

# A new implementation of L<sup>A</sup>T<sub>E</sub>X's indexing commands

David M. Jones

2025/02/03

## 1 Introduction

This style file reimplements L<sup>A</sup>T<sub>E</sub>X's indexing macros to provide better and more robust support for indexes. In particular, it provides the following features:<sup>1</sup>

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 L<sup>A</sup>T<sub>E</sub>X

Conceptually, there are four stages to creating an index. First, L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X what to put in the index. Third, after L<sup>A</sup>T<sub>E</sub>X has been run on your document, the raw index information must be processed and turned into a form that L<sup>A</sup>T<sub>E</sub>X can process to typeset the index. Finally, the finished index must be inserted at the appropriate point in your document.

---

<sup>1</sup>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 occasionally 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 accommodate, 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 accomplish 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` 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

`\shortindexingoff` 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.**

`\proofmodetrue` 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<sup>2</sup> 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

<sup>2</sup>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<sup>3</sup>. 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<sup>4</sup>. 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<sup>5</sup>.

## 4 Caveats

In order to implement this style file, it's been necessary to modify a number of L<sup>A</sup>T<sub>E</sub>X commands seemingly unrelated to indexing, namely, `\starttoc`, `\raggedbottom`, `\flushbottom`, `\addcontents`, `\markboth`, and `\markright`. 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 L<sup>A</sup>T<sub>E</sub>X 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.

---

<sup>3</sup>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.

<sup>4</sup>TeX only has 16 output streams, which are allocated with the `\newwrite` command. The standard L<sup>A</sup>T<sub>E</sub>X 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.

<sup>5</sup>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`.

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 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 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> version.

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

## 6 The code

As is customary, identify this as a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> package.

```

1 <*style>
2 \NeedsTeXFormat{LaTeX2e}[2024/11/01]
3
4 \ProvidesPackage{index}[2025/02/03 v4.03 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.)

```

5 \def\disableindex#1{%
6   \@for\@tempa:=#1\do{%
7     \@namedef{disable@\@tempa}{}%
8     \@ifundefined{tf@\@tempa}{}{%
9       \PackageWarningNoLine{index}{It's too late to disable
10        the '\@tempa' index;\MessageBreak
11        \jobname.\@tempa\space has already
12        been opened for output. You \MessageBreak
13        should put the \string\disableindex\space command
14        before\MessageBreak
15        the declaration of the '\@tempa' index}%
16     }%
17   }%
18 }

```

`\if@newindex` The `\newindex` and `\renewindex` commands are defined on analogy with the `\newindex` `\[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  $\langle tag \rangle$  is kept in the macro `\idx@ $\langle tag \rangle$` , so we can check to see if a particular index has already been defined by checking whether `\idx@ $\langle tag \rangle$`  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 `\writes` for this index should be done `\immediately` or not.

```

19 \newif\if@newindex
20
21 \def\newindex{%
22   \@tempswafalse
23   \@ifnextchar[{\@tempswatrue\x@newindex}{\x@newindex[thepage]}%

```

```

24 }
25
26 \def\x@newindex[#1]{%
27   \@ifstar {\@tempwafalse\y@newindex{#1}}
28             {\y@newindex{#1}}%
29 }
30
31 \def\y@newindex#1#2{%
32   \@ifundefined{idx@#2}%
33     {\@newindextrue\def@index{#1}{#2}}%
34     {%
35       \@latexerr{Index type '\string#2' already defined}\@ehc
36       \expandafter\@gobble\@gobbletwo
37     }%
38 }
39
40 \def\renewindex{%
41   \@tempwafalse
42   \@ifnextchar[{\@tempwatrue\x@renewindex}{\x@renewindex[thepage]}%
43 }
44
45 \def\x@renewindex[#1]{%
46   \@ifstar {\@tempwafalse\y@renewindex{#1}}
47             {\y@renewindex{#1}}%
48 }
49
50 \def\y@renewindex#1#2{%
51   \@ifundefined{idx@#2}%
52     {%
53       \@newindextrue
54       \@latexerr{Index type '\string#2' not defined}\@ehc
55     }%
56     {\@newindexfalse}%
57   \def@index{#1}{#2}%
58 }

```

`\@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.

```

59 \@onlypreamble\newindex
60 \@onlypreamble\renewindex
61 \@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).

```

62 \def\def@index#1#2#3#4{%
63   \@namedef{idx@#2}{#3:#4:#1}%
64   \expandafter\let\csname if@immediate@#2\endcsname\if@tempwa

```

```

65 \if@filesw
66   \if@newindex
67     \expandafter\newtoks\csname idxtitle@#2\endcsname
68     \fi
69     \@ifundefined{disable@#2}{%
70       \if@newindex
71         \expandafter\newwrite\csname tf@#2\endcsname
72       \else
73         \immediate\closeout\@nameuse{tf@#2}%
74       \fi
75       \immediate\openout\@nameuse{tf@#2}\jobname.#3 %
76       \PackageInfo{index}{Writing index file \jobname.#3}%
77     }
78     {\PackageInfo{index}{Index ‘#2’ disabled -- not opening
79       \jobname.#3}}%
80   \fi
81   \expandafter\csname idxtitle@#2\endcsname
82 }

```

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

```

\@third 83 \def\@first#1:#2:#3\@nil{#1}
84
85 \def\@second#1:#2:#3\@nil{#2}
86
87 \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.

```
88 \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.

```

89 \edef\makeindex{%
90   \noexpand\newindex{default}{idx}{ind}{\indexname}%
91 }

```

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

```

92 \newif\if@silentindex\@silentindextrue
93
94 \newif\if@addtoindex\@addtoindextrue
95
96 \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 L<sup>A</sup>T<sub>E</sub>X's commands for dealing with tables of contents and  
page headings. (See below.) \*sigh\*

```

97 \def\index{\protect\p@index}
98
99 \def\p@index{%
100   \if@silentindex\@bsphack\fi
101   \@ifstar{\@silentindexfalse\@xindex}{\@silentindextrue\@xindex}%
102 }
103
104 \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]...`

```

105 \def\@index[#1]{%
106   \ifx\index\@gobble
107     \@addtoindexfalse
108   \fi
109   \def\@tempf{%
110     \begingroup
111       \@sanitize
112       \@index{#1}%
113   }%
114   \if@addtoindex
115     \@ifundefined{idx@#1}%
116     {%
117       \def\@tempf{%
118         \@latexerr{Index type ‘\string#1’ undefined}%
119         \@ehc
120         \@silentindextrue
121         \@gobble

```



```

122         }%
123     }%
124     {}%
125     \fi
126     \@tempf
127 }
128
129 \def\@@index#1#2{%
130     \endgroup
131     \if@addtoindex
132         \if@files\@wrindex{#1}{#2}\fi
133         \ifproofmode\@showidx{#2}\fi
134     \fi
135     \if@silentindex
136         \expandafter\@esphack
137     \else
138         \@silentindextrue#2%
139     \fi
140 }
141
142 \def\@wrindex#1#2{%
143     \begingroup
144         \def\@tempa{#2}%
145         \edef\@tempb{\@nameuse{idx@#1}}%
146         \edef\@tempb{\expandafter\@third\@tempb\@nil}%
147         \csname if@immediate@#1\endcsname \else
148             \expandafter\let\csname\@tempb\endcsname\relax
149         \fi
150         \edef\@tempa{%
151             \write\@auxout{%
152                 \string\@writefile{#1}{%
153                     \string\indexentry{\@nearverbatim\@tempa}%
154                     {\@nameuse{\@tempb}}%
155                 }%
156             }%
157         }%
158     \expandafter\endgroup\@tempa
159     \if@nobeak\ifvmode\nobreak\fi\fi
160 }

```

```

\seename The following are adapted from makeidx.sty, v2.09 (21 Oct 91). \index@prologue
\see adapted from doc.dtx. theindex based on version from classes.dtx, v1.3g, 26
\printindex June 1995.
\@printindex 161 \providecommand{\seename}{\see}
162
163 \providecommand*{\see}[2]{\emph{\seename} #1}
164
165 \@ifclassloaded{article}{%
166
167     \renewenvironment{theindex}{%
168         \edef\indexname{\the\@nameuse{idxtitle@\@indextype}}%
169         \if@twocolumn
170             \@restonecolfalse
171         \else

```

```

172         \@restonecoltrue
173     \fi
174     \columnseprule \z@
175     \columnsep 35\p@
176     \twocolumn[%
177         \section*{\indexname}%
178         \ifx\index@prologue\@empty\else
179             \index@prologue
180             \bigskip
181         \fi
182     ]%
183     \@mkboth{\MakeUppercase\indexname}%
184             {\MakeUppercase\indexname}%
185     \thispagestyle{plain}%
186     \parindent\z@
187     \parskip\z@ \@plus .3\p@\relax
188     \let\item\@idxitem
189 }-%
190 \if@restonecol
191     \onecolumn
192 \else
193     \clearpage
194 \fi
195 }
196 }-%
197 \renewenvironment{theindex}{%
198     \edef\indexname{\the\@nameuse{idxtitle@\@indextype}}%
199     \if@twocolumn
200         \@restonecolfalse
201     \else
202         \@restonecoltrue
203     \fi
204     \columnseprule \z@
205     \columnsep 35\p@
206     \twocolumn[%
207         \@makeschapterhead{\indexname}%
208         \ifx\index@prologue\@empty\else
209             \index@prologue
210             \bigskip
211         \fi
212     ]%
213     \@mkboth{\MakeUppercase\indexname}%
214             {\MakeUppercase\indexname}%
215     \thispagestyle{plain}%
216     \parindent\z@
217     \parskip\z@ \@plus .3\p@\relax
218     \let\item\@idxitem
219 }-%
220 \if@restonecol
221     \onecolumn
222 \else
223     \clearpage
224 \fi
225 }

```

```

226 }
227
228 \def\printindex{\@ifnextchar[{\@printindex}{\@printindex[default]}}
229
230 \def\@printindex[#1]{%
231   \@ifnextchar[{\@print@index[#1]}{\@print@index[#1] []}]%
232 }
233
234 \long\def\@print@index[#1] [#2]{%
235   \def\@indextype{#1}%
236   \long\def\index@prologue{#2}%
237   \@ifundefined{idx@#1}%
238     {\@latexerr{Index type ‘\string#1’ undefined}\@ehc}%
239     {%
240       \edef\@tempa{\@nameuse{idx@#1}}%
241       \edef\@tempb{%
242         \jobname.\expandafter\@second\@tempa\@nil
243       }%
244       \edef\@tempc{%
245         \jobname.\expandafter\@first\@tempa\@nil
246       }%
247       \InputIfFileExists{\@tempb}{}%
248         \typeout{No file \@tempb: Run makeindex -o
249           \@tempb \space \@tempc}%
250       }%
251     }%
252 }

```

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

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

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

`\idx@activebar`

```

254 \def\idx@activehat{%
255   \relax
256   \ifmmode\expandafter\sp\else\expandafter\@indexstar@\fi
257 }
258
259 \def\idx@activebar{%
260   \relax
261   \ifmmode\expandafter\sb\else\expandafter\index\fi
262 }

```

`\shortindexingon` 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.

<sup>6</sup>**Warning:** This feature is deprecated and will be removed entirely in a future release of this package.

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

```

263 \newif\if@shortindexing
264
265 \begingroup
266
267   \catcode'\^ \active
268   \catcode'\_ \active
269
270   \gdef\shortindexingon{%
271     \@shortindexingtrue
272     \chardef\old@idxhatcode\catcode'\^ \relax
273     \chardef\old@idxbarcode\catcode'\_ \relax
274     \catcode'\^ \active
275     \catcode'\_ \active
276     \let\old@idxhat ^%
277     \let\old@idxbar _%
278     \let^ \idx@activehat
279     \let\_ \idx@activebar
280   }
281
282   \gdef\shortindexingoff{%
283     \if@shortindexing
284       \@shortindexingfalse
285       \let^ \old@idxhat
286       \let\_ \old@idxbar
287       \catcode'\^ \old@idxhatcode
288       \catcode'\_ \old@idxbarcode
289     \fi
290   }
291
292 \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).

```

293 \newinsert\@indexbox
294
295 \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.

```

296 \begingroup
297   \catcode'\@active
298   \expandafter\gdef\csname\string @sanitizeat\endcsname
299     {\def @{\char'\@}}
300 \endgroup

```

`\indexproofstyle`

```

\@showidx 301 \newtoks\indexproofstyle
\@leftidx 302
\@rightidx 303 \indexproofstyle{\footnotesize\reset@font\ttfamily}
\@mkidx 304
\raggedbottom 305 \def\@showidx#1{%
\flushbottom 306   \insert\@indexbox{%
\@texttop 307     \@sanitizeat
308     \the\indexproofstyle
309     \hsize\marginparwidth
310     \hangindent\marginparsep \parindent\z@
311     \everypar{}\let\par\@par \parfillskip\@flushglue
312     \lineskip\normallineskip
313     \baselineskip .8\normalbaselineskip\sloppy
314     \raggedright \leavevmode
315     \vrule \@height .7\normalbaselineskip \@width \z@\relax#1\relax
316     \vrule \@height\z@ \@depth.3\normalbaselineskip \@width\z@\relax
317   }%
318   \ifhmode\penalty\@M \hskip\z@skip\fi
319 }
320
321 \def\@leftidx{\hskip-\marginparsep \hskip-\marginparwidth}
322
323 \def\@rightidx{\hskip\columnwidth \hskip\marginparsep}
324
325 \def\@mkidx{%
326   \vbox to \z@{%
327     \rlap{%
328       \if@twocolumn
329         \if@firstcolumn \@leftidx \else \@rightidx \fi
330       \else
331         \if@twoside
332           \ifodd\c@page \@rightidx \else \@leftidx \fi
333         \else
334           \@rightidx
335         \fi
336       \fi
337     \box\@indexbox
338   }%
339   \vss
340 }%
341 }
342
343 \def\raggedbottom{%
344   \def\@textbottom{\vskip\z@ plus.0001fil}%

```

```

345 \let\@texttop\@mkidx
346 }
347
348 \def\flushbottom{\let\@textbottom\relax \let\@texttop\@mkidx}
349
350 \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 L<sup>A</sup>T<sub>E</sub>X'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.

```

351 \CheckCommand\addtocontents[2]{%
352   \protected@write\@auxout
353     {\let\label\@gobble \let\index\@gobble \let\glossary\@gobble}%
354     {\string\@writefile{#1}{#2}}}%
355 }
356
357 \renewcommand{\addtocontents}[2]{%
358   \protected@write\@auxout
359     {\let\label\@gobble \let\glossary\@gobble}%
360     {\string\@writefile{#1}{#2}}}%
361 }

```

`\@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.

```

362 \let\old@starttoc\@starttoc
363
364 \renewcommand{\@starttoc}[1]{%
365   \begingroup
366     \@addtoindexfalse
367     \old@starttoc{#1}%
368   \endgroup
369 }

```

`\markboth` Finally, we have to redefine `\markboth` and `\markright` to keep them from disabling the expansion of `\index` while putting section heads into the `\mark`. Otherwise, we'd end up with "`\index`" in the mark, which would cause problems when `\@outputpage` redefines `\index` to be equal to `\@gobble`. Instead, we want `\index` to expand to `\p@index` in the `\mark`, so we retain control over what happens in `\@outputpage`.

This time, the only change is to remove `\index` from the list of macros that are `\let` equal to `\relax`.

`\markboth` and `\markright` are now defined using `\DeclareRobustCommand`, which means that before we can use `\CheckCommand`, we have to break open `\markboth` and `\markright` to get to the chewy inner centers, `\markboth_` and `\markright_`.

```

370 \begingroup
371
372 \def\@tempa#1{%
373   \begingroup
374     \edef\@tempa{%
375       \let \expandafter\noexpand\csname#1\endcsname
376         \expandafter\noexpand\csname#1 \endcsname
377     }%
378   \expandafter
379 \endgroup
380 \@tempa
381 }
382
383 \@tempa{markboth}
384 \@tempa{markright}
385
386 \ExplSyntaxOn
387
388 \CheckCommand*\markboth[2]{%
389   \begingroup
390     \let\label\relax \let\index\relax \let\glossary\relax
391     \unrestored@protected@xdef\@themark {#1}{#2}%
392     \@temptokena \expandafter{\@themark}%
393     \mark_insert:nn{2e-left}{#1}
394     \mark_insert:nn{2e-right}{#2}
395     \tl_if_empty:nF{#2}{ \mark_insert:nn{2e-right-nonempty}{#2} }
396     \mark{\the\@temptokena}%
397   \endgroup
398   \if@nobreak\ifvmode\nobreak\fi\fi}
399
400 \CheckCommand*\markright[1]{%
401   \begingroup
402     \let\label\relax \let\index\relax \let\glossary\relax
403     \expandafter\@markright\@themark {#1}%
404     \@temptokena \expandafter{\@themark}%
405     \mark_insert:nn{2e-right}{#1}
406     \tl_if_empty:nF{#1}{ \mark_insert:nn{2e-right-nonempty}{#1} }
407     \mark{\the\@temptokena}%
408   \endgroup
409   \if@nobreak\ifvmode\nobreak\fi\fi}
410
411 \ExplSyntaxOff
412
413 \endgroup
414
415 \ExplSyntaxOn
416
417 \DeclareRobustCommand*\markboth[2]{%
418   \begingroup
419     \let\label\relax \let\glossary\relax

```

```

420 \unrestored@protected@xdef\@themark {#1}{#2}%
421 \@temptokena \expandafter{\@themark}%
422 \mark_insert:nn{2e-left}{#1}
423 \mark_insert:nn{2e-right}{#2}
424 \tl_if_empty:nF{#2}{ \mark_insert:nn{2e-right-nonempty}{#2} }
425 \mark{\the\@temptokena}%
426 \endgroup
427 \if@nobreak\ifvmode\nobreak\fi\fi}
428
429 \DeclareRobustCommand*\markright[1]{%
430 \begingroup
431 \let\label\relax \let\glossary\relax
432 \expandafter\@markright\@themark {#1}%
433 \@temptokena \expandafter{\@themark}%
434 \mark_insert:nn{2e-right}{#1}
435 \tl_if_empty:nF{#1}{ \mark_insert:nn{2e-right-nonempty}{#1} }
436 \mark{\the\@temptokena}%
437 \endgroup
438 \if@nobreak\ifvmode\nobreak\fi\fi}
439
440 \ExplSyntaxOff
441 \endstyle

```

## 7 Finale

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

```
442 \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 `\shortindexing` and `\shortindexingoff` 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 `\shortindexing` command. (Question: Should `\shortindexing` 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  $\LaTeX 2_{\epsilon}$  package. Fixed `\@printindex` to work under  $\LaTeX 2_{\epsilon}$  (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 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> package (finally!). Changes too numerous to list, but in general deleted some now-superfluous code, replaced some tricks by tricks from the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 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.

<b>Symbols</b>	<code>\disableindex</code> . . . . . <i>3, 5</i>	<b>N</b>
<code>\@@index</code> . . . . . <u>105</u>		<code>\newindex</code> . . . . . <i>2, <u>19</u></i>
<code>\@first</code> . . . . . <u>83</u>	<b>E</b>	
<code>\@index</code> . . . . . <u>105</u>	environments:	<b>P</b>
<code>\@indexbox</code> . . . . . <u>293</u>	<code>shortindexingon</code> . . . . . <i>3</i>	<code>\p@index</code> . . . . . <u>97</u>
<code>\@indexstar@</code> . . . . . <u>253</u>		<code>\printindex</code> . . . . . <i>3, <u>161</u></i>
<code>\@leftidx</code> . . . . . <u>301</u>	<b>F</b>	<code>\proofmodefalse</code> . . . . . <i>3</i>
<code>\@mkidx</code> . . . . . <u>301</u>	<code>\flushbottom</code> . . . . . <u>301</u>	<code>\proofmodetrue</code> . . . . . <i>3</i>
<code>\@nearverbatim</code> . . . . . <u>88</u>		
<code>\@preamblecmds</code> . . . . . <u>59</u>	<b>I</b>	<b>R</b>
<code>\@printindex</code> . . . . . <u>161</u>	<code>\idx@activebar</code> . . . . . <u>254</u>	<code>\raggedbottom</code> . . . . . <u>301</u>
<code>\@rightidx</code> . . . . . <u>301</u>	<code>\idx@activehat</code> . . . . . <u>254</u>	<code>\renewindex</code> . . . . . <i>3, <u>19</u></i>
<code>\@sanitizeat</code> . . . . . <u>296</u>	<code>\if@addtoindex</code> . . . . . <u>92</u>	
<code>\@second</code> . . . . . <u>83</u>	<code>\if@newindex</code> . . . . . <u>19</u>	<b>S</b>
<code>\@showidx</code> . . . . . <u>301</u>	<code>\if@proofmode</code> . . . . . <u>92</u>	<code>\see</code> . . . . . <u>161</u>
<code>\@starttoc</code> . . . . . <u>362</u>	<code>\if@silentindex</code> . . . . . <u>92</u>	<code>\seename</code> . . . . . <u>161</u>
<code>\@texttop</code> . . . . . <u>301</u>	<code>\index</code> . . . . . <i>3, <u>97</u></i>	<code>\shortindexingoff</code> <i>3, <u>263</u></i>
<code>\@third</code> . . . . . <u>83</u>	<code>\indexproofstyle</code> <i>3, <u>301</u></i>	<code>\shortindexingon</code> <i>3, <u>263</u></i>
<code>\@wrindex</code> . . . . . <u>105</u>		<code>shortindexingon</code> (env.) . . . . . <i>3</i>
<b>A</b>	<b>M</b>	
<code>\addcontentsline</code> . . . . . <u>351</u>	<code>\makeindex</code> . . . . . <i>3, <u>89</u></i>	
<b>D</b>	<code>\markboth</code> . . . . . <u>370</u>	<b>X</b>
<code>\def@index</code> . . . . . <u>62</u>	<code>\markright</code> . . . . . <u>370</u>	<code>\x@index</code> . . . . . <u>97</u>