currently reading. It can be extracted

from this file by the DOCSTRIP program.

```
⟨*driver⟩
\documentclass{ltxdoc}
\begin{document}
\DocInput{makeindx.dtx}
\end{document}
⟨/driver⟩
```

4 Implementation

4.1 Identification

Announce the package and its version:

```
⟨makeidx⟩\ProvidesPackage{makeidx}
⟨showidx⟩\ProvidesPackage{showidx}
                [2021/10/04 v1.0m Standard LaTeX package]
```

4.2 Makeidx

`\see` This macro discards its second argument (typically a page number) and just prints `\seename` together with the entry the reader is pointed to.

```
⟨*makeidx⟩
\newcommand*\see[2]{\emph{\seename} #1}
```

`\seealso` This macro discards its second argument (typically a page number) and just prints `\alsoname` together with the entry the reader is pointed to. We use `\providecommand` to retain compatibility with existing files that define this macro.

```
\providecommand*\seealso[2]{\emph{\alsoname} #1}
```

`\printindex` This command simply inputs the (formatted) index if it exists, otherwise a warning is issued.

```
\newcommand\printindex{\@input@{\jobname.ind}}
```

`\seename` This package is for documents prepared in the English language. To prepare a version for another language, various English words must be replaced. All the English words that require replacement are defined below in command names.

```
\providecommand\seename{see}
```

We used `\providecommand` in case the command is already defined by a package loaded earlier.

`\alsoname` This macro discards its second argument (typically a page number) and just prints `\alsoname` together with the entry the reader is pointed to. We use `\providecommand` to retain compatibility with existing files that define this macro.

```
\providecommand*\alsoname{see also}
⟨/makeidx⟩
```

4.3 showidx

`\indexbox` This package uses TEX's insert mechanism, therefore it needs to allocate an insert register.

```
⟨*showidx⟩
\newinsert\indexbox
\dimen\indexbox=\maxdimen
```

`\index` This is a modified default definition for the user level `\index` command. The difference is the change of the category code of the space character.

```
\renewcommand\index{\@bsphack\begingroup
                    \@sanitize\catcode32=10\relax\@index}
```

`\makeindex` The same change has to be included in the user level `\makeindex` command, which changes the definition of `\index` to actually write index entries to an external file.

```
\renewcommand\makeindex{\if@filesw \newwrite\@indexfile
  \immediate\openout\@indexfile=\jobname.idx
  \def\index{\@bsphack\begingroup
    \def\protect####1{\string####1\space}\@sanitize
    \catcode32=10 \@wrindex\@indexfile}\typeout
   {Writing index file \jobname.idx }\fi}
```

`\@wrindex` This macro takes care of writing the index entries to a file. The definition is modified to call `\@showidx`.

```
\def\@wrindex#1#2{\let\thepage\relax
   \xdef\@gtempa{\write#1{\string
      \indexentry{#2}{\thepage}}}\endgroup\@gtempa
   \@showidx{#2}\if@nobreak \ifvmode\nobreak\fi\fi\@esphack}
```

`\@index` When the user didn't use the `\makeindex` command, the `\index` macro calls `\@index`, which normally does basically nothing. This package changes the definition to call `\@showidx`, which includes the entry in the list of indexentries on the current page.

```
\def\@index#1{\@showidx{#1}\endgroup\@esphack}
```

`\@showidx` This macro adds the current index entry to the insert `\indexbox`. The `\indexbox` is typeset as a flushleft paragraph.

```
\def\@showidx#1{%
  \insert\indexbox{\small
    \hsize\marginparwidth
    \hangindent\marginparsep \parindent\z@
    \everypar{}\let\par\@@par \parfillskip\@flushglue
    \lineskip\normallineskip
    \baselineskip .8\normalbaselineskip\sloppy
    \raggedright \leavevmode
    \vrule \@height .7\normalbaselineskip \@width \z@\relax
        #1\relax
    \vrule \@height \z@ \@depth .3\normalbaselineskip \@width \z@}}
```

`\raggedbottom` `\flushbottom` The definitions of these macros from the LATEX format are changed here to add the execution of `\@mkidx` to `\@texttop`, which is executed at the top of each page.

```
\renewcommand\raggedbottom{\def\@textbottom{\vskip
```

```
      \z@ plus.0001fil}\let\@texttop\@mkidx}
\renewcommand\flushbottom{\let\@textbottom\relax
                          \let\@texttop\@mkidx}
```

`\@mkidx` This macro actually typesets the box containing all the index entries on the current page. They will occur on the left or the right side of the text, or both, depending on the setting of the switches `\if@twocolumn` and `\if@twoside`.

```
\def\@mkidx{\vbox to \z@{\hbox{\if@twocolumn
    \if@firstcolumn \@leftidx \else \@rightidx \fi
  \else \if@twoside \ifodd\c@page \@rightidx
                    \else \@leftidx \fi
        \else \@rightidx \fi
  \fi
  \box\indexbox}\vss}}
```

`\@leftidx` `\@rightidx` These macros give the amount of displacement for the `\indexbox`.

```
\def\@leftidx{\hskip-\marginparsep \hskip-\marginparwidth}
\def\@rightidx{\hskip\columnwidth \hskip\marginparsep}
```

To make this work we have to execute either `\raggedbottom` or `\flushbottom`. Assuming this package is used most often with the document classes `report` and `book`, we execute `\flushbottom`.

```
\flushbottom
⟨/showidx⟩
```