The tocbibind package*

Author: Peter Wilson, Herries Press
Maintainer: LaTeX Project team
https://github.com/LaTeX-Package-Repositories/herries-press

2026/06/01

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.

Contents

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].

*This file (`tocbibind.dtx`) has version number v1.5l, last revised 2026/06/01.

Section 2 describes the usage of the package. Commented source code for the package is in Section 3.

2 The tocbibind package

The 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.

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

The package is designed to work with the standard LaTeX document classes book, report, article, proc and ltxdoc class (which is based to a large extent on the article class). In the article, proc and ltxdoc classes LaTeX uses the `\section*` heading style for the bibliography etc., while for the other two classes it uses the `\chapter*` heading style. 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 chapter or section options can be used to select the appropriate style (but the class must define `\chapter*` and `\@makeschapterhead`, or `\section*` respectively).

The standard classes, except for 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 tocbibind package disables this feature. The disablement has the side effect that the `\columnseprule` and `\columnsep` lengths can be set via `\setlength` to alter the column separation and the thickness of a rule between the two columns in the index. The effect of using the none option is to limit any changes to the single one of disabling this standard feature.

`\tocotherhead` In the standard LaTeX classes the bibliography and index headings are either both defined in terms of the `\chapter*` command or in terms of the `\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 Listof headings. Some commands are provided to simplify doing this.

`\simplechapter` `\simplechapterdelim` `\restorechapter` In chaptered documents, the Listof 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.

`\tocchapter` `\tocsection` Internally, the Listof commands in the tocbibind package use `\toc@chapter` for typesetting the Listof headings in chaptered documents and `\toc@section` for non-chaptered documents. The `\tocchapter` command modifies the `\toc@chapter` command to use a

'simple chapter' heading. The `\tocsection` command modifies `\toc@section` to typeset using `\section` instead of `\section*`.

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

```
\renewcommand{\listoffigures}{\begingroup
   \tocchapter
   \tocfile{\listfigurename}{lof}
\endgroup}
```

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

```
\renewcommand{\listoftables}{\begingroup
   \tocsection
   \tocfile{\listtablename}{lot}
\endgroup}
```

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

```
\renewcommand{\tableofcontents}{\begingroup
   \tocsection
   \tocchapter
   \tocfile{\contentsname}{toc}
\endgroup}
```

The `\begingroup` `\endgroup` pairing keeps the changes local.

2.2 Page styles

The package, by default, supports the standard `empty`, `plain`, and `headings` page styles. Other page styles, for example ones you specify yourself via the fancyhdr package, are indirectly supported.

As an example, assume that you are using the fancyhdr package and you use a `fancy` pagestyle in a book/report class document like:

```
\pagestyle{fancy}
\renewcommand{\chaptermark}[1]{\markboth{\thechapter.\ #1}{}}
```

then you will find that the chapter titles in headers are in normalcase but the ToC, etc., headers are still in uppercase.

`\tocetcmark` In this package, the marks for the ToC, LoF…headers are specified via the command `\tocetcmark{`⟨*head*⟩`}`. To match the `fancy` pagestyle this must be redefined, like:

```
\pagestyle{fancy}
\renewcommand{\chaptermark}[1]{\markboth{\thechapter.\ #1}{}}
\renewcommand{\tocetcmark}[1]{\markboth{#1}{}}
```

which will give normalcase headers for the ToC, LoF…. As these are not normally numbered, it would be a misjudgement to try and get a non-existent chapter number into the header.

Documents with sections, but not chapters, can be treated in a similar manner by redefining `\tocetcmark` appropriately.

2.3 Package Defined Listof...

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

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

```
\renewcommand{\lstlistoflistings}{\begingroup
   \tocfile{\lstlistlistingname}{lol}
\endgroup}
```

and to number the Listof heading do:

```
\renewcommand{\lstlistoflistings}{\begingroup
   \tocsection
   \tocchapter
   \tocfile{\lstlistlistingname}{lol}
\endgroup}
```

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 `\newfloatlist{`⟨*fenv*⟩`}{`⟨*ext*⟩`}{`⟨*listname*⟩`}{`⟨*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 `\listoffenv`, where `fenv` is the name ⟨*fenv*⟩. For example, a new float environment for diagrams could be defined via
`\newfloatlist{diagram}{dia}{List of Diagrams}{Diagram}`, and the Listof called for by
`\listofdiagram`

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 `\listoffenv`. For the diagram example this could be (unnumbered):

```
\newcommand{\listofdia}{\begingroup
   \tocfile{List of Diagrams}{dia}
\endgroup}
```

and correspondingly for a numbered version:

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

and then use `\listofdia` instead of `\listofdiagram`.

2.4 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`, and if you are using the hyperref package you have to use the `\phantomsection` macro.

```
\begin{abstract}
% \phantomsection  % required if using hyperref
\addcontentsline{toc}{section}{\abstractname}
... rest of the abstract
```

3 The package code

Announce the name and version of the package, which requires LaTeX 2_ε.

```
⟨*usc⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{tocbibind}[2026/06/01 v1.5l extra ToC listings]
```

\PRWPackageNote
\PRWPackageNoteNoLine

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}%
}

```

(*End of definition for* `\PRWPackageNote` *and* `\PRWPackageNoteNoLine`.)

\@bibquit
\if@bibchapter

We need to know what sectional divisions are supported.

```
\newcommand{\@bibquit}{}
\newif\if@bibchapter
\@ifundefined{chapter}{%
  \@bibchapterfalse
  \@ifundefined{section}{%
    \PackageWarning{tocbibind}%
      {I don't recognize any sectional divisions.\MessageBreak
       I hope you have used the `other' option\MessageBreak
       otherwise I'll ignore the package}
    \renewcommand{\@bibquit}{\endinput}
    }{\PackageInfo{tocbibind}{The document has section divisions}}
  }{\@bibchaptertrue
    \PackageInfo{tocbibind}{The document has chapter divisions}}

```

(*End of definition for* `\@bibquit` *and* `\if@bibchapter`.)

`\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\if@inltxdoc
\@ifclassloaded{ltxdoc}{\@inltxdoctrue}{\@inltxdocfalse}

```

(*End of definition for* `\if@inltxdoc`.)

`\if@dotocbib` `\if@dotocind` `\if@dotoctoc` `\if@dotoclot` `\if@dotoclof` A set of booleans for deciding what is to go into the ToC. By default add everything.

```
\newif\if@dotocbib\@dotocbibtrue
\newif\if@dotocind\@dotocindtrue
\newif\if@dotoctoc\@dotoctoctrue
\newif\if@dotoclot\@dotoclottrue
\newif\if@dotoclof\@dotocloftrue

```

(*End of definition for* `\if@dotocbib` *and others.*)

`\if@donumbib` `\if@donumindex` A set of booleans for deciding whether or not to produce numbered headings (default is to do unnumbered headings).

```
\newif\if@donumbib\@donumbibfalse
\newif\if@donumindex\@donumindexfalse
```

(*End of definition for* `\if@donumbib` *and* `\if@donumindex`.)

`\if@dot@cb@bsection` If TRUE, use a section heading for the bibliography no matter what the main document divisions are.

```
\newif\if@dot@cb@bsection\@dot@cb@bsectionfalse

```

(*End of definition for* `\if@dot@cb@bsection`.)

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

```
\DeclareOption{section}{\@bibchapterfalse}
\DeclareOption{notbib}{\@dotocbibfalse}
\DeclareOption{notindex}{\@dotocindfalse}
\DeclareOption{nottoc}{\@dotoctocfalse}
\DeclareOption{notlot}{\@dotoclotfalse}
\DeclareOption{notlof}{\@dotocloffalse}
\DeclareOption{numbib}{\@donumbibtrue}
\DeclareOption{numindex}{\@donumindextrue}

```

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.

```
\DeclareOption{chapter}{%
  \if@bibchapter\else
    \PackageWarning{tocbibind}%
                   {Chapters are undefined, using section instead}
  \fi}

```

The other option makes `\@bibquit` a no-op and cancels any chapter based processing.

```
\DeclareOption{other}{\renewcommand{\@bibquit}{}
                      \@bibchapterfalse}
```

The none option turns everything off.

```
\DeclareOption{none}{%
  \@dotocbibfalse
  \@dotocindfalse
  \@dotoctocfalse
  \@dotoclotfalse
  \@dotocloffalse
  \@donumbibfalse
  \@donumindexfalse
}
```

Process the options now, and then quit if necessary.

```
\ProcessOptions\relax
\@bibquit

```

Issue a note about the heading style being used.

```
\if@bibchapter
  \PRWPackageNoteNoLine{tocbibind}{Using chapter style headings, unless overridden}
\else
  \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
\tocotherhead

`\@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}}

```

(*End of definition for* `\@tocextra` *and* `\tocotherhead`.)

\tocetcmark
\prw@mkboth
\toc@section
\toc@headstar

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.

`\tocetcmark{`⟨*text*⟩`}` is the default mark code as called by sectional headings.

```
\newcommand{\tocetcmark}[1]{%
  \@mkboth{\MakeUppercase{#1}}{\MakeUppercase{#1}}}
```

`\prw@mkboth{`⟨*text*⟩`}` is used later for the ToC headings.

```
\newcommand{\prw@mkboth}[1]{\tocetcmark{#1}}
```

`\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\prw@mkboth{#2}}
  \addcontentsline{toc}{#1}{#2}}
\newcommand{\toc@headstar}[2]{%
  \@nameuse{#1}*{{#2}}}
```

(*End of definition for* `\tocetcmark` *and others.*)

`\toc@chapter` `\toc@chapter{⟨text⟩}` is equivalent to `\chapter*{⟨text⟩}` except that it makes an entry into the ToC.
Until version 1.5f the chapter part of the code was `\chapter*{#1\prw@mkboth{#1}}`. On 2003/03/12 James Szinger[1] wrote that this failed for a bibliography in a two column book; the page headings for the previous chapter continued through the bibliography! James suggested that the mark part should be moved outside the chapter part (as is now done). I have no idea why there should have been this problem. As part of looking at it I even replaced the `\toc@chapter` as used in the `thebibliography` environment with the standard book class definition, which failed as well.

```
\newcommand{\toc@chapter}[1]{%
  \chapter*{#1}\prw@mkboth{#1}
  \addcontentsline{toc}{chapter}{#1}}
```

(*End of definition for* `\toc@chapter`.)

`\tocbibname` 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
```

(*End of definition for* `\tocbibname`. *This function is documented on page 3.*)

`\setindexname` `\settocname` `\setlotname` `\setlofname` `\settocbibname` 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}}
```

[1] szinger@lanl.gov

```
\providecommand{\listfigurename}{List of Figures}
\newcommand{\setlofname}[1]{\renewcommand{\listfigurename}{#1}}
\newcommand{\settocbibname}[1]{\renewcommand{\tocbibname}{#1}}
```

(*End of definition for* `\setindexname` *and others. These functions are documented on page 3.*)

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

Following a suggestion by Donald Arseneau (CTT, 'Re: memoir, natbib, and chapterbib', 9 Jan 2003), use `\bibsection` as a hook into `thebibliography` for the style of the heading.

`\t@cb@bchapsection` `\t@cb@bsection` Internal macros holding the heading for `thebibliography`.

```
\newcommand{\t@cb@bchapsec}{%
  \if@bibchapter
    \if@donumbib
      \chapter{\tocbibname}%
    \else
      \toc@chapter{\tocbibname}%
    \fi
  \else
    \if@donumbib
      \@nameuse{\@tocextra}{\tocbibname}%
    \else
      \toc@section{\@tocextra}{\tocbibname}%
    \fi
  \fi}
\newcommand{\t@cb@bsection}{%
  \if@donumbib
    \@nameuse{\@tocextra}{\tocbibname}%
  \else
    \toc@section{\@tocextra}{\tocbibname}%
  \fi}

```

(*End of definition for* `\t@cb@bchapsection` *and* `\t@cb@bsection`.)

Redefine `thebibliography`, but only if requested. Take care that the natbib package has not already modified the environment, noting that natbib defines and uses `\bibsection`.

```
\if@dotocbib
```

```
  \@ifpackageloaded{natbib}{}{% natbib not loaded
```

The natbib package has not been used (yet), so go ahead and change the environment.

`\bibsection` Macro holding heading for `thebibliography`.

```
    \newcommand{\bibsection}{\t@cb@bchapsec}
```

(*End of definition for* `\bibsection`.)

`thebibliography` (*env.*)

```
    \renewenvironment{thebibliography}[1]{%
      \bibsection
      \begin{thebibitemlist}{#1}}{\end{thebibitemlist}}}
```

`thebibitemlist` (*env.*) 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`.

```
\newenvironment{thebibitemlist}[1]{
  \list{\@biblabel{\@arabic\c@enumiv}}%
       {\settowidth\labelwidth{\@biblabel{#1}}%
        \leftmargin\labelwidth
        \advance\leftmargin\labelsep
        \@openbib@code
        \usecounter{enumiv}%
        \let\p@enumiv\@empty
        \renewcommand\theenumiv{\@arabic\c@enumiv}}%
  \sloppy
  \clubpenalty4000
  \@clubpenalty \clubpenalty
  \widowpenalty4000%
  \sfcode`\.\@m}
 {\def\@noitemerr
   {\@latex@warning{Empty `thebibliography' environment}}%
   \endlist}

```

`\sectionbib` The chapterbib package defines a macro `\sectionbib` which, if its `sectionbib` option is used, it calls at the beginning of the document to fiddle with the `thebibliography` environment (but it doesn't work when it is renewed as above). We need to disable the macro because we do our own fiddling

```
\@ifpackagewith{chapterbib}{sectionbib}%
  {\renewcommand{\sectionbib}[2]{}}%
  {}

```

(*End of definition for* `\sectionbib`.)

This is the end of `\if@dotocbib`.

```
\fi

```

At the end of the preamble we have to check if the natbib and/or chapterbib packages have been loaded after the tocbibind package. If this is the case, we have to make sure that we have control with respect to their `sectionbib` options.

```
\AtBeginDocument{%
  \@ifpackagewith{natbib}{sectionbib}{\@dot@cb@bsectiontrue}{}
```

If the chapterbib package was loaded before tocbibind we have already killed `\sectionbib`. If chapterbib has been loaded afterwards we must kill `\sectionbib` now before it gets used.

```
\@ifpackagewith{chapterbib}{sectionbib}%
  {\@dot@cb@bsectiontrue
   \@ifundefined{sectionbib}{}{\def\sectionbib#1#2{}}}%
  {}

```

Lastly, use our definition of `\bibsection` for the `thebibliography` environment.

```
\if@dotocbib
  \if@dot@cb@bsection
    \renewcommand{\bibsection}{\t@cb@bsection}%
  \else
    \renewcommand{\bibsection}{\t@cb@bchapsec}%
  \fi
\fi
```

This is the end of `\AtBeginDocument`

```
}
```

`theindex` (*env.*) 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 this 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`.

```
\if@inltxdoc\else
  \renewenvironment{theindex}%
    {\if@twocolumn
       \@restonecolfalse
     \else
       \@restonecoltrue
     \fi
```

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.

```
     \if@bibchapter
        \if@donumindex
          \refstepcounter{chapter}
          \twocolumn[\vspace*{2\topskip}%
                     \@makechapterhead{\indexname}]%
          \addcontentsline{toc}{chapter}{\protect\numberline{\thechapter}\indexname}
          \chaptermark{\indexname}
        \else
          \if@dotocind
            \twocolumn[\vspace*{2\topskip}%
                       \@makeschapterhead{\indexname}]%
            \prw@mkboth{\indexname}
            \addcontentsline{toc}{chapter}{\indexname}
          \else
            \twocolumn[\vspace*{2\topskip}%
                       \@makeschapterhead{\indexname}]%
            \prw@mkboth{\indexname}
          \fi
        \fi
     \else
        \if@donumindex
```

```
          \twocolumn[\vspace*{-1.5\topskip}%
                     \@nameuse{\@tocextra}{\indexname}]%
          \csname \@tocextra mark\endcsname{\indexname}
        \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
```

Now we are back to the original code.

```
    \thispagestyle{plain}\parindent\z@
    \parskip\z@ \@plus .3\p@\relax
    \let\item\@idxitem}
    {\if@restonecol\onecolumn\else\clearpage\fi}
\fi

```

`\toc@start` `\toc@finish` These two macros deal with the start and finish of the `\tableofcontents` and friends by adjusting the column settings if need be.

```
\newcommand{\toc@start}{%
  \if@bibchapter
    \if@twocolumn
      \@restonecoltrue\onecolumn
    \else
      \@restonecolfalse
    \fi
  \fi}

\newcommand{\toc@finish}{%
  \if@bibchapter
    \if@restonecol\twocolumn\fi
  \fi}
```

(*End of definition for* `\toc@start` *and* `\toc@finish`.)

`\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 `\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`).

```
\newcommand{\tocfile}[2]{%
  \toc@start
```

The next bit is for the heading changes.

```
  \if@bibchapter
    \toc@chapter{#1}
  \else
    \toc@section{\@tocextra}{#1}
  \fi
```

And finish up with a parameterized call to start the listing and tidy up.

```
  \@starttoc{#2}
  \toc@finish}

```

(*End of definition for* `\tocfile`.)

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

```
\if@dotoctoc
  \renewcommand{\tableofcontents}{%
    \tocfile{\contentsname}{toc}
  }
\fi

```

(*End of definition for* `\tableofcontents`.)

`\listoftables` This is almost identical to the code for `\tableofcontents`

```
\if@dotoclot
  \renewcommand{\listoftables}{%
    \tocfile{\listtablename}{lot}
  }
\fi

```

(*End of definition for* `\listoftables`.)

`\listoffigures` This is almost identical to the code for `\tableofcontents`

```
\if@dotoclof
  \renewcommand{\listoffigures}{%
    \tocfile{\listfigurename}{lof}
  }
\fi

```

(*End of definition for* `\listoffigures`.)

`\simplechapter` `\restorechapter` `\simplechapterdelim` The `\simplechapter` command modifies the `\@makechapterhead` command to result in an appearance akin to `\@makeschapterhead`, and is based on the latter. The `\restorechapter` command restores everything back to its original state. The value of `\simplechapterdelim` is appended to the chapter number before the title text. If headings use templates when `\DocumentMetadata` is used, a different definition is needed.

```
\IfDocumentMetadataTF
  {
   \newcommand\simplechapter[1][]{%
```

If we haven't changed the normal `chapter` instance do it now:

```
271           \IfInstanceExistsF{heading}{chapter-orig}{%
272             \DeclareInstanceCopy{heading}{chapter-orig}{chapter}%
273             \DeclareInstanceCopy{headformat}{chapter-orig}{chapter}%
274           }%
```

Then alter some of the instance keys, one important part is to use our own instance for `headformat-instance`.

```
275           \EditInstance{heading}{chapter}{%
276              prefix = #1
277             ,headformat-instance=chapter
278             ,prefix-decls=\Huge
279             ,number-decls=\Huge
280             ,number-format=\thechapter\simplechapterdelim
281           }%
```

Instead of the default of 20pt we only want a word space after chapter number. The other adjustment that is neccessary is to check if the prefix was made empty. In that case we also don't want a `prefix-number-sep`.

```
282           \DeclareInstance{headformat}{chapter}{hang}{%
283              number-title-sep=\WordSpaceAmount{1}%
284             ,prefix-number-sep:e=\IfBlankTF{#1}{0pt}{\noexpand\WordSpaceAmount{1}}%
285           }%
286         }%
```

Restoring then works by copying the saved instance back (if it exists).

```
287     \newcommand\restorechapter{%
288       \IfInstanceExistsT{heading}{chapter-orig}{%
289         \DeclareInstanceCopy{heading}{chapter}{chapter-orig}%
290         \DeclareInstanceCopy{headformat}{chapter}{chapter-orig}%
291       }%
292     }%
293   }
294   {
295     \newcommand{\simplechapter}[1][\@empty]{%
296       \let\@tbiold@makechapterhead\@makechapterhead
297       \renewcommand{\@makechapterhead}[1]{%
298         \vspace*{50\p@}%
299         {\parindent \z@ \raggedright
300          \normalfont
301          \interlinepenalty\@M
302          \Huge\bfseries #1\space\thechapter\simplechapterdelim\space
303             ##1\par\nobreak
304          \vskip 40\p@
305        }}
306     }
307     \newcommand{\restorechapter}{%
308       \@ifundefined{@tbiold@makechapterhead}{}%
309       {\let\@makechapterhead\@tbiold@makechapterhead}
310     }
311   }
312 \newcommand{\simplechapterdelim}{}
```

(*End of definition for* `\simplechapter`*,* `\restorechapter`*, and* `\simplechapterdelim`*. These functions are documented on page 3.*)

\tocchapter
\tocsection

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

```
\newcommand{\tocchapter}{%
  \providecommand{\@makechapterhead}{}
  \simplechapter
  \renewcommand{\toc@chapter}[1]{\chapter{##1}}
}
\newcommand{\tocsection}{%
  \renewcommand{\toc@section}[2]{\@nameuse{##1}{##2}}
}
```

(*End of definition for* \tocchapter *and* \tocsection*. These functions are documented on page 3.*)

The end of this package.

⟨/usc⟩

References

[GMS94] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley Publishing Company, 1994.

[Wil96] Peter R. Wilson. *LaTeX for standards: The LaTeX package files user manual*. NIST Report NISTIR, June 1996.

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

W