The tocbibind package*

Peter Wilson
Catholic University of America
Now at peter.r.wilson@boeing.com

2000/03/05

Abstract

The tocbibind package can be used to add document elements like a bibliography or an index to the Table of Contents. The package is designed to work with the four standard book, report, article and proc classes, and to a limited extent with the ltxdoc class. Results with other classes may be problematical. The package has been tested with the tocloft package, but has not been tested with other packages that change the definitions of the \chapter* or \section* commands.

The package requires the stdclsdv package.

Contents

1 Introduction 1

2 The tocbibind package 2
   2.1 Numbering the List of Figures, etc. 3
   2.2 Package Defined Listof... 4
   2.3 Abstracts 5

3 The package code 6

1 Introduction

Questions about adding the bibliography to the Table of Contents seem to pop up fairly regularly on the comp.text.tex newsgroup.

The tocbibind package provides a solution for automatically inserting references to a bibliography or an index, or other headed document elements into the Table of Contents. (tocbibind is meant to be shorthand for ‘Table of Contents, Bibliography, Index, etc’). Portions of the package were developed as part of a class and package bundle for typesetting ISO standards [Wil96]. This manual is typeset according to the conventions of the \LaTeX\ docstrip utility which enables the automatic extraction of the \LaTeX\ macro source files [GMS94].

Section 2 describes the usage of the package. Commented source code for the package is in Section 3. The package requires the stdclsdv package to be available.

*This file has version number v1.4a, last revised 2000/03/05.
2 The \texttt{tocbibind} package

The \texttt{tocbibind} package enables the titles of the Table of Contents, the List of Figures, the List of Tables, the Bibliography and the Index all to be added to the Table of Contents. By default, all of these document elements, if they exist, will be incorporated into the Table of Contents (ToC for short). Package options are available to switch off any of these inclusions.

- \texttt{notbib} Disables the inclusion of the Bibliography.
- \texttt{notindex} Disables the inclusion of the Index (inclusion of the Index of an \texttt{ltxdoc} class document is permanently disabled).
- \texttt{nottoc} Disables the inclusion of the ToC.
- \texttt{notlot} Disables the inclusion of the List of Tables.
- \texttt{notlof} Disables the inclusion of the List of Figures.
- \texttt{chapter} Use chapter-level headings, if possible.
- \texttt{section} Use section-level headings, if possible.
- \texttt{numbib} Number the Bibliography heading (default is no number).
- \texttt{numindex} Number the Index heading (default is no number).
- \texttt{other} Use a non-traditional heading command. This option effectively requires the use of the \texttt{\tocotherhead} command.
- \texttt{none} Disables everything.

The package is designed to work with the standard \LaTeX{} document classes \texttt{book}, \texttt{report}, \texttt{article}, \texttt{proc} and \texttt{ltxdoc} class (which is based to a large extent on the \texttt{article} class). In the \texttt{article}, \texttt{proc} and \texttt{ltxdoc} classes \LaTeX{} uses the \texttt{\section*} heading style for the bibliography etc., while for the other two classes it uses the \texttt{\chapter*} heading style. \texttt{tocbibind} honours these conventions. However, if the package is used with another class (perhaps with a class for typesetting theses which has different conventions), then the \texttt{chapter} or \texttt{section} options can be used to select the appropriate style (but the class must define \texttt{\chapter*} and \texttt{\@makeschapterhead}, or \texttt{\section*} respectively).

The standard classes, except for \texttt{ltxdoc}, have a feature whereby the height of the title for an index is at a different height than any other in a document (latex bug 3126). The \texttt{tocbibind} package disables this feature. The disablement has the side effect that the \texttt{\columnseprule} and \texttt{\columnsep} lengths can be set via \texttt{\setlength} to alter the column separation and the thickness of a rule between the two columns in the index. The effect of using the \texttt{none} option is to limit any changes to the single one of disabling this standard feature.

In the standard \LaTeX{} classes the bibliography and index headings are either both defined in terms of the \texttt{\chapter*} command or in terms of the \texttt{\section*} command. The package assumes that any class, other than the standard classes already mentioned, will either use code from the standard classes for implementing the bibliography and other headings, or will use very similar code. Some classes
(and maybe packages) change the names of the heading commands. One example that I am aware of uses \clause instead of \section, \sclause instead of \subsection and so on. If your document’s headings are defined like this and the same heading level is used for the bibliography, etc., then you can use the other option and the \tocotherhead{(headingname)} command to cater for this. If your document uses \clause then put \tocotherhead{clause} in the preamble after loading the package. The package then assumes that the bibliography heading is defined in terms of \clause*.

If you use the \tocotherhead command, then it overrides any \chapter or section option.

\tocbibname
The package attempts to pick up the name for the Bibliography from the class definition. (Note that the article class and its derivatives stores the name text in the \refname whilst the book and report classes store the name in \bibname).

This package uses \tocbibname to store the name of the bibliography.

\setindexname \settocname \setlotname \setlofname \settocbibname
These commands set the heading texts for the index, ToC, list of tables and list of figures. When used with the three standard classes, the heading text is picked up from the \indexname, \contentsname, \listtablename and \listfigurename commands respectively. The heading texts can be changed by changing the standard commands, or by using \setindexname{(name)}; and similarly for the other headings. Thus, the following two lines of code have the same effect:

\renewcommand{\listfigurename}{Figures}
\setlofname{Figures}

Note that these commands replace the \toc...name commands that were in version 1.1.

2.1 Numbering the List of Figures, etc.

Some authors like, or are required, to number the List of headings. Some commands are provided to simplify doing this.

In chaptered documents, the List of headings are effectively typeset as \chapter*{}. The natural way to get numbered headings would be to typeset them as \chapter{} but this has the potential disadvantage that the word ‘Chapter’, or equivalent, would be written before the heading, which is probably not what is required. The \simplechapter{⟨name⟩} command modifies any subsequent \chapter commands so that the result looks like that of \chapter* except that the chapter number is put on the same line as the title and the value of \simplechapterdelim is typeset immediately after the number. By default, \simplechapterdelim is empty. If the optional ⟨name⟩ argument is present, the ⟨name⟩ is typeset before the number. For example:

\renewcommand{\simplechapterdelim}{:}
\simplechapter[Chap]

will result in \chapter{First chapter} being typeset like:

Chap 1: First chapter

The \restorechapter command resets any subsequent \chapter commands to their default behaviour.

Internally, the List of commands in the tocbibind package use \toc@chapter
for typesetting the Listof headings in chaptered documents and \texttt{toc@section} for non-chaptered documents. The \texttt{tocchapter} command modifies the \texttt{toc@chapter} command to use a ‘simple chapter’ heading. The \texttt{tocsection} command modifies \texttt{toc@section} to typeset using \texttt{section} instead of \texttt{section*}.

For example, to get a numbered List of Figures heading in a chaptered document, put the following in the preamble:

\begin{verbatim}
\renewcommand{\listoffigures}{\begingroup
  \tocchapter
  \tocfile{\listfigurename}{lof}
\endgroup}
\end{verbatim}

while to get a numbered List of Tables in a non-chaptered document:

\begin{verbatim}
\renewcommand{\listoftables}{\begingroup
  \tocsection
  \tocfile{\listtablename}{lot}
\endgroup}
\end{verbatim}

More generally, to number the Table of Contents in a (non-)chaptered document you can do:

\begin{verbatim}
\renewcommand{\tableofcontents}{\begingroup
  \tocsection
  \tocchapter
  \tocfile{\contentsname}{toc}
\endgroup}
\end{verbatim}

The \texttt{begingroup \endgroup} pairing keeps the changes local.

\section*{2.2 Package Defined Listof...}

There are packages, such as \texttt{listings} and \texttt{ccaption}, that provide new Listof lists. These can be handled by the \texttt{tocbibind} package in a similar manner to the usual Listofs. Two examples are given below.

The \texttt{listings} package version 0.2 provides a \texttt{\lstlistoflistings} command to print a list of listings. The header name for this list is in \texttt{\lstlistingname} and the listing file has the extension \texttt{lol}. This can be treated just like the \texttt{listoffigure}, etc., commands. To add the List of Listings header to the ToC do:

\begin{verbatim}
\renewcommand{\lstlistoflistings}{\begingroup
  \tocfile{\lstlistingname}{lol}
\endgroup}
\end{verbatim}

and to number the Listof heading do:

\begin{verbatim}
\renewcommand{\lstlistoflistings}{\begingroup
  \tocsection
  \tocchapter
  \tocfile{\lstlistingname}{lol}
\endgroup}
\end{verbatim}
The `ccaption` package enables authors to define new kinds of floats (together with their captions) and Listof for each new kind of float. The command to define a new float is essentially \newfloatenv{⟨fenv⟩}{⟨ext⟩}{⟨capname⟩}, where ⟨fenv⟩ is the name of the new float environment and ⟨ext⟩ is the file extension for the listof file. The typesetting of the Listof listing is called by the command \listfloats{⟨fenv⟩}{⟨heading⟩}, where ⟨fenv⟩ is the name of the float environment and ⟨heading⟩ is the heading text for the Listof. For example, a new float environment for diagrams could be defined via \newfloatenv{diagram}{dia}{Diagram}, and the Listof called for by \listfloats{diagram}{List of Diagrams}.

In this case, to add the `List of Diagrams` to the ToC it is necessary to define a new listof command, and use this in place of the \listfloats{...}{...}. For the diagram example this could be (unnumbered):

\newcommand{\listofdiagrams}{\begingroup
\tocfile{List of Diagrams}{dia}
\endgroup}

and correspondingly for a numbered version:

\newcommand{\listofdiagrams}{\begingroup
\tocsection
\tocchapter
\tocfile{List of Diagrams}{dia}
\endgroup}

2.3 Abstracts

On rare occasions a publisher may want an abstract listed in the ToC. This package does not provide for that, partly because it is easier to do than the other headings. Just proceed along the lines below, where section might have to be chapter.

\begin{abstract}
\addcontentsline{toc}{section}{\abstractname}
... rest of the abstract
\end{abstract}

A question that arises from time to time on `comp.text.tex` is how to have a fullwidth abstract in a two column document. The code:

\documentclass[twocolumn]{article}
% all in twocolumn mode
...
\begin{abstract} ... \end{abstract}

puts the abstract into one of the columns. The trick is to start the document in single column mode and at the appropriate place issue a \twocolumn command with the abstract forming the optional argument to the command.\footnote{On `ctt` Donald Arseneau pointed out the desireability of the \@twocolumnfalse environment.} For example:

\documentclass[article]
In spite of the appearance of the @ symbol in the above, it is not necessary to enclose it all in a \makeatletter and \makeatother pair.

If you want, for example, to have the abstract treated like any other numbered part of the document or to change other aspects of the abstract environment, then you will have to do some work on your own. To save you looking it up, the default code for the abstract environment is:

\if@titlepage
  \newenvironment{abstract}{%
    \titlepage
    \null\vfil
    @beginparpenalty@lowpenalty
    \begin{center}%
    \bfseries \abstractname
    @endparpenalty@M
    \end{center}%
  }{%
  \par\vfil\null\endtitlepage
}
\else
  \newenvironment{abstract}{%
    \if@twocolumn
      \section*{\abstractname}%
    \else
      \small
      \begin{center}%
      \bfseries \abstractname\vspace{-.5em}\vspace{\z@} \end{center}%
      \quotation
    \fi}%
  }{\if@twocolumn\else\endquotation\fi}
\fi

This code is from the report and article classes; the book class does not have an abstract environment.

3 The package code

Announce the name and version of the package, which requires \LaTeX\ 2\epsilon and the stdclsdv package.

\footnote{\texttt{usc}}
These two commands write a package Note to the terminal and log file. Use as \PRWPackageNote{⟨package name⟩}{⟨note text⟩}. The NoLine version does not show the line number. The commands are intermediate between the kernel \PackageWarning and \PackageInfo commands. I have provided them as other packages (of mine) may also incorporate them. The code is based on lterror.dtx.

\providecommand{\PRWPackageNote}[2]{\GenericWarning{(\#1)\@spaces\@spaces\@spaces\@spaces}{\Package #1 Note: \#2}}
\providecommand{\PRWPackageNoteNoLine}[2]{\PRWPackageNote{\#1}{\#2\@gobble}}

Issue warning(s) if there are no recognized sectional divisions and prepare to skip the rest of the package code.

\ifSCDnodivs
\PackageWarning{tocbibind}{I don’t recognize any sectional divisions.}
I hope you have used the ‘other’ option
\renewcommand{\SCDquit}{\endinput}
\fi

\ifSCDknownclass\else
\PackageWarning{tocbibind}{I don’t recognize the class but I’ll do my best}
\fi

\if@inltxdoc This is used as a flag for the ltxdoc class. This has a particular kind of index that I am not going to mess with.
\newif{\inltxdoc}
\ifclassloaded{ltxdoc}{\inltxdoctrue}{\inltxdocfalse}

\if@bibchapter Used for knowing whether to use chapter or section style headings.
\newif{\bibchapter}\bibchapterfalse
\ifSCDchapter
\bibchaptertrue
\fi

\if@dotocbib A set of booleans for deciding what is to go into the ToC. By default add everything.
\newif{\dotocbib}\dotocbibtrue
\if@dotocind
\dotocbinlinefalse
\if@dotoctoc
\dotocfalse
\fi
\if@dotoclot
\dotocfalse
\fi
\fi
A set of booleans for deciding whether or not to produce numbered headings (default is to do unnumbered headings).

Now we can do the options. Most of them are easy.

The chapter option needs to check whether or not the chapter heading commands are defined. If they are not, then go with the section level headings.

The other option makes \texttt{SCDquit} a no-op and cancels any chapter based processing.

The none option turns everything off.
Process the options now, and then quit if necessary.

\ProcessOptions
\SCDquit

Issue a note about the heading style being used.
\if@bibchapter
  \PackageWarning{tocbibind}{Using chapter style headings}
  \PRWPackageNoteNoLine{tocbibind}{Using chapter style headings}
\else
  \PackageWarning{tocbibind}{Using section or other style headings}
  \PRWPackageNoteNoLine{tocbibind}{Using section or other style headings}
\fi

Ensure that the index is not processed if it is an \ltxdoc class.
\if@inltxdoc \dotocindfalse \fi

\@tocextra
\@tocextra is the internal command to store the heading command name.
\tocotherhead⟨name⟩ is the user command to set the heading command ⟨name⟩ (without the backslash). The default is section.

\newcommand{\@tocextra}{section}
\newcommand{\tocotherhead}[1]{\renewcommand{\@tocextra}{#1}}

\prw@mkboth
Utility macros, as the code that they represent gets used several times over. They deal with marking for page headers (code taken from \classes dtx), and adding starred sectional headings to the ToC.
\prw@mkboth{⟨left⟩}{⟨right⟩} as called by sectional headings.
\providecommand{\prw@mkboth}[1]{\@mkboth{MakeUppercase(#1)}}
\@nameuse{#1}*{{#2}}
\toc@section{⟨sec⟩}{⟨text⟩} is a generalised version of \sec*[⟨text⟩] which also makes an entry of ⟨text⟩ into the ToC, where ⟨sec⟩ is the name of a sectional division (with no backslash). \toc@headstar{⟨sec⟩}{⟨text⟩} is similar except that it makes no entry into the ToC.
\newcommand{\toc@section}[2]{%\@nameuse{#1}*{#2}}
\newcommand{\toc@headstar}[2]{%\@nameuse{#1}*{#2}}
\toc@chapter{⟨text⟩} is equivalent to \chapter*[⟨text⟩] except that it makes an entry into the ToC.
\newcommand{\toc@chapter}[1]{%\chapter*[#1]}
This holds the text for the Bibliography heading. We try and get the text from the class (either \bibname or \refname).

\ifx\bibname\undefined
\ifx\refname\undefined
\newcommand{\tocbibname}{References}
\else
\newcommand{\tocbibname}{\refname}
\fi
\else
\newcommand{\tocbibname}{\bibname}
\fi

The remaining heading texts are simpler as we only need to check if their respective names are defined in the class. Note that these commands in version 1.2 have been changed from version 1.1 in order to integrate with the tocloft package (which operates with the \contentsname etc commands).

\providecommand{\indexname}{Index}
\newcommand{\setindexname}[1]{\renewcommand{\indexname}{#1}}
\providecommand{\contentsname}{Contents}
\newcommand{\settocname}[1]{\renewcommand{\contentsname}{#1}}
\providecommand{\listtablename}{List of Tables}
\newcommand{\setlotname}[1]{\renewcommand{\listtablename}{#1}}
\providecommand{\listfigurename}{List of Figures}
\newcommand{\setlofname}[1]{\renewcommand{\listfigurename}{#1}}
\newcommand{\settocbibname}[1]{\renewcommand{\tocbibname}{#1}}

The rest is just hacking the various environments and commands from classes.dtx.

thebibliography redefine thebibliography, but only if requested.

\if@dotocbib
\renewenvironment{thebibliography}[1]
{\if@bibchapter
 {\if@onumbib
 \chapter{\tocbibname}
 \else
 \tocchapter{\tocbibname}
 \fi
 \else
 {\if@onumbib
 \@nameuse{@tocextra}{\tocbibname}
 \else
 \tocsection{\tocbibname}
 \fi
 \fi
 \begin{thebibitemlist}{#1}\end{thebibitemlist}}
\fi

Just as a matter of style, I have extracted the list making code from the definition of the thebibliography. It might also make it easier for someone to change the list environment. The code is a straight copy from classes.dtx.
theindex  In an earlier version of this package, for reasons that I didn’t understand, I had to add/remove some vertical space around the Index heading to make its height match other chapter/section headings. In an unrelated thread on the comp.text.tex newsgroup, Donald Arseneau pointed out that that effect was a known feature of the standard classes and recorded as latex bug 3126, and was caused by misplaced topskips. The following removes this feature for all except the doc class.

The first bit of code is a copy from classes.dtx.

This next bit is where we make the package changes. Note that in the default definition the values for \columnseprule and \columnsep were set at this point to be 0pt and 35pt respectively. They are not set in this definition so that they can be adjusted by the user, if necessary, before starting the environment.
Now we are back to the original code.

\twocolumn[vspace*2\topskip]%
\@makeschapterhead{\indexname}%
\@mkboth{\MakeUppercase\indexname}%
{\MakeUppercase\indexname}%
\fi
\fi
\else
\if@donumindex
\twocolumn[vspace*{-1.5\topskip}]%
\@nameuse{@tocextra}{\indexname}]
%% \prw@mkboth{\indexname}
\prw@mkboth{\csname the\@tocextra\endcsname \quad\indexname}
\else
\else
\if@dotocind
\twocolumn[vspace*{-1.5\topskip}]%
\toc@headstar{@tocextra}{\indexname}]
\prw@mkboth{\indexname}
\addcontentsline{toc}{@tocextra}{\indexname}
\else
\twocolumn[vspace*{-1.5\topskip}]%
\toc@headstar{@tocextra}{\indexname}]
\prw@mkboth{\indexname}
\fi
\fi
\fi
\fi
\else
\if@dotocind
\twocolumn[vspace*{-1.5\topskip}]%
\toc@headstar{@tocextra}{\indexname}]
\prw@mkboth{\indexname}
\addcontentsline{toc}{@tocextra}{\indexname}
\else
\twocolumn[vspace*{-1.5\topskip}]%
\toc@headstar{@tocextra}{\indexname}]
\prw@mkboth{\indexname}
\fi
\fi
\fi
\fi
\thispagestyle{plain}\parindent\z@
\parskip\z@ \@plus .3\p@\relax
\let\item\@idxitem}
\if@restonecol\onecolumn\else\clearpage\fi
\fi
\fi
\toc@start These two macros deal with the start and finish of the \tableofcontents and friends by adjusting the column settings if need be.
\toc@finish
\newcommand{\toc@start}{%
\ifSCDchapter
\if@twocolumn
\@restonecoltrue\onecolumn
\else
\@restonecolfalse
\else
\@restonecoltrue\onecolumn
\else
\@restonecolfalse
\fi
\fi
\fi
\newcommand{\toc@finish}{%
\ifSCDchapter
\if@restonecol\twocolumn\fi
\fi
\fi
\tocfile The code for \tableofcontents, \listoftables and \listoffigures is virtually identical in each case, except for the heading text. \tocfile embodies the common code. This is virtually a parameterized copy from classes.dtx, except
that it handles the differences between the article class and the other two, and incorporates the code for additions to the ToC. It is a useful hook if any other package wants to extend tocbibind for other kinds of listings.

The command is \texttt{\textbackslash tocfile\{⟨head-text⟩\}\{⟨file-extension⟩\}}, where ⟨head-text⟩ is the heading (e.g., List of Figures) and ⟨file-extension⟩ is the file extension (e.g., lof).

\begin{verbatim}
\newcommand{\tocfile}[2][]{% \\
toc@start
\if@bibchapter\texttt{\textbackslash toc@chapter\{#1\}}\else\texttt{\textbackslash toc@section\{\@tocextra\}{#1}}\fi
\texttt{\@starttoc\{#2\}}
\texttt{\toc@finish}}
\end{verbatim}

The next bit is for the heading changes.

\begin{verbatim}
\if@bibchapter \texttt{\textbackslash toc@chapter\{#1\}} \else \texttt{\textbackslash toc@section\{\@tocextra\}{#1}} \fi
\texttt{\@starttoc\{#2\}}
\texttt{\toc@finish}
\end{verbatim}

\texttt{\tableofcontents} If requested, we redefine this command, using \texttt{\tocfile} to do all the work for us.

\begin{verbatim}
\if@dotoctoc \renewcommand{\tableofcontents}{% \\
\texttt{\textbackslash tocfile\{\contentsname\}\{toc\}} \fi
\end{verbatim}

\texttt{\listoftables} This is almost identical to the code for \texttt{\tableofcontents}

\begin{verbatim}
\if@dotoclot \renewcommand{\listoftables}{% \\
\texttt{\textbackslash tocfile\{\listtablename\}\{lot\}} \fi
\end{verbatim}

\texttt{\listoffigures} This is almost identical to the code for \texttt{\tableofcontents}

\begin{verbatim}
\if@dotoclof \renewcommand{\listoffigures}{% \\
\texttt{\textbackslash tocfile\{\listfigurename\}\{lof\}} \fi
\end{verbatim}

\texttt{\simplechapter} The \texttt{\simplechapter} command modifies the \texttt{\@makechapterhead} command to result in an appearance akin to \texttt{\@makeschapterhead}, and is based on the latter.

\texttt{\restorechapter} The \texttt{\restorechapter} command restores everything back to its original state.

The value of \texttt{\simplechapterdelim} is appended to the chapter number before the title text.

\begin{verbatim}
\newcommand{\simplechapter}[1]{[\@empty]}% \\
\let\@tbiold@makechapterhead\@makechapterhead
\renewcommand{\@makechapterhead}[1]{% \\
\let\@tbiold@makechapterhead\@makechapterhead
\renewcommand{\@makechapterhead}{[1]{% \\
\let\@tbiold@makechapterhead\@makechapterhead
\renewcommand{\@makechapterhead}{[1]{% \\
\let\@tbiold@makechapterhead\@makechapterhead
\end{verbatim}

13
These two commands modify the \texttt{toc@chapter} and \texttt{toc@section} commands to make numbered Listof headings.

\begin{verbatim}
\newcommand{\tocchapter}{\simplechapter\renewcommand{\toc@chapter}[1]{\chapter{##1}}}
\newcommand{\tocsection}{\renewcommand{\toc@section}[2]{\@nameuse{##1}{##2}}}
\end{verbatim}

The end of this package.

References
