The tableof package

Package version: v1.4d (2024/09/17)
Documentation generated from tableof.dtx with timestamp 17-09-2024 at 10:15:25 CEST

Abstract

Provides `\toftagstart{}`, `\toftagstop{}`, `\toftagthis{}`, `\tofuntagthis{}` to tag chapters, sections or any other sectioning units destined to end up in the table(s) of contents. Then each one of

```
\tableof{required tags}
or \tablenotof{excluded tags}
or \tableoftaggedcontents{required tags}{excluded tags}
```

typesets a table of contents (with no heading) obeying the conditions. These macros can each be used multiple times in the document with varying arguments.

If the document contains no usage of `\tableofcontents`, the preamble should also contain an additional `\AtBeginDocument{\tofOpenTocFileForWrite}`.

The main `\tableofcontents` can also be influenced by tags like this:

```
\nexttocwithtags{required tags}{excluded tags}
\tableofcontents
```

Depending on the document class and packages, `\tableofcontents` however may be usable only once, contrarily to the `\tableof{}` et al. package macros.

Contents

1 Tagging commands

1.1 \toftagstart {tag1, tag2, ...} and \toftagstop {tag1, tag2, ...}

These commands have a mandatory argument which is a comma separated list of tags. The tags need not have been predeclared.

```
\toftagstart{kitchenware, weaponry, gastronomy}
\section{Dealing with knives}    % tagged with kitchenware+weaponry+gastronomy

\toftagstop{kitchenware}
\section{Hunting rabbits}        % tagged with weaponry+gastronomy

\toftagstart{tag1}
\subsection{This is tagged, too} % tagged with weaponry+gastronomy+tag1

\toftagstop{weaponry}
\section{Eating rabbits}         % tagged with gastronomy+tag1
```

1.2 \toftagthis {tag1, tag2, ...} and \tofuntagthis {tag1, tag2, ...}

The `\toftagthis` command flags with the comma separated values from its list argument only the *next* sectioning command. The effect is not cumulative: it is the last use of `\toftagthis` which counts; use it for all the needed tags at once.

The `\tofuntagthis` command similarly untags only the *next* entry. Similarly, only the tags in the last call of `\tofuntagthis` are taken into account.

```
\toftagstart{kitchenware, rabbits}
\section{Knives and rabbits}     % tagged with kitchenware and rabbits

\tofuntagthis{kitchenware}
\subsection{Hunting rabbits}     % tagged only with rabbits

\subsection{Best knives for cooking} % tagged with kitchenware and rabbits

\toftagstart{ecology}
\toftagthis{climate}
\section{Knives and global climate} % tagged with kitchenware+rabbits+ecology+climate

\toftagstop{kitchenware}
\section{The rabbit in the wild}    % tagged with rabbits+ecology
```

1.3 Spaces

Spaces in tags and around commas in tag lists are transparently removed. Tags may be macros, they are completely expanded before use.

2 Table of contents commands

2.1 \nexttocwithtags {required tag1, required tag2, ...}{excluded tag1, excluded tag2, ...}

This command influences the next \tableofcontents (or equivalent) command:

```
\nexttocwithtags{A, B}{C, D, E}
\tableofcontents
```

will let \tableofcontents print only the sectioning units having been flagged with both A and B and none of C, D, or E.

DO NOT FORGET THE SECOND PAIR OF BRACES EVEN IF YOU ONLY WANT TO REQUIRE SOME TAGS: \nexttocwithtags{tag1, tag2}{}.

2.1.1 starred variants

There are starred variants:

```
\nexttocwithtags{A, B}{C, D, E}   % A and B and neither C nor D nor E
\nexttocwithtags*{A, B}{C, D, E}  % (A or B) and neither C nor D nor E
\nexttocwithtags{A, B}*{C, D, E}  % A and B and (not C or not D or not E)
\nexttocwithtags*{A, B}*{C, D, E} % (A or B) and (not C or not D or not E)
```

2.2 \tableoftaggedcontents {required tag1, required tag2, ...}{excluded tag1, excluded tag2, ... }

This command is provided in case the document class allows only a single use of \tableofcontents, indeed \tableoftaggedcontents can be used arbitrarily many times. **However it does not typeset a heading**. Example:

```
\section*{A table of tagged contents}  % <- needs to be explicitely added
\tableoftaggedcontents{weaponry, hunting}{ecology, climate}
```

This will limit the printed TOC entries to the ones which have been tagged with weaponry and also with hunting, but not with ecology and neither with climate.

2.2.1 starred variants

There are starred variants:

```
\tableoftaggedcontents{A, B}{C, D, E}   % A and B and neither C nor D nor E
\tableoftaggedcontents*{A, B}{C, D, E}  % (A or B) and neither C nor D nor E
\tableoftaggedcontents{A, B}*{C, D, E}  % A and B and (not C or not D or not E)
\tableoftaggedcontents*{A, B}*{C, D, E} % (A or B) and (not C or not D or not E)
```

2.3 \tableof {required tag1, required tag2, ...}

This is equivalent to \tableoftaggedcontents{required tag1, ...}{}.

```
\tableof{weaponry,hunting}  % will print the entries tagged with weaponry
                                                        AND hunting
```

There is a starred variant:

```
\tableof*{weaponry, hunting} % will print the entries tagged with weaponry
                                                         OR hunting
```

2.4 \tablenotof {excluded tag1, excluded tag2, ...}

This is equivalent to \tableoftaggedcontents{}{excluded tag1, ...}.

```
\tablenotof{weaponry, hunting}
% will print the entries NOT tagged with weaponry NEITHER with hunting
```

There is a starred variant:

```
\tablenotof*{weaponry, hunting}
% will print the entries NOT tagged with weaponry OR NOT tagged with hunting
```

2.5 \tofOpenTocFileForWrite

The contents of the .toc file (if it already exists) are read into memory by **tableof** once, at the time of \begin{document}.[1] Notice that this reading of the .toc file into memory is only of relevance if the document makes use of one of the \tableoftaggedcontents, \tableof or \tablenotof commands.

The creation of the .toc file is *not* dealt with by **tableof** itself: either this will be done by a standard \tableofcontents command somewhere in the document, or, one may use the package provided command \tofOpenTocFileForWrite which does not display anything and just does what its name indicates. This command should **not** be used in the preamble, however

```
\usepackage{tableof}
\AtBeginDocument{\tofOpenTocFileForWrite}
```

is possible.

Please consider: this command should *not* be used when the document makes use of its own \tableofcontents or equivalent. It is provided *only* for the case where the document uses *exclusively* \tableoftaggedcontents, \tableof and \tablenotof.

3 Compatibility with other packages

tableof requires the document .toc file to use the \contentsline macro (possibly patched by other packages). It is thus incompatible with the beamer class. However, if beamer is used in an article mode, i.e., with the article class in conjunction with the beamerarticle package, then **tableof** should work.

[1] New with 1.3. Earlier versions read the .toc file at the time of \usepackage{tableof}. But this could cause a problem if the .toc file used characters not yet activated by Babel.

tableof adds the tag data to the `.toc` file, but this data self-defines itself to do nothing when not activated by **tableof**'s own commands. Since release `1.3`, these redefinitions are done in *one single* line of the `.toc` file to circumvent interference of `biblatex` which adds commands to the `.toc` line on every other line (thus, potentially right in the middle of multi-line commands `:-(`.)

No testing has actually been done of compatibility with packages manipulating tables of contents, apart from **etoc**.[2] It went fine.

4 Version History

`2024/09/17 v1.4d:` reestablish compatibility with `hyperref`, which got broken at the latter `v7.01c` release. The package now requires LaTeX `2020-10-01`. Thanks to Ekkart Kleinod for report.

`2021/07/05 v1.4c:` ensure the added scope-limiting group for `\tableof{}` and alike macros encompasses all contents of the `.toc` file even in presence of packages hacking into these contents (in particular `biblatex`).

`2018/10/02 v1.4b:` fix for situations when a `\clearpage` before the `\end{document}` resulted in the loss of the `\tof@finish` token from `.toc` file, causing the package to misbehave. The package `atveryend` is now required.[3]

`2015/03/10 v1.4a: i.` changes in the code to make it more easily patchable by other packages (I have especially the next release of **etoc** in mind): changes to the way `\tof@begin`, `\tof@finish` are set up, new `\tof@global` which defaults to nothing but may be set to be `\global` (this is in view of doing TOCs as tabulars of longtables, in the way **etoc** `1.08` will permit).

`ii.` improved sectioning of the documentation.

`2015/02/20 v1.4: i.` some code efficiency improvements (perhaps...).

`ii.` improved documentation.

`2015/02/11 v1.3: i.` comma separated lists of tags now allow spaces.

`ii.` only one line used for the `.toc` code silently unactivating `tableof` if its `\usepackage{tableof}` has been removed from the document source. (needed due to aggressive `biblatex` interference with `.toc` files.)

`iii.` reading of .toc file is delayed to `\begin{document}` to account for possible Babel active characters used therein. Not relevant if document uses standard `\tableofcontents` or like commands.

`2013/03/04 v1.2: i.` added `\tableoftaggedcontents` as a wrapper for using `\nexttocwithtags` followed with `tableof`'s private copy of the `.toc` data.

`ii.` added `\if@filesw` test to `\tofOpenTocFileForWrite`.

`2012/12/13 v1.1: i.` new command `\nexttocwithtags`.

[2] `http://www.ctan.org/pkg/etoc`
[3] `http://www.ctan.org/pkg/atveryend`

ii. .toc may be input in another document not loading tableof.

2012/12/06 v1.0: initial version.

5 Implementation

```
\ProvidesPackage{tableof}
 [2024/09/17 v1.4d Tables of tagged contents (JFB)]
\NeedsTeXFormat{LaTeX2e}[2020/10/01]
\RequirePackage{atveryend}
\DeclareOption*{\PackageWarning{tableof}{Option `\CurrentOption' is unknown.}}
\ProcessOptions\relax
\newtoks\tof@toctoks
```

1.3 codes this \tof@readtoc slightly better (copied from etoc.dtx).

```
\def\tof@readtoc {%
  \ifeof\tof@tf
  \else
     \read\tof@tf to \tof@buffer
     \tof@toctoks\expandafter\expandafter\expandafter
       {\expandafter\the\expandafter\tof@toctoks\tof@buffer}%
     \expandafter\tof@readtoc
  \fi }
```

1.3 of 2015/02/11 moves the reading of the toc file to At Begin Document. This is needed for Babel activated characters. I also re-use \endlinechar-1 which I had commented out since release 1.1. Notice though that this is irrelevant if the document uses tableof only via its tagging abilities, and has standard \tableofcontents command to print the TOC.

1.4c injects \tof@begingroup and \tof@endgroup to wrap the gathered the contents of the toc file, rather than having them arise from expansion of \tof@begin and respectively \tof@finish. This avoids a problem with biblatex additions to the .toc file happening before \tof@begin. They need to have their scope limited. The \tableof{} macro and variants will thus achieve this automatically via the \tof@begingroup/\tof@endgroup pair now explicitly added to \tof@toctoks contents.

etoc (a.t.t.o.w 1.09c 2020/05/15) has some handling of \tof@begingroup/\tof@endgroup which as far as I understand can currently remain as it is. But there is something weird in etoc with a test of \tof@finish which probably is in need of revision (even independently of changes here).

These changes however mean for usage of \nexttocwithtags that whatever macro is used to typeset the TOC, it is now the one bearing the responsability for creating the scope-limiting group.

```
\AtBeginDocument{%
  \IfFileExists{\jobname.toc}
    {{\endlinechar\m@ne
        \makeatletter
        \newread\tof@tf
        \openin\tof@tf\@filef@und
        \tof@toctoks{\tof@begingroup}%
        \tof@readtoc
        \global\tof@toctoks=\expandafter{\the\tof@toctoks\tof@endgroup}%
        \closein\tof@tf}}
    {}%
}
```

The trick is that \@ifundefined chooses the undefined branch if the meaning is \relax. Thus it is possible for tableof to define and use \tof@begin and later set it to \relax, the entire influence of tableof will then be turned off.

1.2 of 2013/03/04: \string{->{, idem with }. And added \if@filesw test.

1.3 of 2015/02/11: to circumvent aggressive biblatex manipulation of the .toc file, I put things to the .toc file in one-go; else weird commands might get inserted right in the middle of arguments to a multi-line macro! also removed a bunch of \string's: when I first wrote this package I had still limited understanding of TEX's macros.

1.4a uses \@empty for the default \tof@finish rather than \relax, for no special reason. Also the \let's in case \tof@begin is undefined or \relax are all made global, because for compatibility with the fancy things etoc 1.08 will allow for TOC as table we need a global mode, and the simplest is here to do the things global by default.

1.4c replaces {} (last argument of \@ifundefined) by \relax. No strong reason. We and etoc use \endlinechar-1 anyhow.

```
\AtBeginDocument{%
    \addtocontents{toc}{\string\@ifundefined{tof@begin}%
       {\global\let\string\tof@begin\relax
        \global\let\string\tof@finish\string\@empty
        \global\let\string\tof@starttags\string\@gobble
        \global\let\string\tof@stoptags\string\@gobble
        \global\let\string\tof@tagthis\string\@gobble
        \global\let\string\tof@untagthis\string\@gobble}\relax}%
    \addtocontents{toc}{\string\tof@begin}%
}
```

Now obsolete hyperref-related remark (see next paragraph): *LaTeX of 2020 or 2021 always has* \contentsline *with four arguments. So an update should be done here to always gobble four, else in absence of* hyperref *some {} are left. Does not seem to matter a lot except if all is executed in math mode... thanks to* etoc *for example. No urgency here, only mentioning for the record.*

hyperref stopped defining its \hyper@last at its 7.01c release in fall of 2023, but tableof tested for it, if hyperref was found to be present, in case a document previously not using hyperref started using it on second compilation, hence with a wrong .toc file format (in earlier years). As tableof was doing

```
\ifx\hyper@last\@undefined\tof@toctoks{}\fi
```

it now unconditionally emptied \tof@toctoks in presence of hyperref. This is the same problem which impacted etoc and got fixed at its 1.2c release but unfortunately tableof was not updated then. Belatedly fixed at 1.4d of 2024/09/17.

Formerly this was a short macro, but it doesn't look like a problem to simply alias it to the long \@gobblefour.

```
\let\tof@gobblethree@orfour\@gobblefour
```

1.4b (belatedly) fixes issue with \tof@finish getting lost due to a final \clearpage before \end{document}. Indeed, formerly code did:

```
\AtEndDocument{\addtocontents{toc}{\string\tof@finish}}
```

But we can't replace this by some \immediate\write\@auxout at end document, because it would act *before* the writes triggered by the \clearpage from inside \end{document}, if no such \clearpage ended the document body. Thus \AfterLastShipout comes to the rescue, from package atveryend.

```
\AfterLastShipout
    {\immediate\write\@auxout{\string\@writefile{toc}{\string\tof@finish}}}
```

1.4a makes the things more easily patchable by other packages, especially I have **etoc** in mind.

\tof@@starttags and \tof@@stoptags are defined a bit later in the code. And we use \tof@global. And we need \tof@@finish to reset \tof@tags to empty, just in case \tof@global was \global. And also \contentsline !! But here it is a problem redefining it globally. In normal use of **tableof**, nothing needs to be done, because the \tof@endgroup closes the scope. In global mode (triggered only by **etoc** with \etocglobaldefs for tabulars), it needed to globally modify \contentsline, thus it or **etoc** has to globally unmodify it. **etoc** 1.08 by itself without **tableof** doesn't have to reset \contentsline because it re-defined it *before* an eventual tabular. But as it forced **tableof** to do global things, in such case **etoc** will have to proceed globally as well.

```
\let\tof@global     \@empty
\let\tof@begingroup \begingroup
\let\tof@endgroup   \endgroup
```

No more \tof@endgroup here at 1.4c. See above explanations.

```
\def\tof@@finish   {\tof@global\let\contentsline\tof@savedcontentsline
                    \global\let\tof@begin\relax
                    \global\let\tof@tags\@empty }
\def\tof@@tagthis   #1{\def\tof@tags@tmp{#1}}
\def\tof@@untagthis #1{\def\tof@untags@tmp{#1}}
```

No more \tof@begingroup here at 1.4c. See above explanations.

```
\def\tof@init#1{%
    \def\tof@begin{%
        \tof@global\let\tof@tagthis  \tof@@tagthis
        \tof@global\let\tof@untagthis\tof@@untagthis
        \tof@global\let\tof@starttags\tof@@starttags
        \tof@global\let\tof@stoptags \tof@@stoptags
        \tof@global\let\tof@finish   \tof@@finish
        \tof@global\let\tof@savedcontentsline\contentsline
        \tof@global\def\contentsline {#1}}}
```

place holder for comments

```
\newcommand\tofOpenTocFileForWrite{%
  \if@filesw
   \newwrite \tf@toc
   \immediate \openout \tf@toc \jobname.toc\relax
  \fi}
```

Creating our booleans is not the most economical, (one could have used a single list macro) but heck, how many tags are there going to be anyhow in normal use? a dozen at the very most I think.

1.4a: fortunately no need for the \tof@global thing! However, careful with \tof@tags it will have to be reset to empty by \tof@finish in case \tof@global is \global.

```
\let\tof@tags      \@empty
\let\tof@tags@tmp  \@empty
\let\tof@untags@tmp\@empty
\def\tof@true #1{\expandafter\let\csname tofsw@#1\endcsname\iftrue}
```

```
\def\tof@false#1{\expandafter\let\csname tofsw@#1\endcsname\iffalse}
\def\tof@secondiftrue#1%
   {\csname tofsw@#1\endcsname \let\tof@next\@secondoftwo\fi}
\def\tof@secondiffalse#1%
   {\csname tofsw@#1\endcsname\else\let\tof@next\@secondoftwo\fi}
```

1.4a: fortunately no need for the `\tof@global` thing; the `\tof@setflags` is used at *each* `\contentsline`. Quite useful design, congrats to the original architect !

```
\def\tof@setflags #1{\let\tof@next\@firstoftwo
        \@for\@tempa:=#1\do             {\tof@true {\@tempa}}%
        \@for\@tempa:=\tof@tags\do      {\tof@false{\@tempa}}%
        \@for\@tempa:=\tof@tags@tmp\do  {\tof@false{\@tempa}}%
        \@for\@tempa:=\tof@untags@tmp\do{\tof@true {\@tempa}}}
```

Release 1.4 uses here a bunch of `\expandafter`'s in place of some `\edef`'s.

```
\def\tof@filter#1#2{\ifx#1#2\else
    \ifx\tof@tmptags\@empty
         \expandafter\def\expandafter\tof@tmptags\expandafter{#2}%
    \else
         \expandafter\expandafter\expandafter\def
         \expandafter\expandafter\expandafter\tof@tmptags
         \expandafter\expandafter\expandafter{\expandafter
         \tof@tmptags\expandafter,#2}%
    \fi\fi}
```

1.4a needs the `\tof@global` in the hope for **tableof** to be compatible with future **etoc** 1.08 when employed in the delicate art of TOC as a table !

```
\def\tof@@starttags#1{%
    \ifx\tof@tags\@empty
      \tof@global\def\tof@tags{#1}%
    \else
      \tof@global
      \expandafter\def\expandafter\tof@tags\expandafter{\tof@tags,#1}%
    \fi }
\def\tof@@stoptags#1{%
    \@for\@tempa:=#1\do{%
          \def\tof@tmptags{}%
          \@for\@tempb:=\tof@tags\do{\tof@filter\@tempa\@tempb}%
          \tof@global
          \expandafter\def\expandafter\tof@tags\expandafter{\tof@tmptags}%
         }%
    }
```

Until 1.3, I was just using the `\@for` thing with no attempt at any extra parsing, for spaces in particular. With 1.3 of 2015/02/11 I use an `\edef` to remove all spaces first, using a `\zapspaces` macro pioneered in `xintkernel.sty`. Thus, comma separated lists of tags can have spaces. They will be removed if not protected by braces.

```
\def\tof@zapspaces #1 #2{#1#2\tof@zapspaces }%
\def\tof@cleanspaces #1#2{\edef\tof@tmp {{\tof@zapspaces #2 \@gobble}}%
    \expandafter #1\tof@tmp }
```

placeholder for comments

```
\def\tof@and  #1{%
    \tof@init{\tof@setflags{#1}\def\tof@tags@tmp{}\def\tof@untags@tmp{}%
        \@for\@tempa:=#1\do{\tof@secondiftrue{\@tempa}}%
```

```
        \tof@next\tof@savedcontentsline\tof@gobblethree@orfour}%
    \the\tof@toctoks }
\def\tof@or   #1{%
    \tof@init{\tof@setflags{#1}\def\tof@tags@tmp{}\def\tof@untags@tmp{}%
        \@for\@tempa:=#1\do{\tof@secondiffalse{\@tempa}}%
        \tof@next\tof@gobblethree@orfour\tof@savedcontentsline}%
    \the\tof@toctoks }
\def\tof@nor  #1{%
    \tof@init{\tof@setflags{#1}\def\tof@tags@tmp{}\def\tof@untags@tmp{}%
        \@for\@tempa:=#1\do{\tof@secondiffalse{\@tempa}}%
        \tof@next\tof@savedcontentsline\tof@gobblethree@orfour}%
   \the\tof@toctoks }
\def\tof@nand #1{%
    \tof@init{\tof@setflags{#1}\def\tof@tags@tmp{}\def\tof@untags@tmp{}%
        \@for\@tempa:=#1\do{\tof@secondiftrue{\@tempa}}%
        \tof@next\tof@gobblethree@orfour\tof@savedcontentsline}%
    \the\tof@toctoks }
\newcommand*\tableof{\@ifstar{\tof@cleanspaces\tof@or}
                             {\tof@cleanspaces\tof@and}}
\newcommand*\tablenotof{\@ifstar{\tof@cleanspaces\tof@nand}
                                {\tof@cleanspaces\tof@nor}}
```

placeholder for comments

```
\def\tof@nextof@or #1{\toks@{\tof@setflags{#1}%
        \@for\@tempa:=#1\do{\tof@secondiffalse{\@tempa}}%
        \tof@next
        {\def\tof@tags@tmp{}\def\tof@untags@tmp{}\tof@gobblethree@orfour}}%
        \@ifstar{\tof@cleanspaces\tof@nextof@nand}
                {\tof@cleanspaces\tof@nextof@nor}}
\def\tof@nextof@and #1{\toks@{\tof@setflags{#1}%
             \@for\@tempa:=#1\do{\tof@secondiftrue{\@tempa}}%
             \tof@next\@secondoftwo\@firstoftwo
             {\def\tof@tags@tmp{}\def\tof@untags@tmp{}\tof@gobblethree@orfour}}%
        \@ifstar{\tof@cleanspaces\tof@nextof@nand}
                {\tof@cleanspaces\tof@nextof@nor}}
\def\tof@nextof@nor #1{%
       \toks@\expandafter{\the\toks@
             {\tof@setflags{#1}\def\tof@tags@tmp{}\def\tof@untags@tmp{}%
              \@for\@tempa:=#1\do{\tof@secondiffalse{\@tempa}}%
              \tof@next\tof@savedcontentsline\tof@gobblethree@orfour}}%
       \expandafter\tof@init\expandafter{\the\toks@}%
       \tof@printtoc }
\def\tof@nextof@nand #1{%
       \toks@\expandafter{\the\toks@
             {\tof@setflags{#1}\def\tof@tags@tmp{}\def\tof@untags@tmp{}%
              \@for\@tempa:=#1\do{\tof@secondiftrue{\@tempa}}%
              \tof@next\tof@gobblethree@orfour\tof@savedcontentsline}}%
       \expandafter\tof@init\expandafter{\the\toks@}%
       \tof@printtoc }
\newcommand*{\nexttocwithtags}{\let\tof@printtoc\relax
             \@ifstar{\tof@cleanspaces\tof@nextof@or}
                     {\tof@cleanspaces\tof@nextof@and}}
\newcommand*{\tableoftaggedcontents}{\def\tof@printtoc{\the\tof@toctoks}%
             \@ifstar{\tof@cleanspaces\tof@nextof@or}
```

```
                    {\tof@cleanspaces\tof@nextof@and}}
```

placeholder for comments

```
\newcommand*\toftagthis[1]
  {\addtocontents{toc}{\string\tof@tagthis  {\tof@zapspaces #1 \@gobble }}}
\newcommand*\tofuntagthis[1]
  {\addtocontents{toc}{\string\tof@untagthis{\tof@zapspaces #1 \@gobble }}}
\newcommand*\toftagstart[1]
  {\addtocontents{toc}{\string\tof@starttags{\tof@zapspaces #1 \@gobble }}}
\newcommand*\toftagstop[1]
  {\addtocontents{toc}{\string\tof@stoptags {\tof@zapspaces #1 \@gobble }}}
\endinput
```