

The aeb-minitoc Package

D. P. Story

Email: `dpstory@uakron.edu`

processed September 23, 2018

Contents

1	Description	1
2	Documentation and Code.	2
3	Index	13
4	Change History	15

1 `\package`

`\ifMiniTocListings` The `\ifMiniTocListings` is a Boolean switch, which when true signals that there is a non-empty listing; otherwise, it is set to false. It is used to display a latex warning to the user that the listing is empty. Also, globally, `\ifMiniTocListings` is set to false when `\noinitocs` is taken. The other option is `!noinitocs` is a convenience option; it is not ‘not’ version of `\noinitocs`; when `!noinitocs` is specified, mini-tocs are created. This is the same as specifying no option at all.

`\noinitocs`
`\EXCL!noinitocs`

```
2 \newif\ifMiniTocListings \MiniTocListingstrue
3 \DeclareOption{noinitocs}{\AtEndOfPackage{\MiniTocListingsfalse}
4 \let\insertminitoc\insertminitocNOT}}
5 \DeclareOption{!noinitocs}{\MiniTocListingstrue}
6 \ProcessOptions
```

1 Description

A simple mini-toc package; originally designed for `web`, but now works for all standard `LATEX` classes. The main user command is `\insertminitoc`, defined below.

Our approach is to use each entry the `\jobname.toc` as the first argument of the macro `\mtocCL`, a second argument keeps a running count on the number of entries.

```
\mtocCL{\contentsline{section}{\numberline{1}Section Title}{2}}{cnt} or
\mtocCL{\contentsline{section}
{\numberline{1}Section Title}{2}{section.1}}{cnt}
```

`\contentsline` has four arguments when `hyperref` is loaded and three otherwise. When inserting the full table of contents, we define `\def\mtocCL#1#2{#1}` to do nothing. When we are building a mini-toc, we `\let\mtocCL\mtoc@CL@mtoc`. The effect of this macro is to remove any entry (in `\jobname.toc`) that does not contain `\contentsline` as its first token and to position the `cnt` argument for later use. But by then `\contentsline` has already been `\let` to `\cl@LOOKFORSEC`. Now `\cl@LOOKFORSEC` determines whether any particular entry should be displayed in the current mini-toc.

2 Documentation and Code.

As a demonstration of this package, we present a mini-toc for this section, which only has `\paragraph` and `\subparagraph` section headings.

The verbatim listing for this mini-toc is

```
\TOCLevels{section}{subparagraph}
\begin{minitocfmt}{\minitocFmt}
  \declaretocfmt{paragraph}{\@W{1em}\@D{0em}}
  \declaretocfmt{subparagraph}{\@W{1.5em}\@D{1em}}
\end{minitocfmt}

\begin{center}\minitocFmt
\fbbox{\begin{minipage}{0.8\linewidth}\centering
\textbf{Contents of this section}\vadjust{\kern3pt}}%
\insertminitoc\relax
\end{minipage}}
\end{center}
```

Contents of this section	
¶ The top and bottom most	3
¶¶ Manually set the top and bottom levels	3
¶¶ Automatically set the top and bottom levels	3
¶ Modify <code>\tableofcontents</code>	5
¶ Modify <code>\addtocontents</code>	5
¶ Modify <code>\@startsection</code> and referencing	6
¶ <code>\insertminitoc</code> : The main command	6
¶ The mini-toc format environment: <code>minitocfmt</code>	8
¶¶ The <code>\declaretocfmt</code> command defined	10
¶¶ The <code>minitocfmt</code> environment defined	11

We begin by saving the definitions macros we modify later.

```
7 \let\mtoc@contentsline\contentsline
8 \let\mtoc@starttoc\@starttoc
9 \let\mtoc@tableofcontents\tableofcontents
10 \let\mtoc@addtocontents\addtocontents
```

`\@minitocCnt` Some counters and utility macros. The counter `\@minitocCnt` is incremented

`\mtocgobble` in the redefined `\addtocontents` command. The command `\mtocgobble` is a ‘public’ version of the core L^AT_EX command `\@gobble`.

```
11 \newcount\@minitocCnt \@minitocCnt=0\relax
12 \def\csarg#1#2{\expandafter#1\csname#2\endcsname}
13 \let\mtoc@One=1 \let\mtoc@Zero=0
14 \let\mtocgobble\@gobble
```

¶¶ **The top and bottom most.** The package assigns the top level and bottom level automatically, based upon the class being used; the document author can override these for the whole document, or for particular mini-tocs.

`\TOPLevel` `\TOPLevel{<name>}` is the name of the top level. It is expected that a mini-toc will be inserted with each top level in the document, as the author’s discretion.

`\BTMLevel` The `\BTMLevel{<name>}` is the name of the bottom most level. A mini-toc consists of all sections *beneath* the top level and *above* the bottom level. Thus, if `\TOPLevel{chapter}` and `\BTMLevel{subsubsection}`, then the mini-toc contains all `\section` and `\subsection` title headings within the current chapter.

`\TOPLevel` ¶¶ **Manually set the top and bottom levels.** `\TOPLevel{<top-level>}`
`\BTMLevel` and `\BTMLevel{<btm-level>}` are used to determine what entries are to be included in the mini-toc. As a convenience,

`\TOCLevels` `\TOCLevels{<top-level>}{<btm-level>}`

can declare both at once. If an argument is empty, the current level is used.

```
15 \def\TOPLevel#1{\def\x{#1}\ifx\x\@empty\else
16 \def\mtoc@TOPLevel{#1}\edef\TOPLevelNum{\@nameuse{sl@#1}}\fi}
17 \def\BTMLevel#1{\def\x{#1}\ifx\x\@empty\else
18 \def\mtoc@BTMLevel{#1}\edef\BTMLevelNum{\@nameuse{sl@#1}}\fi}
19 \def\TOCLevels#1#2{\TOPLevel{#1}\BTMLevel{#2}}
```

¶¶ **Automatically set the top and bottom levels.** We make reasonable choices for book, report, and article; these are the three classes that this package supports. In the course, we define, in macro form, the levels of each of these section names (`\sl@<sec-name>` and `\sl@<sec-name>*`).

```
20 \@ifclassloaded{book}{%
21 \TOCLevels{chapter}{subsection}
22 \def\sl@part{-1}\def\sl@chapter{0}
23 \csarg{\edef}{sl@part*}{\sl@part}
24 \csarg{\edef}{sl@chapter*}{\sl@chapter}
25 }{%
26 \@ifclassloaded{report}{%
27 \TOCLevels{chapter}{subsection}
28 \def\sl@part{-1}\def\sl@chapter{0}
29 \csarg{\edef}{sl@part*}{\sl@part}
30 \csarg{\edef}{sl@chapter*}{\sl@chapter}
31 }{%
32 \TOCLevels{section}{subsubsection}
33 \def\sl@part{0}\csarg{\edef}{sl@part*}{\sl@part}
34 }%
```

```

35 }
36 \def\sl@section{1}\def\sl@subsection{2}\def\sl@subsubsection{3}
37 \def\sl@paragraph{4}\def\sl@subparagraph{5}%\def\sl@all{17}
38 \csarg{\edef}{\sl@section*}{\sl@section}
39 \csarg{\edef}{\sl@subsection*}{\sl@subsection}
40 \csarg{\edef}{\sl@subsubsection*}{\sl@subsubsection}
41 \csarg{\edef}{\sl@paragraph*}{\sl@paragraph}
42 \csarg{\edef}{\sl@subparagraph*}{\sl@subparagraph}
43 \newif\if@foundTOPLLevel \@foundTOPLLevelfalse

```

The command `\insertminitoc`, just before inputting `\jobname.toc`, `\lets`
`\c1@LOOKFORSEC` `\contentsline` to `\c1@LOOKFORSEC`. This command then looks for lines at the

top most section level, if it finds one, and the section number matches the one set by `\insertminitoc` (`\mtoc@sec`), it sets `\if@foundTOPLLevel` to `true`, and stores all subsequent lines in `\toks@` until another section is encountered, at which time `\if@foundTOPLLevel` is set to `false`. There are two versions of

`\c1@LOOKFORSEC@LX` `\c1@LOOKFORSEC`: (1) `\c1@LOOKFORSEC@LX` for when `hyperref` is not loaded; and
`\c1@LOOKFORSEC@HY` (2) `\c1@LOOKFORSEC@HY` for when `hyperref` is loaded.

`\mtoc@@contentsline` takes five arguments, we save the page number (#3) the `hyperref` anchor (#4) and the TOC entry number (#5). The definitions made within `\mtoc@@contentsline` are later `\let` to `\@PgNum`, `\@L`, and `\@E`. We grab #5, which is the entry count, and pass the rest to `\mtoc@contentsline`.

```

44 \def\mtoc@@contentsline#1#2#3#4#5{\def\mtoc@PgNum{#3}%
45 \def\mtoc@HY@anchor{#4}\def\TOCEntryNum{#5}%
46 \mtoc@contentsline{#1}{#2}{#3}{#4}}

```

All but the last argument in both of these next two command are the standard arguments for `\contentsline`. The last argument is one introduced by this package; it keeps the count of the TOC entries. This last argument is used to identify the top level section.

```

47 \long\def\c1@LOOKFORSEC@LX#1#2#3#4{%
48 \c1@LOOKFORSEC@HY{#1}{#2}{#3}{\@empty}{#4}}
49 \long\def\c1@LOOKFORSEC@HY#1#2#3#4#5{\def\mt@rgi{#1}% dps
50 \if@foundTOPLLevel\ifx\mtocCL\@gobbletwo\else
51 \edef\NUMLevel{\@nameuse{\sl@#1}}%
52 \ifnum\NUMLevel > \TOPLLevelNum\relax\else
53 \let\mtocCL\@gobbletwo
54 \fi
55 \fi\fi
56 \@chkForNl#2\@nil % is it a * section
57 \ifx\mtocCL\@gobbletwo\else
58 \edef\mtoc@tmp{\the\mtocs@toks@}%
59 \ifx\mt@rgi\mtoc@TOPLLevel
60 \def\SECNUM{MTOC.#5}%
61 \set@display@protect
62 \edef\mtoc@sec{\mtoc@sec}%
63 \ifx\SECNUM\mtoc@sec
64 \@foundTOPLLeveltrue
65 \else

```

```

66         \@foundTOPLLevelfalse\fi
67         \set@typeset@protect
68     \else
69         \if@foundTOPLLevel
70         \ifnum\NUMLevel > \BTMLevelNum\relax\else
71         \ifx\@nlrtn\relax
72         \mtocs@toks@=\expandafter{\mtoc@tmp
73         \mtoc@@contentsline{#1}{#2}{#3}{#4}{#5}}\else
74         \mtocs@toks@=\expandafter{\mtoc@tmp
75         \mtoc@@contentsline{#1*}{#2}{#3}{#4}{#5}}\fi
76     \fi
77 \fi
78 \fi
79 \fi
80 }

```

This version of \mtoc@BTMLevel only accepts lines that are not subsubsection.

```

81 \@ifpackageloaded{hyperref}{\let\c1@LOOKFORSEC\c1@LOOKFORSEC@HY}
82 {\let\c1@LOOKFORSEC\c1@LOOKFORSEC@LX}

```

\@chkForNl determines if the first token is \numberline.

```

83 \def\@chkForNl#1#2\@nil{% check for number line
84 \ifx#1\numberline\let\@nlrtn\relax\else
85 \def\@nlrtn{\numberline{\hfill}}\fi}

```

¶ Modify \tableofcontents

```

86 \def\mtoc@st@rttoc#1{\begingroup
87 \if@filesw \expandafter\newwrite\csname tf@#1\endcsname
88 \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
89 \fi\global\@nbreakfalse\endgroup}
90 \let\mtoc@@starttoc\mtoc@One
91 \def\mtoc@start@toc{\let\mtoc@start@next\relax
92 \@ifundefined{aebLastPage}
93 {%
94 \ifx\mtoc@@starttoc\mtoc@One
95 \global\let\mtoc@@starttoc\mtoc@Zero
96 \def\mtoc@start@next{\mtoc@st@rttoc{toc}}\fi
97 \mtoc@start@next
98 }{%
99 \ifnum\aeblastPage<\thepage\relax
100 \def\mtoc@start@next{\mtoc@st@rttoc{toc}}\fi
101 \mtoc@start@next
102 }%
103 }
104 \AtEndDocument{\mtoc@start@toc}

```

¶ **Modify \addtocontents.** (\addtocontents{toc}{<content>}) If the document author inserts vertical spacing, or other formatting, that could be problems in the minitoc. So we'll try to remove it. We begin by placing the second argument <content> as the argument of a command, \mtocCL{<content>}. Initially,

`\mtocCL` just passes its argument into the \TeX stream. Later, it will be redefined within `\insertminitoc`.

```
105 \def\mtoc@TOC{toc}
106 \def\mtocCL#1#2{#1}
```

Here, we modify the macro `\addtocontents` to insert `\mtocCL`.

```
107 \renewcommand\addtocontents[2]{%
108   \def\mt@rgi{#1}\ifx\mt@rgi\mtoc@TOC
109   \global\advance\@minitocCnt\@ne
110   \mtoc@addtocontents{#1}{\protect\mtocCL{#2}{\the\@minitocCnt}}\else
111   \mtoc@addtocontents{#1}{#2}\fi}
```

Modify the `\tableofcontents` to `\mtoc@tableofcontents`. We compensate later by executing `\mtoc@start@toc` at the end of the document.

```
112 \def\tableofcontents{%
113   \def\@starttoc##1{\makeatletter
114     \@input{\jobname.##1}\makeatother}%
115   \NoFmtOCEntry
116   \mtoc@tableofcontents
117   \global\let\@starttoc\mtoc@starttoc}
```

`\mtoc@CL@mtoc` `\mtoc@CL@mtoc` is the redefined version of `\mtocCL`, as described above. We attempt to see if the first token of its argument is `\contentsline`, if yes we pass it on, otherwise, we gobble it.

```
118 \newcommand{\mtoc@CL@mtoc}[1]{\mtoc@parse#1\@nil}
119 \def\mtoc@parse#1#2\@nil{\ifx#1\contentsline
120   \def\mtoc@next##1{#1#2{##1}}\else
121   \let\mtoc@next\@gobble\fi\mtoc@next}
```

¶ **Modify `\@startsection` and referencing.** We redefine `\@startsection` to pick up the first argument (the section name) and define `\@currentsecname`, which is use in a simple cross referencing system needed for this mini-toc package. This package should be loaded after `hyperref` for sure.

```
122 \let\@startsection@mtoc@SAVE\@startsection
123 \def\@startsection#1{\def\@currentsecname{#1}%
124   \@startsection@mtoc@SAVE{#1}}
```

`\mtoclabel` `\mtocref` The use of `\mtoclabel` and `\mtocref` are not needed unless you redefine a section heading to a non-numerical value. This system needs a section number.

```
125 \def\mtoclabel#1{\label{#1}\@bsphack
126   \protected@write\@auxout{}{\string
127     \csarg{\string\gdef}{mtoclbl#1}{\the\@minitocCnt}
128   }}%
129   \@esphack
130 }
131 \def\mtocref#1{\@nameuse{mtoclbl#1}}
```

`\insertminitoc` ¶ **`\insertminitoc`: The main command.** `\insertminitoc` is the main user command for this package, it places a “minitoc” for a section (`\mtoc@TOPLevel`) of a document, listing only the subsections within that section. It takes an optional

argument for indicating the section number, the subsections of which are to be displayed. The default is the current section, `\@nameuse{the\mtoc@TOPLevel}`.

`\if@minitoc` This Boolean is set to true, in a group, when `\insertminitoc` is expanded. This is to support a feature for formatting a mini-toc entry; `\miniorfulltoc` is used for this purpose. `\miniorfulltoc` is inserted in the optional argument of a section command:

```
\subsection[\protect
\miniorfulltoc{\textbf}{Subsection Entry}]{Subsection Entry}
```

`\miniorfulltoc{<fmt>}{<entry>}` The first argument of `\miniorfulltoc` is passed to the second entry; for example, `{\textbf{Subsection Entry}}` as an argument and in a group. Thus, the first argument can be a command with one argument, or a command with not arguments.

```
132 \newif\if@minitoc \@minitocfalse
133 \newif\if@MiniTocListings \@MiniTocListingstrue
134 \def\NoFmtTOCEntry{\@minitocfalse}
135 \def\FmtTOCEntry{\@minitoctrue}
136 \def\miniorfulltoc#1#2{\if@minitoc
137   {\#1{\#2}}\else{\#2}\fi}
```

`\insertminitoc[<label-name>]` After the above preliminaries, we get to `\insertminitoc`. The default value of the optional parameter is `MTOC.\the\@minitocCnt`; thus, we use the most recent value of `\@minitocCnt`. An explicit argument is needed when the mini-toc is placed somewhere else (after `\minitocCnt` has been incremented). You can also say `\insertminitoc[<label-name>]`, where `<label-name>` is a label name set by the `\mtoclabel` command.

```
138 \newcommand{\insertminitoc}[1][]{%
139   \def\mtoc@rgi{#1}\ifx\mtoc@rgi\empty
140     \edef\mtoc@rgi{MTOC.\the\@minitocCnt}\else
141     \edef\mtoc@rgi{MTOC.\mtocref{#1}}\fi
142   \ifnum\TOPLevelNum > \BTMLevelNum
143     \PackageError{aeb-minitoc}{%
144       The top level (\mtoc@TOPLevel) must be\MessageBreak
145       must be higher on the hierarchy then at bottom level}
146     {Try switching the two}\fi
147   \begingroup
148   \edef\mtoc@sec{\mtoc@rgi}\mtocs@toks@={}%
149   \let\contentsline to \cl@LOOKFORSEC
150   \let\mtocCL to \mtoc@CL@mtoc
151   \let\mtocCL\mtoc@CL@mtoc
152   \let\mtoc@numberline\numberline
153   \let\mtoc@numberline\mtoc@numberline
154   \let\mtoc@numberline\mtoc@numberline
```

Insert formatting (`\Pg = \sl@@sNumFmt`) for the page number here.

```
153 \def\numberline##1{\makebox[\mtoc@numBoxWidth][l]{%
154   {\sl@@sNumFmt{##1}}\sl@@EntryFmt}%
155 }
```

```

155 \makeatletter\inputIfFileExists{\jobname.toc}%
156 {\PackageInfo{aeb-minitoc}{TOC file read}}
157 {\PackageInfo{aeb-minitoc}{TOC file not available}}%
158 \edef\x{\the\mtocs@toks@}\ifx\x\@empty
159   \global\@MiniTocListingsfalse\else
160   \global\@MiniTocListingstrue\fi

```

Insertion point. This is where the mini-toc entries are entered into the latex stream to be typeset.

```

161 \the\mtocs@toks@\par\makeatother
162 \if@MiniTocListings\else
163   \PackageWarning{aeb-mintoc}{No mini-toc built here}\fi
164 \endgroup
165 }

```

When the `nominitocs` option is in effect, we `\let` the command `\insertminitoc` to `\insertminitocNOT`, which absorbs all its arguments.

```

166 \newcommand{\insertminitocNOT}[1] [] {}

```

`\numBoxWidth{<length>}` The `\mtoc@numBoxWidth` determines the width of the `\hbox` that contains the section number. It is conveniently set using `\numBoxWidth`. The default declaration is `\numBoxWidth{2.5em}`. The `<length>` should be measured in em units. Within the `minitocfmt`, `\@W` is `\let` to `\numBoxWidth`.

```

167 \def\numBoxWidth#1{\def\mtoc@numBoxWidth{#1}}
168 \numBoxWidth{2.5em}

```

In its “raw” expansion, `\insertminitoc` may not be what you want; in this case, enclose it in some appropriate environment. The following is an example of how to use this command. This can be part of a command that inserts code just after every `\section`.

```

\begin{center}\minitocFmt
\begin{tabular}{c}\toprule
\begin{minipage}[c]{0.8\linewidth}
\insertminitoc\relax
\end{minipage}\\bottomrule
\end{tabular}
\end{center}

```

where `\minitocFmt` is a command that expands to some formatting, see demo files.

¶ **The mini-toc format environment: `minitocfmt`.** To help facilitate designing and declaring the mini-toc format, we define the `minitocfmt` environment. The environment defines a command (`<cmd>`) that contains all the formatting information for the mini-toc. The body of the environment consists of a series of `\declaretocfmt{<toc-fmt>}` declarations. Within argument of `\declaretocfmt`, `\@W` is an alias for `\numBoxWidth` and `\@D` is an alias for `\sl@dots`. If `\@D` appears (`\@D = @dottedtocline`, a dotted line is created in the usual L^AT_EX manner. `\@N` is an alias `\sl@sNumFmt` and `\@P` is an alias for `\sl@pNumFmt`. All are optional.

	<code>\@A{<i>various</i>}</code> is a command that is not used very often, but is available when needed. The argument <i>various</i> is various commands to support the min-toc being generated.
<i>pg num box width</i>	<code>\@PW{<i>length</i>}</code> Within the argument of <code>\@A</code> , insert <code>\@PW{<i>length</i>}</code> to set the width of the box that contains the page number (<code>\@pnumwidth</code>). The value set by L ^A T _E X is 1.55em.
<i>dots separation</i>	<code>\@DS{<i>num</i>}</code> The <i>num</i> determines the separation between dots for a TOC entry that uses a dotted rule line. This command is only recognized within the argument of <code>\@A</code> . The default is 4.5.
<i>right margin of title</i>	<code>\@R{<i>length</i>}</code> is a convenience command, it takes its argument and defines the L ^A T _E X command <code>\@tocrmarg</code> , which sets the right margin for the sec-title. The length set by L ^A T _E X is 2.55em. The <i>length</i> of <code>\@R</code> should be <i>larger than</i> the <i>length</i> set by <code>\@PW</code> .
<code>\declaretocfmt{<i>sec-name</i>}{<i>various</i>}</code> formats all <i>sec-name</i> (section, subsection, etc.) entries.	
A ‘typical’ table of contents entry has the form:	
<i>sec-num</i> <i>title-heading</i> <i>pg-num</i>	
Within the <i>various</i> argument, there are a number of commands that are recognized:	
<i>sec num box width</i>	<code>\@W{<i>length</i>}</code> is the width of the box that encloses <i>sec-num</i> . Normally, all lengths are measured in em units (<code>\@W{<i>num</i>}em</code>). The default length is 2.5em
<i>use dots</i>	<code>\@D{<i>length</i>}</code> is the amount to indent prior to <i>sec-num</i> . Again, em units preferred (<code>\@D{<i>num</i>}em</code>). When the <code>\@D</code> command is present in the argument, a dotted line is to be used for the entry (this is the norm). If <code>\@D</code> not present, there is an opportunity within the <i>various</i> argument to create a custom entry.
<i>no dots</i>	<code>\@B{<i>length</i>}</code> Same as <code>\@D</code> , but no dotted leaders are created.
<i>fmt sec num</i>	<code>\@N{<i>fmt</i>}</code> is the formatting for <i>sec-num</i> . You can pass a command with one argument that will operate on the section number; for example, <code>\@N{\textbf}</code> , <code>\@N{\color{blue}}</code> , or <code>\@N{\color{blue}\textbf}</code> . Note that changing the style to bold might require a corresponding change in <code>\@W</code> .
<i>fmt title</i>	<code>\@F{<i>fmt</i>}</code> is the formatting for the title heading of the current section; for example, <code>\@F{\bfseries}</code> turns all heading, for this <i>sec-name</i> , bold.
<i>fmt pg num</i>	<code>\@P{<i>fmt</i>}</code> is the formatting for the page number (<i>pg-num</i>). You can pass a command with one argument that will operate on the page number. When <code>hyperref</code> is loaded with the <code>colorlinks</code> option, we cannot change the color of the page number (see the discussion of <code>\@A</code> above), but <code>\@P{\textit}</code> changes the numbers to italics. If <code>hyperref</code> is not loaded, <code>\@P{\color{red}\textit}</code> changes page numbers to a red italic.

<i>right margin of title</i>	<code>\@R{<length>}</code> is a convenience command, it takes its argument and defines the <code>\@tocrmarg</code> command, which sets the right margin for the sec-title. The length set by <code>\@R</code> is <code>2.55em</code> . Setting <code>\@R</code> within the <code><various></code> argument of <code>\declaretocfmt</code> affects the current section level as well as all lower section levels. If you want to make this ‘local’ change, you need to put <code>\@R</code> back to its default of <code>2.55em</code> locally for other declarations.
<i>TOC number</i>	<code>\@E</code> Within the <code>\minitocfmt</code> environment, the command <code>\@E</code> expands to the current TOC entry number of the TOC entry being read in.
<i>link anchor</i>	<code>\@L</code> This macro expands to the <code>hyperref</code> anchor of the page entry reference, it is empty if <code>hyperref</code> is not loaded.
<i>pg number</i>	<code>\@Pg</code> This macro expands to the page number this entry references.

Usually, the `<length>` argument is measured in `em` units (`<num>em`).

```
169 \newtoks\mtoc@toks
170 \newtoks\mtocs@toks@
```

Within the `\minitocfmt`, `\@D` is `\let` to `\sl@dots`.

```
171 \def\sl@dots#1{\def\sl@@dots{%
172   \@dottedtocline{\sl@current}{#1}{\mtoc@numBoxWidth}}
173 \let\sl@@dots@gobble
174 \def\sl@nodots#1{\def\sl@@dots{%
175   \@nodottedtocline{\sl@current}{#1}{\mtoc@numBoxWidth}}
176 \let\sl@@nodots@gobble
```

Within the `\minitocfmt`, `\@F` is `\let` to `\@EntryFmt`.

```
177 \def\sl@EntryFmt#1{\def\sl@@EntryFmt{#1}} % dps
178 \let\sl@@EntryFmt\relax % dps
```

Within the `\minitocfmt`, `\@N` is `\let` to `\sl@sNumFmt`.

```
179 \def\sl@sNumFmt#1{\def\sl@@sNumFmt{#1}}
180 \let\sl@@sNumFmt\relax
```

Within the `\minitocfmt`, `\@P` is `\let` to `\sl@pNumFmt`.

```
181 \def\sl@pNumFmt#1{\def\sl@@pNumFmt{#1}}
182 \let\sl@@pNumFmt\relax
183 \def\sl@tocrmarg#1{\def\@tocrmarg{#1}}
184 \def\sl@dotsep#1{\def\@dotsep{#1}}
185 \def\mtoc@star#1*#2\@nil{\def\@rgii{#2}\ifx\@rgii\@empty
186   \let\mtoc@@star\mtoc@Zero\else\let\mtoc@@star\mtoc@One\fi}
```

¶¶ The `\declaretocfmt` command defined. The `\declaretocfmt` is used to designed how a mini-toc entry is displayed.

`\declaretocfmt{<sec-name>}{<various>}` The aeb-mintoc way of declaring the formatting for a toc `<sec-name>` entry. The `<various>` argument consists of various combinations of `\@W`, `\@D`, `\@N`, and `\@P`.

```
187 \long\def\declaretocfmt#1#2{%
188   \xdef\sl@current{\@nameuse{sl@#1}}%
```

```

189 \global\@namedef{\mtoc@CmdName @l@#1}##1##2{%
190   \normalfont\normalcolor\let\@E\TOCEntryNum
191   \let\@L\mtoc@HY@anchor\let\@Pg\mtoc@PgNum
192   \let\sl@dots\@empty\let\sl@sNumFmt\relax
193   \let\sl@pNumFmt\relax\let\sl@EntryFmt\relax
194   \let\@W\numBoxWidth\let\@R\sl@tocrmarg\let\@D\sl@dots
195   \let\@B\sl@nodots\let\@F\sl@EntryFmt\let\@N\sl@sNumFmt
196   \let\@P\sl@pNumFmt
197   #2\ifx\sl@dots\@empty\let\sl@next\relax\else
198     \mtoc@star#1*\@nil % dps
199     \ifx\mtoc@star\mtoc@Zero
200       \def\sl@next{\sl@dots{##1}{\sl@pNumFmt{##2}}}\else
201       \def\sl@next{\sl@dots{\sl@EntryFmt{##1}{\sl@pNumFmt{##2}}}\fi
202       \fi\sl@next}%
203   \edef\x{\expandafter\noexpand\csname l@#1\endcsname}%
204   \edef\y{\expandafter\noexpand\csname\mtoc@CmdName @l@#1\endcsname}%
205   \edef\mtoc@tmp{\the\mtoc@toks\let\expandafter\noexpand\x=
206   \expandafter\noexpand\y}
207   \global\mtoc@toks=\expandafter{\mtoc@tmp}}
208 \def\mtoc@getCmdName#1{\edef\mtoc@CmdName{\expandafter
209   \@gobble\string#1}}

```

¶¶ The minitocfmt environment defined. Is a ‘simplified’ way of designing toc entries.

`minitocfmt{\<cmdName>}` The definition of the environment. The argument is a command that will hold the expanded content of the environment. The body of the environment consists of one or more `\declaretocfmt` commands.

```

210 \newenvironment{minitocfmt}[1]{\makeatletter
211   \gdef\@mtoc@cmd@@{#1}\let\@A\mtoc@addto

```

The `\mtoc@getCmdName` returns the `cmdName` (without backslash). `cmdName` is used the creating command sequences, using to this definition.

```

212   \mtoc@getCmdName{#1}\mtoc@toks={\let\@PW\mtoc@PW\let\@DS\sl@dotsep
213     \let\@R\sl@tocrmarg}%

```

The body of the environment consists of one or more `\declaretocfmt` commands, these commands contribute to `\mtoc@toks`. `\mtoc@toks` consists of all the formatting declarations requested.

```

214 }{\expandafter\xdef\@mtoc@cmd@@{\the\mtoc@toks}}

```

`\mtoc@addto` is a macro to add to the declarations. Within `minitocfmt` is `\@A` is `\let` to `\mtoc@addto`.

```

215 \def\mtoc@addto#1{\edef\mtoc@tmp{\the\mtoc@toks}%
216   \global\mtoc@toks=\expandafter{\mtoc@tmp #1}}
217 \def\mtoc@PW#1{\def\@pnumwidth{#1}}

```

Here is code from `latex.ltx` for `\@dottedtocline`, we modify it so there are no leaders.

```

218 \def\no@dottedtocline#1#2#3#4#5{%

```

```

219 \ifnum #1>\c@tocdepth \else
220   \vskip \z@ \@plus.2\p@
221   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
222   \parindent #2\relax\@afterindenttrue
223   \interlinepenalty\@M
224   \leavevmode
225   \@tempdima #3\relax
226   \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
227   {#4}\nobreak
228 %   \leaders\hbox{$\m@th
229 %       \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
230 %       mu$}\hfill
231 % Insert an \hfill
232   \hfill
233   \nobreak
234   \hb@xt@\@pnumwidth{\hfil\normalfont \normalcolor #5}%
235   \par}%
236 \fi}
237 \end{package}

```

3 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.

Symbols	
!noinitocs (option)	1
\QA	211
\@B	195
\@D	194
\@DS	212
\@E	190
\@F	195
\@L	191
\@MiniTocListingsfalse	159
\@MiniTocListingstrue	133, 160
\@N	195
\@P	196
\@PW	212
\@Pg	191
\@R	194, 213
\@W	194
\@afterindenttrue	222
\@auxout	126
\@bsphack	125
\@chkForNl	5, 56, 83
\@currentsecname	123
\@dotsep	184, 229
\@dottedtocline	172
\@foundTOPLevelfalse	43, 66, 151
\@foundTOPLeveltrue	64
\@ifclassloaded	20, 26
\@input	114
\@minitocCnt	2, 11, 109, 110, 127, 140
\@minitocfalse	132, 134
\@minitoctrue	135
\@mtoc@cmd@@	211, 214
\@nlrtn	71, 84, 85
\@nobreakfalse	89
\@plus	220
\@pnumwidth	217, 234
\@rgii	185
\@startsection	122, 123
\@startsection@mtoc@SAVE	122, 124
\@starttoc	8, 113, 117
\@tocrmarg	183, 221
\AtEndDocument	104
\AtEndOfPackage	3
B	
\BTMLevel	3, 17, 19
\BTMLevelNum	18, 70, 142
C	
\c@tocdepth	219
\cl@LOOKFORSEC	4, 81, 82, 149
\cl@LOOKFORSEC@HY	4, 48, 49, 81
\cl@LOOKFORSEC@LX	4, 47, 82
\contentsline	7, 119, 149
\csarg	12, 23, 24, 29, 30, 33, 38–42, 127
D	
\DeclareOption	3, 5
\declaretocfmt	<u>187</u>
E	
environments:	
minitocfmt	<u>210</u>
F	
\FmtTOCEntry	135
H	
\hb@xt@	234
I	
\if@filesw	87
\if@foundTOPLevel	43, 50, 69
\if@minitoc	6, 132, 136
\if@MiniTocListings	133, 162
\ifMiniTocListings	<i>1</i> , 2
\InputIfFileExists	155
\insertminitoc	4, 6, 7, 138
\insertminitocNOT	4, 8, 166
\interlinepenalty	223
L	
\label	125
\leaders	228
\leftskip	221, 226
M	
\m@th	228
A	
\addtocontents	10, 107
\aebLastPage	99

\makeatletter	113, 155, 210	\NoFmtTOCEntry	115, 134
\makeatother	114, 161	nominitocs (option)	1
\miniorfulltoc	7, 136	\normalcolor	190, 234
minitocfmt (environment)	210	\normalfont	190, 234
\MiniTocListingsfalse	3	\null	226
\MiniTocListingstrue	2, 5	\numberline	84, 85, 152, 153
\mkern	229	\numBoxWidth	8, 167, 168, 194
\mt@rgi	49, 59, 108	\NUMLevel	51, 52, 70
\mtoc@@contentsline	44, 73, 75		
\mtoc@@star	186, 199	O	
\mtoc@@starttoc	90, 94, 95	\openout	88
\mtoc@@tmp	205, 207, 215, 216	options:	
\mtoc@addto	11, 211, 215	!nominitocs	1
\mtoc@addtocontents	10, 110, 111	nominitocs	1
\mtoc@BTMLevel	18		
\mtoc@CL@mtoc	6, 118, 150	P	
\mtoc@CmdName	189, 204, 208	\p@	220
\mtoc@contentsline	7, 46	\PackageError	143
\mtoc@getCmdName	208, 212	\PackageInfo	156, 157
\mtoc@HY@anchor	45, 191	\PackageWarning	163
\mtoc@next	120, 121	\parfillskip	221
\mtoc@numberline	152	\parindent	222
\mtoc@numBoxWidth	153, 167, 172, 175	\ProcessOptions	6
\mtoc@One	13, 90, 94, 186	\protected@write	126
\mtoc@parse	118, 119		
\mtoc@PgNum	44, 191	R	
\mtoc@PW	212, 217	\rightskip	221
\mtoc@rgi	139–141, 148		
\mtoc@sec	62, 63, 148	S	
\mtoc@st@rttoc	86, 96, 100	\SECNUM	60, 63
\mtoc@star	185, 198	\set@display@protect	61
\mtoc@start@next	91, 96, 97, 100, 101	\set@typeset@protect	67
\mtoc@start@toc	91, 104	\sl@@dots	171, 173, 174, 192, 197, 200, 201
\mtoc@starttoc	8, 117	\sl@@EntryFmt	154, 177, 178, 193, 201
\mtoc@tableofcontents	9, 116	\sl@@nodots	176
\mtoc@tmp	58, 72, 74	\sl@pNumFmt	181, 182, 193, 200, 201
\mtoc@TOC	105, 108	\sl@sNumFmt	154, 179, 180, 192
\mtoc@toks	169, 205, 207, 212, 214–216	\sl@all	37
\mtoc@TOPLLevel	16, 59, 144	\sl@chapter	22, 24, 28, 30
\mtoc@Zero	13, 95, 186, 199	\sl@current	172, 175, 188
\mtocCL	50, 53, 57, 106, 110, 150	\sl@dots	171, 194
\mtocgobble	2, 14	\sl@dotsep	184, 212
\mtoclabel	6, 125	\sl@EntryFmt	177, 195
\mtocref	6, 131, 141	\sl@next	197, 200–202
\mtocs@toks@	58, 72, 74, 148, 158, 161, 170	\sl@nodots	174, 195
		\sl@paragraph	37, 41
N		\sl@part	22, 23, 28, 29, 33
\newwrite	87	\sl@pNumFmt	181, 196
\no@dottedtocline	175, 218	\sl@section	36, 38
\nobreak	226, 227, 233	\sl@sNumFmt	179, 195

<code>\sl@subparagraph</code>	37, 42	<code>\TOCLevels</code>	3, 19, 21, 27, 32
<code>\sl@subsection</code>	36, 39	<code>\TOPLevel</code>	3, 15, 19
<code>\sl@subsubsection</code>	36, 40	<code>\TOPLevelNum</code>	16, 52, 142
<code>\sl@tocrmarg</code>	183, 194, 213		
		X	
T		<code>\x</code>	15, 17, 158, 203, 205
<code>\tableofcontents</code>	9, 112		
<code>\thepage</code>	99	Y	
<code>\TOCEntryNum</code>	45, 190	<code>\y</code>	204, 206

4 Change History

v1.2 (2018/08/29)	v1.4 (2018/08/29)
General: Created <code>aeb-minitoc.ins</code>	General: Some renaming of commands
2	2
v1.3 (2018/08/29)	v1.6 (2018/09/21)
General: Remove <code>hyperref</code> as a requirement	General: Code cleanup in preparation for
2	release
	2