A new implementation of LaTeX's indexing commands

David M. Jones

2025/02/06

1 Introduction

This style file reimplements LaTeX's indexing macros to provide better and more robust support for indexes. In particular, it provides the following features:[1]

1. Support for multiple indexes.
2. Indexing of items by counters other than the page number.
3. A $*$-variant of the `\index` command that, in addition to putting it's argument in the index, also typesets it in the running text.
4. The `showidx` style option has been merged into this file. The command `\proofmodetrue` can be used to enable the printing of index entries in the margin of pages. The size and style of font can be controlled with the `\indexproofstyle` command.
5. A two-stage process, similar to that used to create tables of contents, for creating the raw index files. This means that when processing a portion of a document using the `\includeonly` command, the index entries from the rest of the document are not lost.
6. A more robust `\index` command. In particular, it no longer depends on `\catcode` changes to work properly, so the new `\index` command can be used in places that the original couldn't, such as inside the arguments of other macros.

2 Creating an index with LaTeX

Conceptually, there are four stages to creating an index. First, LaTeX must be informed of your intention to include an index in your document. Second, you must add appropriate markup commands to your document to tell LaTeX what to put in the index. Third, after LaTeX has been run on your document, the raw index information must be processed and turned into a form that LaTeX can process to typeset the index. Finally, the finished index must be inserted at the appropriate point in your document.

[1] Earlier versions of this package provided a "shortindexing" feature (see below for description). This feature is now deprecated and will be removed in a future release of this package.

In LaTeX, these steps are accomplished with the commands `\makeindex`, `\index`, `\printindex`, and (typically) with the auxiliary program MakeIndex. For example, assuming that your main file is called `foo.tex`, `\makeindex` opens the file `foo.idx` and initializes it for holding the raw index entries, and `\index` is used to add raw index entries into `foo.idx`. Then the raw index file is processed by MakeIndex, which puts the finished index in `foo.ind`. Finally, the `\printindex` command is used in your LaTeX document to indicate where the file `foo.idx` should be inserted, i.e., where the index should appear in your document.

The `index` package modifies the `\makeindex`, `\index`, and `\printindex` commands, as described below.

3 The user interface

There are four pieces of information associated with each index:

1. A short, unique tag that identifies the index.
2. The extension of the output file where the raw index information will be put by LaTeX.
3. The extension of the input file where the processed information created by MakeIndex will be stored to be read in later by LaTeX.
4. The title of the index.

`\newindex` Correspondingly, the `\newindex` command has four required arguments. For example, to declare an author index, you might use the following:

```
\newindex{aut}{adx}{and}{Name Index}
```

Here, `aut` is the tag used to identify the author index, and "Name Index" is the title of the index. If the name of your main file is `root.tex`, then LaTeX will write the raw index entries to the file `root.adx`, and you will execute the following MakeIndex command to process the author index:

```
makeindex -o root.and root.adx
```

By default, the `\index` tags its argument with the page number (i.e., the value of `\thepage`), but occasionaly you may want to index items according to a different counter. For example, you may want an index that contains figure numbers instead of page numbers. To accomodate, this, the `\newindex` command takes an optional argument, which is the name of the command that generates the number that should be included in the index. For instance, to include the number of a figure, you might say

```
\newindex[thefigure]{fig}{fdx}{fnd}{Figures}
```

However, this introduces a new technicality: When creating an index with page numbers, the choice of which page number is to be written to the `aux` file should be deferred until the page containing the entry is shipped out to the `dvi` file, otherwise the wrong number will sometimes be chosen. However, when using counters other than the page counter, one normally wants the opposite behaviour: the number written to the `aux` file should be chosen immediately, otherwise every

item on a given page will be tagged with the number of the last item on that page. So, when a counter is specified using the optional argument of `\newindex`, it is assumed that the counter should be evaluated immediately. If for some reason you need the choice to be deferred until the page is written to the `dvi` file, you can force this behaviour by putting a $*$ *after* the optional argument:

```
\newindex[thefigure]*{fig}{fdx}{fnd}{Figures}
```

(One consequence of this scheme is that if, for some reason, you need the choice of page number to be made immediately instead of being deferred until a page is shipped out to the `dvi` file, you can acomplish this by beginning your index declaration with

```
\newindex[thepage]*
```

`\renewindex` The `\renewindex` command takes the same arguments as the `\newindex` command and can be used to redefine indexes that have been previously declared.

`\makeindex` For backwards compatibility, the `\makeindex` command is redefined to use `\newindex`. It is essentially equivalent to

```
\newindex{default}{idx}{ind}{Index}
```

The index labeled `default` is special: it is the one that will be used by `\index` and `\printindex` unless another index is specified (see below).

`\printindex` The `\printindex` command is modified by the addition of an optional argument, which is the tag of the index that should be printed.

`\index` The `\index` command is modified in two ways. First, there is a $*$-variant of the command that, in addition to putting its argument into an index, also typesets it on the page. Second, `\index` now takes an optional argument to indicate which index the new entry should be added to. If given, the optional argument should be the identifying tag of a previously-defined index. If no such tag is supplied, the `default` index (such as that opened by `\makeindex` above) is used.

`\shortindexingon` `\shortindexingoff` Perhaps the most dubious feature of `index.sty` is that it allows you to define the characters `^` and `_` to be abbreviations for `\index*` and `\index` outside of math mode. These abbreviations are enabled by the `\shortindexingon` command and disabled by the `\shortindexingoff` command. The scope of both of these latter commands is local to the current group. (This might be useful, for example, if you wanted the abbreviations turned on throughout most of the documentation, but turned off in one particular environment.) In addition, `shortindexingon` can be used as an environment if that seems appropriate. **Warning: This feature is deprecated and will disappear in a future release of this package.**

`shortindexingon` (*env.*)

`\proofmodetrue` `\proofmodefalse` `\indexproofstyle` As mentioned above, the `showidx` document-style option has been merged into `index.sty`. It can be turned on with `\proofmodetrue` and turned off with `\proofmodefalse`. When it is turned on, all index entries[2] will be put in the margin of the page where they appear. By default, they appear in the typewriter font at `\footnotesize`, but the user can override this with the `\indexproofstyle` command; for example,

```
\indexproofstyle{\footnotesize\it}
```

will cause them to be put in italics instead.

`\disableindex` There are some circumstances where it might be helpful to suppress the writing

[2] Well, most, at least. There are some circumstances under which the index entries won't show up in the proofs, although they will show up in the index.

of a particular index. The `\disableindex` command is provided for this purpose. It takes one argument, a comma-separated list of tags of the indexes that should be disabled. This command should come *before* the declarations for the indexes that are being disabled[3]. One situation where the `\disableindex` command might be useful is if there are so many indexes that you are exhausting TEX's supply of output streams[4]. For example, suppose you have 10 indexes, but only 5 output streams available for indexes. Then you could add a `\disableindex` command to the top of your file to suppress the writing of all but 5 of the indexes. (Note that the index entries would still get written to the `aux` file; they just wouldn't get copied to the individual raw index files at the end of the run.) At the end of the run, you could then re-run your main file a couple of times with different indexes disabled until you had created all of the raw index files. This is somewhat clumsy, but safer than any alternative I've come up with[5].

4 Caveats

In order to implement this style file, it's been necessary to modify a number of LATEX commands seemingly unrelated to indexing, namely, `\@starttoc`, `\raggedbottom`, `\flushbottom`, and `\addcontents`. Naturally, this could cause incompatibilities between `index.sty` and any style files that either redefine these same commands or make specific assumptions about how they operate. See Section 6 for explanations of why these various commands needed modification.

The redefinition of `\@starttoc` is particularly bad, since it introduces an incompatibility with the AMS document classes. This will be addressed soon.

Unfortunately, it's also been necessary to modify the `theindex` environment, so if you don't like the default LATEX definition, you'll need copy the definition of `theindex` from this file and modify it appropriately.

In the current implementation, `index.sty` uses one output stream for each index. Since there are a limited number of output indexes, this means that there is a limit on the number of indexes you can have in a document. See the description of `\disableindex` for a fuller discussion of this problem and one way around it.

5 To do's

It might be nice if the `\index*` command parsed its argument so that, for example, instead of writing '`\index{sin@$\sin$}$\sin$`', one could write '`index*{sin@$\sin$}`'. However, this is fraught with numerous dangers, and I'm both too lazy and too cowardly to undertake it now.

It would be reasonable to add support for `\makeglossary` and similar things, if they were well-defined enough to decide what the general syntax for defining

[3] This limits its usefulness somewhat, but since the output file for an index is opened when the index is declared, the damage has already been done. We could close the file, but we can't prevent a new output stream from being allocated and we can't keep the old file from being truncated.

[4] TEX only has 16 output streams, which are allocated with the `\newwrite` command. The standard LATEX styles use from 3 to 7 of these, which should leave room for up to 9 indexes. Of course, if you have extra output files, then there will be fewer output streams left for indexes.

[5] A less clumsy (for the user, at least) solution would be to read the `aux` file multiple times at the end of the run, each time writing just one of the raw index files. The main disadvantage of this scheme at present is that it would require a modification of `\enddocument`.

them should be.

The documentation should be carefully read, edited, and finished, especially since it's still based on the 2.09 version, even though a few substantial changes have been made for the LaTeX 2_ε version.

For some truly outlandish ideas, see the file `TODO` in the distribution.

6 The code

As is customary, identify this as a LaTeX 2_ε package.

```
⟨*style⟩
\NeedsTeXFormat{LaTeX2e}[2024/11/01]

\ProvidesPackage{index}[2025/02/06 v4.04 Improved index support (dmj)]
```

`\disableindex` The `\disableindex` should come before the declarations of the indexes it refers to. (Question: If an index has been disabled, should it show up in index proofs? Maybe there should be a separate command to disable index proofs on and index-by-index basis.)

```
\def\disableindex#1{%
    \@for\@tempa:=#1\do{%
        \@namedef{disable@\@tempa}{}%
        \@ifundefined{tf@\@tempa}{}{%
            \PackageWarningNoLine{index}{It's too late to disable
                the `\@tempa' index;\MessageBreak
                \jobname.\@tempa\space has already
                been opened for output. You \MessageBreak
                should put the \string\disableindex\space command
                before\MessageBreak
                the declaration of the `\@tempa' index}%
        }%
    }%
}
```

`\if@newindex` `\newindex` `\renewindex` The `\newindex` and `\renewindex` commands are defined on analogy with the `\[re]newcommand` macros. Each index is identified by a unique tag, which is specified in the first required argument of `\newindex`. Much of the information about the index labeled ⟨*tag*⟩ is kept in the macro `\idx@`⟨*tag*⟩, so we can check to see if a particular index has already been defined by checking whether `\idx@`⟨*tag*⟩ is defined. `\newindex` and `\renewindex` both check to see if their first argument is already associated with an index and then either issue an appropriate error message or call `\def@index`.

The `\if@newindex` flag will be used to keep `\renewindex` from re-allocating `\write` and `\toks` registers later. The `\if@tempswa` switch will be used to determine whether the `\write`s for this index should be done `\immediate`ly or not.

```
\newif\if@newindex

\def\newindex{%
    \@tempswafalse
    \@ifnextchar[{\@tempswatrue\x@newindex}{\x@newindex[thepage]}%
}

```

```
\def\x@newindex[#1]{%
    \@ifstar {\@tempswafalse\y@newindex{#1}}
             {\y@newindex{#1}}%
}

\def\y@newindex#1#2{%
    \@ifundefined{idx@#2}%
        {\@newindextrue\def@index{#1}{#2}}%
        {%
            \@latexerr{Index type `\string#2' already defined}\@ehc
            \expandafter\@gobble\@gobbletwo
        }%
}

\def\renewindex{%
    \@tempswafalse
    \@ifnextchar[{\@tempswatrue\x@renewindex}{\x@renewindex[thepage]}%
}

\def\x@renewindex[#1]{%
    \@ifstar {\@tempswafalse\y@renewindex{#1}}
             {\y@renewindex{#1}}%
}

\def\y@renewindex#1#2{%
    \@ifundefined{idx@#2}%
        {%
            \@newindextrue
            \@latexerr{Index type `\string#2' not defined}\@ehc
        }%
        {\@newindexfalse}%
    \def@index{#1}{#2}%
}
```

`\@preamblecmds` Neither `\newindex`, `\renewindex`, nor `\disableindex` should be used anywhere except inside style files or in the preamble of a document, so we add them to the `\@preamblecmds` list.

```
\@onlypreamble\newindex
\@onlypreamble\renewindex
\@onlypreamble\disableindex
```

`\def@index` `\def@index` does most of the work. First, it picks up the first three arguments of the `\[re]newindex` command and stores the second two in an appropriate `\idx@` macro. The title of the index is treated differently, however, since it is potentially fragile in a particularly odd way. To prevent mishaps, it is stored in a token register. In addition to stashing away the information about the index, `\def@index` also opens an appropriate output file if we are writing auxiliary files (i.e., unless `\nofiles` is in effect).

```
\def\def@index#1#2#3#4{%
    \@namedef{idx@#2}{#3:#4:#1}%
    \expandafter\let\csname if@immediate@#2\endcsname\if@tempswa
    \if@filesw
        \if@newindex
```

```
            \expandafter\newtoks\csname idxtitle@#2\endcsname
        \fi
        \@ifundefined{disable@#2}{%
            \if@newindex
                \expandafter\newwrite\csname tf@#2\endcsname
            \else
                \immediate\closeout\@nameuse{tf@#2}%
            \fi
            \immediate\openout\@nameuse{tf@#2}\jobname.#3 %
            \PackageInfo{index}{Writing index file \jobname.#3}%
        }
        {\PackageInfo{index}{Index `#2' disabled -- not opening
                    \jobname.#3}}%
    \fi
    \expandafter\csname idxtitle@#2\endcsname
}
```

`\@first` `\@second` `\@third` These are useful macros for retrieving individual components of an index specification.

```
\def\@first#1:#2:#3\@nil{#1}

\def\@second#1:#2:#3\@nil{#2}

\def\@third#1:#2:#3\@nil{#3}
```

`\@nearverbatim` `\@nearverbatim\foo` is much like `\meaning\foo`, except that it suppresses the "`macro ->`" string produced when `\meaning` expands a macro. It is used by `\@wrindex` to produce an "almost verbatim" copy of their arguments. This method replaces the use of `\@sanitize` from latex.tex and allows indexing macros to be used in places (such as inside macro arguments) where the original `\index` command could not. Thanks to Donald Arseneau ⟨`asnd@erich.triumf.ca`⟩ for pointing out this trick to me. (For more information on this trick, see Dirty Trick #3 of the TEXbook, page 382).

As defined, nearverbatim only works on macros. It would be nice if it could work with other tokens, but it's more important that it work only by expansion, which means we can't put in tests to see what the next token is.

```
\def\@nearverbatim{\expandafter\strip@prefix\meaning}
```

Now we define the `\index` macro itself. The following definitions are adapted from `latex.tex` v2.09 ⟨25 March 1992⟩.

`\makeindex` First we redefine `\makeindex` to define the default index using `\newindex`. We use `\edef` to make sure that `\indexname` gets expanded here. Otherwise we'll get into an infinite loop later on when we try to redefine `\indexname` inside the `\theindex` environment.

Unfortunately, this means that if the user changes `\indexname` in the preamble, the index will come out with the wrong heading.

```
\edef\makeindex{%
    \noexpand\newindex{default}{idx}{ind}{\indexname}%
}
```

`\if@silentindex` `\if@addtoindex` `\if@proofmode` We need three new flags. The first, `\if@silentindex`, indicates whether the entry should be typeset in running text, as well as written out to the index;

this is used to implement the `\index*` command. The second, `\if@addtoindex`, indicates whether entries should be written to the index; this is used to disable the `\index` command inside of page headings and tables of contents. The third, `\ifproofmode`, indicates whether index entries should be put in the margin of the page for proofing purposes.

```
\newif\if@silentindex\@silentindextrue

\newif\if@addtoindex\@addtoindextrue

\newif\ifproofmode\proofmodefalse
```

`\index` `\index` will be made self-protecting (a la `\em`, etc.) so it can be used inside, for `\p@index` example, sectioning commands. Unfortunately, to really make `\index` robust, we `\x@index` have to redefine some of LaTeX's commands for dealing with tables of contents and page headings. (See below.) ∗sigh∗

```
\protected\def\index{\protect\p@index}

\def\p@index{%
    \if@silentindex\@bsphack\fi
    \@ifstar{\@silentindexfalse\@xindex}{\@silentindextrue\@xindex}%
}

\def\@xindex{\@ifnextchar[{\@index}{\@index[default]}}
```

`\@index` The following is much more complicated than it should have to be. First, note the `\@@index` check to see if `\index` is equal to `\@gobble`. This is so I don't have to redefine `\@wrindex` `\@outputpage`, which temporarily disables `\label`, `\index`, and `\glossary` by `\let`'ing them equal to `\@gobble`. (For this reason, we have to be very careful to make sure that `\index` has expanded to `\p@index` before it gets to `\@outputpage`.) Second, note that if `\if@addtoindex` is false, we don't complain about undefined index types. This is because if your page headings, for example, are being typeset in all uppercase, you might end up with something like `\index[AUT]...` instead of `\index[aut]...`.

```
\def\@index[#1]{%
    \ifx\index\@gobble
        \@addtoindexfalse
    \fi
    \def\@tempf{%
        \begingroup
            \@sanitize
            \@@index{#1}%
    }%
    \if@addtoindex
        \@ifundefined{idx@#1}%
            {%
              \def\@tempf{%
                  \@latexerr{Index type `\string#1' undefined}%
                  \@ehc
                  \@silentindextrue
                  \@gobble
              }%
            }%
```

```
            {}%
    \fi
    \@tempf
}

\def\@@index#1#2{%
    \endgroup
    \if@addtoindex
        \if@filesw\@wrindex{#1}{#2}\fi
        \ifproofmode\@showidx{#2}\fi
    \fi
    \if@silentindex
        \expandafter\@esphack
    \else
        \@silentindextrue#2%
    \fi
}

\def\@wrindex#1#2{%
    \begingroup
        \def\@tempa{#2}%
        \edef\@tempb{\@nameuse{idx@#1}}%
        \edef\@tempb{\expandafter\@third\@tempb\@nil}%
        \csname if@immediate@#1\endcsname \else
            \expandafter\let\csname\@tempb\endcsname\relax
        \fi
        \edef\@tempa{%
           \write\@auxout{%
              \string\@writefile{#1}{%
                  \string\indexentry{\@nearverbatim\@tempa}%
                                    {\@nameuse{\@tempb}}%
              }%
           }%
        }%
    \expandafter\endgroup\@tempa
    \if@nobreak\ifvmode\nobreak\fi\fi
}
```

\seename
\see
\printindex
\@printindex

The following are adapted from `makeidx.sty`, v2.09 ⟨21 Oct 91⟩. `\index@prologue` adapted from `doc.dtx`. `theindex` based on version from `classes.dtx`, v1.3g, 26 June 1995.

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

\providecommand*{\see}[2]{\emph{\seename} #1}

\@ifclassloaded{article}{%

    \renewenvironment{theindex}{%
        \edef\indexname{\the\@nameuse{idxtitle@\@indextype}}%
        \if@twocolumn
            \@restonecolfalse
        \else
            \@restonecoltrue
        \fi
```

```
        \columnseprule \z@
        \columnsep 35\p@
        \twocolumn[%
            \section*{\indexname}%
            \ifx\index@prologue\@empty\else
                \index@prologue
                \bigskip
            \fi
        ]%
        \@mkboth{\MakeUppercase\indexname}%
                {\MakeUppercase\indexname}%
        \thispagestyle{plain}%
        \parindent\z@
        \parskip\z@ \@plus .3\p@\relax
        \let\item\@idxitem
    }{%
        \if@restonecol
            \onecolumn
        \else
            \clearpage
        \fi
    }
}{%
    \renewenvironment{theindex}{%
        \edef\indexname{\the\@nameuse{idxtitle@\@indextype}}%
        \if@twocolumn
            \@restonecolfalse
        \else
            \@restonecoltrue
        \fi
        \columnseprule \z@
        \columnsep 35\p@
        \twocolumn[%
            \@makeschapterhead{\indexname}%
            \ifx\index@prologue\@empty\else
                \index@prologue
                \bigskip
            \fi
        ]%
        \@mkboth{\MakeUppercase\indexname}%
                {\MakeUppercase\indexname}%
        \thispagestyle{plain}%
        \parindent\z@
        \parskip\z@ \@plus .3\p@\relax
        \let\item\@idxitem
    }{%
        \if@restonecol
            \onecolumn
        \else
            \clearpage
        \fi
    }
}
```

```
\def\printindex{\@ifnextchar[{\@printindex}{\@printindex[default]}}

\def\@printindex[#1]{%
    \@ifnextchar[{\@print@index[#1]}{\@print@index[#1][]}%
}

\long\def\@print@index[#1][#2]{%
    \def\@indextype{#1}%
    \long\def\index@prologue{#2}%
    \@ifundefined{idx@#1}%
        {\@latexerr{Index type `\string#1' undefined}\@ehc}%
        {%
            \edef\@tempa{\@nameuse{idx@#1}}%
            \edef\@tempb{%
                \jobname.\expandafter\@second\@tempa\@nil
            }%
            \edef\@tempc{%
                \jobname.\expandafter\@first\@tempa\@nil
            }%
            \InputIfFileExists{\@tempb}{}{%
                \typeout{No file \@tempb: Run makeindex -o
                    \@tempb \space \@tempc}%
            }%
        }%
}
```

`\@indexstar@` Now we set things up for `\shortindexing`.[6] First, we define a one-token shorthand for `\index*`. This will be needed in the definition of `\idx@activehat`.

```
\def\@indexstar@{\index*}
```

`\idx@activehat` `\idx@activebar` Next, we define the values that `^` and `_` will have when shortindexing is turned on.

```
\def\idx@activehat{%
    \relax
    \ifmmode\expandafter\sp\else\expandafter\@indexstar@\fi
}

\def\idx@activebar{%
    \relax
    \ifmmode\expandafter\sb\else\expandafter\index\fi
}
```

`\shortindexingon` `\shortindexingoff` Now we define the `\shortindexingon` and `\shortindexinoff` commands to turn shortindexing on and off (surprise!). `\shortindexingon` saves the old definitions and `\catcode`'s of `^` and `_` so they can later be restored by `\shortindexingoff`. Both of these make their changes local to any enclosing group, so they can be used as declarations to disable or enable shortindexing temporarily. In addition, `shortindexingon` can also be used as an environment.

This is potentially very confusing. My basic rationale (if it can be described as such) was that under normal circumstances, one would put `\shortindexingon` in the preamble of one's document, and never want to turn it off. `\shortindexingoff`

[6] **Warning:** This feature is deprecated and will be removed entirely in a future release of this package.

is an attempt to make allowance for the contingency that someone might want to turn shortindexing off, either permanently or temporarily.

```
\newif\if@shortindexing

\begingroup

    \catcode`\^\active
    \catcode`\_\active

    \gdef\shortindexingon{%
        \@shortindexingtrue
        \chardef\old@idxhatcode\catcode`\^\relax
        \chardef\old@idxbarcode\catcode`\_\relax
        \catcode`\^\active
        \catcode`\_\active
        \let\old@idxhat ^%
        \let\old@idxbar _%
        \let^\idx@activehat
        \let_\idx@activebar
    }

    \gdef\shortindexingoff{%
        \if@shortindexing
            \@shortindexingfalse
            \let^\old@idxhat
            \let_\old@idxbar
            \catcode`\^\old@idxhatcode
            \catcode`\_\old@idxbarcode
        \fi
    }

\endgroup
```

Now we take some code from `showidx.sty` and merge it into our new system. There are four reasons for redefining the commands here rather than just inputting `showidx.sty` (or requiring the user to do so). First, `showidx.sty` ends with a call to `\flushbottom`, which I want to avoid. Second, the instructions for successfully using `showidx.sty` along with `index.sty` would be somewhat tricky. This way, I can just tell users not to use `showidx.sty` at all. Third, I need to make some alterations to `\@showidx` anyway. In particular, (a) I need to add the `\@sanitizeat` command so this works correctly with AMS-LaTeX and (b) I want to add the `\indexproofstyle` command so the user can customize the size and font used for the index proofs. Finally, `showidx.sty` has at least two annoying bugs in it. See the edit-history for version 2.01 for a description.

`\@indexbox` This code is adapted from `showidx.sty`, v2.09 ⟨16 Jun 1991⟩.

```
\newinsert\@indexbox

\dimen\@indexbox\maxdimen
```

`\@sanitizeat` The definition of `\@sanitizeat` is slightly tricky, since we need `@` to be active when this macro is defined, but we also need it to be part of the control sequence name.

```
\begingroup
    \catcode`\@\active
    \expandafter\gdef\csname\string @sanitizeat\endcsname
        {\def @{\char`\@}}
\endgroup

\newtoks\indexproofstyle

\indexproofstyle{\footnotesize\reset@font\ttfamily}

\def\@showidx#1{%
    \insert\@indexbox{%
        \@sanitizeat
        \the\indexproofstyle
        \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@\relax
    }%
    \ifhmode\penalty\@M \hskip\z@skip\fi
}

\def\@leftidx{\hskip-\marginparsep \hskip-\marginparwidth}

\def\@rightidx{\hskip\columnwidth \hskip\marginparsep}

\def\@mkidx{%
    \vbox to \z@{%
        \rlap{%
            \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
    }%
}

\def\raggedbottom{%
    \def\@textbottom{\vskip\z@ plus.0001fil}%
    \let\@texttop\@mkidx
}
```

\indexproofstyle
\@showidx
\@leftidx
\@rightidx
\@mkidx
\raggedbottom
\flushbottom
\@texttop

```
\def\flushbottom{\let\@textbottom\relax \let\@texttop\@mkidx}

\let\@texttop\@mkidx
```

Now, this next bit really gets up my nose. The only way to make sure that the `\index` command gets handled correctly when used inside of sectioning commands is to redefine a bunch of LaTeX's table of contents and running-heads macros. ∗blech∗ Fragility rears its ugly head again.

These are based on `latex.tex` 2.09 ⟨25 March 1992⟩.

`\addcontentsline` We need to redefine `\addcontentsline` to keep it from expanding `\index` commands too far. In particular, we have removed `\index` from the list of macros that are set equal to `\@gobble` and we substitute `\@vwritefile` for `\@writefile`. This latter change also means that we can simplify the definition of `\protect` somewhat.

```
\CheckCommand\addtocontents[2]{%
  \protected@write\@auxout
      {\let\label\@gobble \let\index\@gobble \let\glossary\@gobble}%
      {\string\@writefile{#1}{#2}}%
}

\renewcommand{\addtocontents}[2]{%
    \protected@write\@auxout
      {\let\label\@gobble \let\glossary\@gobble}%
      {\string\@writefile{#1}{#2}}%
}
```

`\@starttoc` We need to redefine `\@starttoc` to `\@addtoindexfalse` so that items don't get written to the index from within tables of contents. The only change here is the addition of `\@addtoindexfalse`.

Unfortunately, this will break pretty badly with the AMS document classes, since they redefine `\@starttoc` to take two arguments rather than one. This must be addressed.

```
\let\old@starttoc\@starttoc

\renewcommand{\@starttoc}[1]{%
    \begingroup
        \@addtoindexfalse
        \old@starttoc{#1}%
    \endgroup
}
```

7 Finale

The usual `\endinput` to ensure that random garbage at the end of the file doesn't get copied by `docstrip`.

```
\endinput
```

8 Edit history

v1.00 (4 Mar 1993) initial version, posted to comp.text.tex.

v1.01 (4 Mar 1993) added `\renewindex` command and checking to make sure index is (or is not) defined in `\newindex`, `\index` and `\printindex`. Also tightened up the code in various places and added check to make sure file is only loaded once.

v2.00 (24 Mar 1993) added support for `\index*`, proofmode, `\shortindexingon` and `\shortindexingoff`.

v2.01 (24 Jun 1993) Fixed 3 bugs. (1) If proofmode was turned on, then something like "`\indexWORDWORD`" would suppress the hyphenation of WORD. This was fixed by adding "`\penalty\@M\hskip\z@skip`" to the end of `\@showidx`. (This is just the definition of `\allowhyphens` borrowed from `german.sty`, v2 ⟨4 Nov 1988⟩). (2) The `\hbox` in `\@mkidx` was being set at its natural width, which had a tendency to interfere with the width of the page. The `\hbox` is now replaced by `\rlap`. (3) If the title of an index (i.e., the fourth argument of `\newindex`) contained a particularly fragile command like `\d`, havoc would ensue when `\theindex` tried to extract the title. Titles are now kept in token registers to prevent such unpleasantness. Bugs (2) and (3) were reported by Dominik Wujastyk ⟨`D.Wujastyk@ucl.ac.uk`⟩ on 24 June 1993. Note that bugs (1) and (2) are actually bugs in showidx.sty, v2.09 ⟨16 Jun 1991⟩.

v2.02 (25 Jun 1993) Rewrote the code that implements the short indexing commands (`^` and `_`) to make index.sty compatible with other style files that need to make `^` and `^` active in some contexts. See the code for more details.

v2.03 (30 Jun 1993) Once again rewrote the code that implements the short indexing commands. Dumped the shortindexing environment and rewrote the `\shortindexingon` and `\shortindxingoff` commands to save and restore the `\catcode`'s and meanings of `^` and `^` in the safest possible (I hope) order. Also added the `\if@shortindexing` flag to keep `\shortindexingoff` from doing anything if it is called outside of the scope of a `\shortindexingon` command. (Question: Should `\shortindexingon` check that flag before doing anything?)

v2.04 (beta) (14 Jul 1993) Added `\disableindex` command. Added `\newindex` and `\renewindex` to `\@preamblecmds`. Added `\if@newindex` flag to `\@newindex` to prevent `\renewindex` from re-allocating new `\write` and `\toks` registers. Rewrote using `doc.sty` and DocStrip. Also cleaned up the code somewhat.

v3.00 (15 Jul 1993) Made further minor tweaks to code and internal documentation. Booted version number up to 3.00 and released on the world.

v3.01 (19 Jul 1993) Fixed DocStrip CheckSum.

v3.02 (15 Sep 1993) Corrected spelling of `\@shortindexingfalse` in definition of `\shortindexingoff`. Thanks to Hendrik G. Seliger ⟨`hank@Blimp.automat.uni-essen.de`⟩ for this bug report. Also added redefinitions of `\@leftmark` and `\@rightmark` to fix a bug reported by Dominik Wujastyk ⟨`D.Wujastyk@ucl.ac.uk`⟩.

v3.03 (beta) (20 Feb 1994) Added `\long` to the definition of `\@ifundefined` to cover the unlikely contingency that someone wanted to use, for example, `\string\par` in the middle of a control sequence name. Added an optional argument to `\newindex` to specify which counter to use in place of `\thepage`. The first change was suggested by Martin Schröder ⟨`l15d@zfn.uni-bremen.de`⟩; the second was suggested independently by Schröder and Stefan Heinrich Höning ⟨`hoening@pool.informatik.rwth-aachen.de`⟩. The `\@newindex` command was renamed `\def@index`. Also fixed the `\disableindex` command.

v3.04 (7 Mar 1994) Rewrote the user documentation (Sections 1–5) and released on the world. Also deleted some extraneous spaces that had crept into some macros.

v4.00beta, (20 Feb 1995) Preliminary conversion to a native LaTeX 2ε package. Fixed `\@printindex` to work under LaTeX 2ε (bug reported by Carsten Folkertsma ⟨`cai@butler.fee.uva.nl`⟩). Removed much code that had been put in to work around various ancient versions of LaTeX 2.09. Added `\index@prologue` support (modelled on `doc.sty`) at suggestion of Nick Higham ⟨`higham@ma.man.ac.uk`⟩.

v4.01beta (28 Sep 1995) Rewrote as a LaTeX 2ε package (finally!). Changes too numerous to list, but in general deleted some now-superfluous code, replaced some tricks by tricks from the LaTeX 2ε kernel, and added some bullet-proofing. Much still remains to be done, but this should be good enough for testing.

Changed definition of `\protect` in `\markright` and `\markboth` to fix bug reported by Dominik Wujastyk.

??? (5 Jan 2004)

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.