LaTeX ToC Management[*]

Eureka

November 14, 2024

Contents

[*] All implement can be found in “latex.ltx”

1 Preface

Some macors you need to know, these macros can be found in 'The TeXbook', Page 347: 'Appendix B: BasicControlSequences'

```
\let\bgroup={       \let\egroup=}
\let\endgraf=\par   \let\endline=\cr

\def\lq{‘}          \def\rq{’}
\def\lbrack{[}      \def\rbrack{]}

\newlinechar=‘\^^J  \def\^^I{\ }    \def\^^M{\ }
\def\space{ }       \def\empty{}    \def\null{\hbox{}}

\def\obeyspaces{\catcode‘\ =\active}
\obeylines says "\let^^M=\par"
```

2 Main commands

The main commands for making a table of contents are:

```
* \@starttoc{<ext>}

* \addcontentsline{<table>}{<type>}{<entry>}

* \addtocontents{<table>}{<text>}

* \contentsline{<type>}{<entry>}{<page>}{<hyperref option>}

* \@dottedtocline{<level>}{<indent>}{<numwidth>}{<title>}{<page>}

* \numberline{<number>}
```

Firstly, we need to know that what is **contents(table)** ? Then we could know what the following commands will operator will. It is a file with extension **.toc**, **.lof**, **.lot**, etc. which contains the information about the table of contents, list of figures, list of tables, etc. respectively.

When and How does these commands add these contentsline ? If you see the definition of `\@sect`, you will see:

```
\def\@sect#1#2#3#4#5#6[#7]#8{%
  ...
  \ifdim \@tempskipa>\z@
    ...
    \csname #1mark\endcsname{#7}%
    \addcontentsline{toc}{#1}{% The key command lies here !!
      \ifnum #2>\c@secnumdepth \else
        \protect\numberline{\csname the#1\endcsname}%
      \fi
      #7}%
  \else
    \def\@svsechd{%
      ...
      \addcontentsline{toc}{#1}{% The key command lies here !!
        \ifnum #2>\c@secnumdepth \else
          \protect\numberline{\csname the#1\endcsname}%
        \fi
        #7}}%
  \fi
  \@xsect{#5}}

% Remark:
% * counter definition: \newcount\c@secnumdepth
%                       \newcount\c@tocdepth
```

To see these counter values, use:

```
\makeatletter
\the\c@tocdepth
\the\c@secnumdepth

\setcounter{tocdepth}{2}
\setcounter{secnumdepth}{5}
\the\c@tocdepth
\the\c@secnumdepth
\makeatother

% Result:
% 3 3 2 5
```

Please use commands `\newcount` only in package code and with much care. The other counter is used in command `\@dottedtocline`, see:

```
\def\@dottedtocline#1#2#3#4#5{%
  \ifnum #1>\c@tocdepth \else
  ...}
```

Almost all the section commands, like `\section`, `\subsection`, etc. are defined by `\@sect`. Thus they will call this command to add the contentsline. Something like:

```
\contentsline {section}{\numberline {1}<sse I>}{<page>}{}%
\contentsline {subsection}{\numberline {1.1}<sss I>}{<page>}{}%
\contentsline {subsection}{\numberline {1.2}<sss II>}{<page>}{}%
\contentsline {subsection}{\numberline {1.3}<sss III>}{<page>}{}%

% Remark:
% * The \numberline was added for that '\ifnum #2 < \c@secnumdepth'
```

If you load package ‘hyperref’, then the ToC file will look like:

```
\contentsline {section}{<sse I>s}{<page>}{section.1}%
\contentsline {subsection}{<sss I>}{<page>}{subsection.1.1}%
\contentsline {subsection}{<sss II>}{<page>}{subsection.1.2}%
\contentsline {subsection}{<sss III>}{<page>}{subsection.1.3}%
```

Thus, sometimes you need to renewcommand the `\theHchapter`, `\theHsection` etc as follows:

```
\renewcommand\theHchapter{Appendix-\thechapter}
\renewcommand\theHsection{\thechapter-\thesection}
```

Then the hyperref jumping will be correct.

2.1 The command \@starttoc

This command can be used to define `\tableofcontents`, `\listoffigures` etc. For example:

- `\@starttoc{lof}` is used in `\listoffigures`.
- `\@starttoc{lot}` is used in `\listoftables`.
- `\@starttoc{toc}` is used in `\tableofcontents`.

The full definition of this command is:

```
\def\@starttoc#1{%
  \begingroup
    \makeatletter
    \@input{\jobname.#1}% The key command lies here !!
    \if@filesw
      \expandafter\newwrite\csname tf@#1\endcsname
      \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
    \fi
    \@nobreakfalse
  \endgroup}
```

The switch `\if@filesw` is used to control whether the file will be written or not. The default value is true in latex.ltx or in the very begining of the document.

```
\newif\if@filesw \@fileswtrue
```

To set this switch to false, there are 2 ways:

```
% 1. use \nofiles
\def\nofiles{%
  \@fileswfalse
  \typeout{No auxiliary output files.^^J}%
  ....}

% 2. use \@fileswfalse directly
```

Thus, the first time , the handle `\csname tf@ #1\ endcsname` will be used to write `<table>` entries to the file. And the second time, the `<table>` file will be read in. This commands will automatically check if the file exists or not, see definition:

```
\def\@input#1{%
  \IfFileExists{#1}{\@@input\@filef@und}{\typeout{No file #1.}}}
```

2.2 The command \addcontentsline

Usually, user have to use this command to add her/his own entries to the table of contents. The syntax is:

```
\addcontentsline{<table>}{<type>}{<entry>}
```

What does this command do for us ? For example, after use command:

```
\addcontentsline{toc}{chapter}{References}
\addcontentsline{toc}{chapter}{Index}
```

There will be two entries(lines) in file: `\jobname.toc` as follows:

```
% The last empty brace is for hyperref option
\contentsline {chapter}{References}{1}{}%
\contentsline {chapter}{Index}{1}{}%
```

Thus you can simply see this command as: **addcontentsline** = **add** + **contentsline**
The full definition of this command is:

```
\def\addcontentsline#1#2#3{%
  \addtocontents{#1}{\protect\contentsline{#2}{#3}{\thepage}{}%
  \protected@file@percent}}

% Remark:
% * \protected@file@percent:a percent sign in toc file
```

2.3 The command \addtocontents

This command just add a text to the `<table>` file without any manipulation. Thus if you use a command like:

```
\addtocontents{toc}{something to be added}
```

There will be a line in toc file, something like:

```
\contentsline {<type-1>}{<entry-1>}{<page-1>}{}
something to be added
\contentsline {<type-2>}{<entry-2>}{<page-2>}{}
```

The full definition is:

```
\long\def\addtocontents#1#2{%
  \protected@write\@auxout
    {\let\label\@gobble \let\index\@gobble \let\glossary\@gobble}%
    {\string\@writefile{#1}{#2}}}% The key command lies here !!
```

`\@writefile` is used to write content to an **existing file stream**(`tf@` is the prefix), and definition of this command is:

```
\long\def\@writefile#1#2{%
  \@ifundefined{tf@#1}\relax
    {\@temptokena{#2}%
      \immediate\write\csname tf@#1\endcsname{\the\@temptokena}%
    }%
}
```

If you want to create a "list of theorems" file: `\jobname.lom`, you need creat this stream first:

```
\makeatletter
\newwrite\tf@lom
\immediate\openout\tf@lom\jobname.lom
\addcontentsline{lom}{section}{<section name>}
\makeatother
```

or simply use `\@starttoc` to create such stream:

```
\@starttoc{lom}
```

Thus you can add any text to not only toc file but also **lof, lot, lom** etc. by using this command.

2.4 The command \contentsline

This commands lies in the `<table>` file, **it is not a command in normal context**. But if you want to add the contentsline manually, you can use this command in some places, such as the very begining of document, chapter, etc.
The full definition is:

```
\def\contentsline#1#2#3#4{\gdef\@contentsline@destination{#4}%
  \csname l@#1\endcsname{#2}{#3}}

% By default:
\let\@contentsline@destination\@empty
```

`\csname l@#1\endcsname` is the old interface for `<table>` creating. They are defined in each document class. In `article.cls`, they are defined as:

```
\newcommand*\l@section[2]{%
  \ifnum \c@tocdepth >\z@
    \addpenalty\@secpenalty
    \addvspace{1.0em \@plus\p@}%
    \setlength\@tempdima{1.5em}%
    \begingroup
      \parindent \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
      \leavevmode \bfseries
      \advance\leftskip\@tempdima
      \hskip -\leftskip
      #1\nobreak\hfil
      \nobreak\hb@xt@\@pnumwidth{\hss #2%
```

```
                                \kern-\p@\kern\p@}\par
    \endgroup
  \fi}
\newcommand*\l@subsection{\@dottedtocline{2}{1.5em}{2.3em}}
\newcommand*\l@subsubsection{\@dottedtocline{3}{3.8em}{3.2em}}
\newcommand*\l@paragraph{\@dottedtocline{4}{7.0em}{4.1em}}
\newcommand*\l@subparagraph{\@dottedtocline{5}{10em}{5em}}
```

Remark:

- The fourth argument is used when 'hyperref' was loaded. This argument is saveed in a macro `\@contentsline@destination`.
- This command doese not print the table line, but just call the `\csname l@#1\endcsname` and command `\@dottedtocline` to generate the table line.

2.5 The command \@dottedtocline

This command is used to create a (dotted) tabel line in table of contents, for sections, for figures, for tables, etc. If we'd like to custom this dotted line, just modify these 4 arguments. The full definition is:

```
\def\@dottedtocline#1#2#3#4#5{%
  \ifnum #1>\c@tocdepth \else
    \vskip \z@ \@plus.2\p@
    {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
      \parindent #2\relax\@afterindenttrue
      \interlinepenalty\@M
      \leavevmode
      \@tempdima #3\relax
      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
      {#4}\nobreak
      \leaders\hbox{$\m@th
          \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
          mu$}\hfill
      \nobreak
      \hb@xt@\@pnumwidth{\hfil\normalfont \normalcolor #5}%
      \par}%
  \fi}
\def\numberline#1{\hb@xt@\@tempdima{#1\hfil}}
```

When and How this command generate the dotted line? This commands has not been used in `latex.ltx` but in some other basic document class, such as article class, book class, etc. See the usage in `article.cls`:

```
\newcommand*\l@section[2]{%
  \ifnum \c@tocdepth >\z@
    \addpenalty\@secpenalty
    \addvspace{1.0em \@plus\p@}%
    \setlength\@tempdima{1.5em}%
    \begingroup
```

```
      \parindent \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
      \leavevmode \bfseries
      \advance\leftskip\@tempdima
      \hskip -\leftskip
      #1\nobreak\hfil
      \nobreak\hb@xt@\@pnumwidth{\hss #2%
                                \kern-\p@\kern\p@}\par
    \endgroup
  \fi}
\newcommand*\l@subsection{\@dottedtocline{2}{1.5em}{2.3em}}
\newcommand*\l@subsubsection{\@dottedtocline{3}{3.8em}{3.2em}}
\newcommand*\l@paragraph{\@dottedtocline{4}{7.0em}{4.1em}}
\newcommand*\l@subparagraph{\@dottedtocline{5}{10em}{5em}}
\newcommand\listoffigures{%
    \section*{\listfigurename}%
      \@mkboth{\MakeUppercase\listfigurename}%
              {\MakeUppercase\listfigurename}%
    \@starttoc{lof}%
    }
\newcommand*\l@figure{\@dottedtocline{1}{1.5em}{2.3em}}
\newcommand\listoftables{%
    \section*{\listtablename}%
      \@mkboth{%
          \MakeUppercase\listtablename}%
        {\MakeUppercase\listtablename}%
    \@starttoc{lot}%
    }
\let\l@table\l@figure
```

Remark: This command defines the page number format, the number align, the number width, the number color. See the following definition:

```
\def\@dottedtocline#1#2#3#4#5{%
  ...
  \hb@xt@\@pnumwidth{\hfil\normalfont \normalcolor #5}%
  ...
}
```

2.6 The command \numberline

This command is simple to use. This command pull page number to right most in a box of width `\@tempdim`. The full definition is:

```
% \hb@xt@ = \hbox to
\def\numberline#1{\hb@xt@\@tempdima{#1\hfil}}
```

Remark: This command is the left label(such as 1.1, 1.1.1, etc.) in `<table>`, Thus we can redefine this command the modify the left margin of toc, lof, lot entries.

3 Conclusion

3.1 traditional way

Then we can see how the `table` is generated in LaTeX:

- Firstly, In each section command, the `\addcontentsline` will be called, a line starting with command `\contentsline` will be added to the `<table>` file afterward.
  - For the command `\contentsline` in the `<table>` file, the `\csname l@#1\endcsname` will be called. The first 3 arguments of command `\@dottedtocline` are provided by it.
  - Then the `\@dottedtocline` will be called by `\csname l@#1\endcsname`. The "entry label(like 1.1, 1.1.1, etc) + entry name"[2nd in `\contentsline`, 4th in `\@dottedtocline`] and "page number"[3rd in `\contentsline`, 5th in `\@dottedtocline`] will be grabbed by this command.
  - If package "hyperref" is loaded, then the last argument of `\contentsline` will be saved in a macro `\@contentsline@destination`, which will be used by `hyperref` package in the future.
  - The "entry label" will be packed in a box of width `@tempdima`, "entry name" will follow the "entry label" out of the box. Then they will be put in the left side of the `<table>` line; The page number will also be packed in a box of width `\@pnumwidth` and a dotted line will be added by command `\leaders\hbox{$<item>$}\nobreak`.
- Secondly, In the `\@starttoc{toc}` command will be called by `\tableofcontents`,
- Finally, `<table>` file will be inputed and the contents will be shown in the document.

3.2 titletoc package

If you use package 'titletoc' package or 'titlesec' package, then there will be something defferent.

4 Application

4.1 custom toc

4.2 aux file

Lets write a commands like `\ref`, `\tableofcontents`, we call it `\AfterItemCollector`. The definition of this new command is:

```
\makeatletter\ExplSyntaxOn
\NewDocumentCommand{\activeCollect}{O{ac}}{
  \if@filesw\@ifundefined{tf@#1}{}{
    \expandafter\newwrite\csname tf@#1\endcsname
    \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
    % Do not open a same file handle twice,
    % it will overwrite the original file content
  }\fi
}
\NewDocumentCommand{\AllItems}{O{ac}}{
  \@input {\jobname.#1}
}
\NewDocumentCommand{\collectedItem}{O{ac}+m}{
  \@writefile{#1}{#2}
}
\ExplSyntaxOff\makeatother

\activeCollect
The BEGIN
\collectedItem{\par --> First Invoke}
\collectedItem{\par --> Second Invoke}
The whole collected item are: \AllItems
\collectedItem{\par --> Third Invoke}
\collectedItem{\par --> Fourth Invoke}
\collectedItem{\par --> Fifth Invoke}
The END
```

After the first compilation, a file named `\jobname.ac` will be crerated; After the 2 compilations, you will get the output:

```
The BEGIN
The whole collected item are:
--> First Invoke
--> Second Invoke
--> Third Invoke
--> Fourth Invoke
--> Fifth Invoke
The END
```

The last 3 items have been successfully collected.

Remark:

- In this command, you can implement the `\write`,`\openout` function by LaTeX3, which will looks more elegant and more powerful.
- Command `\label` writing something to auxiliary file `*.aux` instead of `*.toc` file. It is like:

```
% ==> use \label-\ref
\label{hello}\ref{hello}



% ==> file: \jobname.aux
\newlabel{hello}{{4.2}{10}{}{}{}}
% If you load package 'hyperref', it will look like:
\newlabel{hello}{{4.2}{10}{aux file}{lstnumber.-27.8}{}}
```

“4.2” is the value of `currentlabel`, which will be updated by command `\refstepcounter`, see the definition for `\refstepcounter` in latex.ltx, it is:

```
\def\@currentcounter{}
\def\refstepcounter#1{\stepcounter{#1}%
  \edef\@currentcounter{#1}%
  \protected@edef\@currentlabel
  {\csname p@#1\expandafter\endcsname\csname the#1\endcsname}%
}
```