

# The **tableof** package

JEAN-FRAN OIS BURNOL  
jfbu (at) free (dot) fr  
Package version: v1.4b (2018/10/02)  
Documentation generated from the source file  
with timestamp “02-10-2018 at 11:43:43 CEST”

## Abstract

The commands `\toftagstart`, `\toftagstop`, `\toftagthis`, `\torefntagthis` are used to tag chapters, sections or any other sectioning units destined to end up in the table(s) of contents. Then:

```
\nexttocwithtags{req. 1, req. 2, ...}{excl. 1, excl. 2, ...}  
\tableofcontents % or equivalent command
```

specifies which tags are to be required and which ones are to be excluded from the typeset TOC.

For documents with classes where `\tableofcontents` is only single-use, the package provides:

```
\tableoftaggedcontents{req. 1, req. 2, ...}{excl. 1, excl. 2, ...}
```

which does not have this restriction.

## Contents

<b>1 Tagging commands</b>	<b>1</b>
1.1 <code>\toftagstart {tag1, tag2, ...}</code> and <code>\toftagstop {tag1, tag2, ...}</code>	1
1.2 <code>\toftagthis {tag1, tag2, ...}</code> and <code>\torefntagthis {tag1, tag2, ...}</code>	1
1.3 Spaces	2
<b>2 Table of contents commands</b>	<b>2</b>
2.1 <code>\nexttocwithtags {required tag1, required tag2, ...}{excluded tag1, excluded tag2, ...}</code>	2
2.1.1 starred variants	2
2.2 <code>\tableoftaggedcontents {required tag1, required tag2, ...}{excluded tag1, excluded tag2, ... }</code>	3
2.2.1 starred variants	3
2.3 <code>\tableof {required tag1, required tag2, ...}</code>	3
2.4 <code>\tablenotof {excluded tag1, excluded tag2, ...}</code>	3
2.5 <code>\topenTocFileForWrite</code>	3
<b>3 Compatibility with other packages</b>	<b>4</b>
<b>4 Version History</b>	<b>4</b>
<b>5 Generating the package file and the test file</b>	<b>5</b>
<b>6 Implementation</b>	<b>5</b>

## 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 \torefntagthis {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 \torefntagthis command similarly untags only the *next* entry. Similarly, only the tags in the last call of \torefntagthis are taken into account.

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

\torefntagthis{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. (*new with release 1.3*). 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 explicitly 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}`.<sup>1</sup> 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** checks if `hyperref` is loaded as `hyperref` modifies the format of the lines in the .toc file (and this must be taken into account). This check is done at the `\begin{document}` so the order of loading is not important.

---

<sup>1</sup>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** 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.

**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**.<sup>2</sup> It went fine.

## 4 Version History

- 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 (cf. <https://tex.stackexchange.com/q/358014/4686>). The package atveryend is now required.<sup>3</sup>
- 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 (some \edef's replaced by a bunch of \expandafter's.)
  - 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.
  - ii. .toc may be input in another document not loading **tableof**.
- 2012/12/06 v1.0: initial version.

---

<sup>2</sup><http://www.ctan.org/pkg/etoc>

<sup>3</sup><http://www.ctan.org/pkg/atveryend>

## 5 Generating the package file and the test file

Running `latex` on `tableof.dtx` generates: 1) the package file `tableof.sty` (which should be moved to a suitable location within the  $\text{\TeX}$  installation), 2) `tableof.ins` for  $\text{\TeX}$  distributions expecting such a file, 3) `tableoftest.tex` which demonstrates the package features (one should run `latex` twice on it), and 4) the documentation itself. After running `latex` twice, use `dvipdfmx` to produce the pdf documentation.

## 6 Implementation

```

1 \ProvidesPackage{tableof}
2 [2018/10/02 v1.4b Tables of tagged contents (JFB)]
3 \NeedsTeXFormat{LaTeX2e}
4 \RequirePackage{atveryend}
5 \DeclareOption*{\PackageWarning{tableof}{Option '\CurrentOption' is unknown.}}
6 \ProcessOptions\relax
7 \newtoks\totoks@tots

```

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

```

8 \def\totoks@readtoc {%
9   \ifeof \totoks@tf
10  \else
11    \read\totoks@tf to \totoks@buffer
12    \totoks@expandafter\expandafter\expandafter
13      {\expandafter\the\expandafter\totoks@tots\totoks@buffer}%
14    \expandafter\totoks@readtoc
15  \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.

```

16 \AtBeginDocument{\IfFileExists{\jobname.toc}%
17   {{\endlinechar\m@ne
18     \makeatletter
19     \newread\totoks@tf
20     \openin\totoks@tf@filef@und
21     \totoks@readtoc
22     \global\totoks@tots=\expandafter{\the\totoks@tots}%
23     \closein\totoks@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 `\totoks@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  $\text{\TeX}$ 's macros.

1.4a uses `\empty` for the default `\totoks@finish` rather than `\relax`, for no special reason. Also the `\let`'s in case `\totoks@begin` is undefined or `\relax` are all made global,

## 6 Implementation

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.

```
24 \AtBeginDocument{  
25     \addtocontents{toc}{\string\@ifundefined{tof@begin}{%  
26         \global\let\string\tof@begin\relax  
27         \global\let\string\tof@finish\string\@empty  
28         \global\let\string\tof@starttags\string\@gobble  
29         \global\let\string\tof@stoptags\string\@gobble  
30         \global\let\string\tof@tagthis\string\@gobble  
31         \global\let\string\tof@untagthis\string\@gobble}{}}  
32     \addtocontents{toc}{\string\tof@begin}  
33     \@ifpackageloaded{hyperref}{%  
34         \def\tof@gobblethree@orfour#1#2#3#4{}%  
35         \ifx\hyper@last\@undefined\toc@toctoks{}\fi  
36     \def\tof@gobblethree@orfour#1#2#3{}%}
```

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

```
\AtEndDocument{\addtocontents{toc}{\string\tobefinish}}
```

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`.

```
37 \AfterLastShipout{  
38     {\immediate\write\auxout{\string\@writefile{toc}{\string\tobefinish}}}}
```

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

`\tobefinish` and `\tobefinish` are defined a bit later in the code. And we use `\tobegingroup`. And we need `\tobefinish` to reset `\tobefinish` to empty, just in case `\tobefinish` 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 `\tobefinish` 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.

```
39 \let\tobefinish\empty  
40 \let\tobefinish\begin{group}  
41 \let\tobefinish\end{group}  
42 \def\tobefinish{\tobefinish\let\contentsline\tobefinish  
43             \tobefinish\global\let\tobefinish\relax  
44             \global\let\tobefinish\empty}  
45 \def\tobefinish#1{\def\tobefinish{\tobefinish#1}}  
46 \def\tobefinish#1{\def\tobefinish{\tobefinish#1}}  
47 \def\tobefinish{\tobefinish  
48     \def\tobefinish{\tobefinish  
49         \tobefinish\begin{group}  
50         \tobefinish\let\tobefinish\tobefinish\tobefinish  
51         \tobefinish\let\tobefinish\tobefinish\tobefinish  
52         \tobefinish\let\tobefinish\tobefinish\tobefinish  
53         \tobefinish\let\tobefinish\tobefinish\tobefinish}
```

## 6 Implementation

```

54      \t@f@global\let\t@f@finish  \t@f@@finish
55      \t@f@global\let\t@f@savedcontentsline\contentsline
56      \t@f@global\def\contentsline {\#1}}}

```

place holder for comments

```

57 \newcommand\t@fOpenTocFileForWrite{%
58   \if@filesw
59     \newwrite \tf@toc
60     \immediate \openout \tf@toc \jobname.toc\relax
61   \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 `\t@f@global` thing! However, careful with `\t@f@tags` it will have to be reset to empty by `\t@f@finish` in case `\t@f@global` is `\global`.

```

62 \let\t@f@tags      \@empty
63 \let\t@f@tags@tmp  \@empty
64 \let\t@f@untags@tmp@empty
65 \def\t@f@true #1{\expandafter\let\csname tofsw@#1\endcsname\iftrue}
66 \def\t@f@false#1{\expandafter\let\csname tofsw@#1\endcsname\iffalse}
67 \def\t@f@secondiftrue#1%
68   {\csname tofsw@#1\endcsname \let\t@f@next@\secondoftwo\fi}
69 \def\t@f@secondiffalse#1%
70   {\csname tofsw@#1\endcsname\else\let\t@f@next@\secondoftwo\fi}

```

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

```

71 \def\t@f@setflags #1{\let\t@f@next@\firstoftwo
72   @for@tempa:=#1\do          {\t@f@true {\@tempa}}%
73   @for@tempa:=\t@f@tags\do    {\t@f@false{\@tempa}}%
74   @for@tempa:=\t@f@tags@tmp\do {\t@f@false{\@tempa}}%
75   @for@tempa:=\t@f@untags@tmp\do{\t@f@true {\@tempa}}}

```

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

```

76 \def\t@f@filter#1#2{\ifx#1#2\else
77   \ifx\t@f@tmptags@\empty
78     \expandafter\def\expandafter\t@f@tmptags\expandafter{#2}%
79   \else
80     \expandafter\expandafter\expandafter\def
81     \expandafter\expandafter\expandafter\t@f@tmptags
82     \expandafter\expandafter\expandafter{\expandafter
83       \t@f@tmptags\expandafter,\#2}%
84   \fi\fi}

```

1.4a needs the `\t@f@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 !

```

85 \def\t@f@starttags#1{%
86   \ifx\t@f@tags\empty
87     \t@f@global\def\t@f@tags{\#1}%
88   \else
89     \t@f@global
90     \expandafter\def\expandafter\t@f@tags\expandafter{\t@f@tags,\#1}%
91   \fi }

```

## 6 Implementation

```

92 \def\tof@stoptags#1{%
93   \@for@tempa:=#1\do{%
94     \def\tof@tmptags{}%
95     \@for@tempb:=\tof@tags\do{\tof@filter@\tempa@\tempb}%
96     \tof@global
97     \expandafter\def\expandafter\tof@tags\expandafter{\tof@tmptags}%
98   }%
99 }

```

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.

```

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

placeholder for comments

103 \def\tof@and #1{%
104   \tof@init{\tof@setflags{#1}\def\tof@tags@tmp{}\def\tof@untags@tmp{}%
105     \@for@tempa:=#1\do{\tof@secondiftrue{\tempa}}%
106     \tof@next\tof@savedcontentsline\tof@gobblethree@orfour}%
107   \the\tof@toctoks }

108 \def\tof@or #1{%
109   \tof@init{\tof@setflags{#1}\def\tof@tags@tmp{}\def\tof@untags@tmp{}%
110     \@for@tempa:=#1\do{\tof@secondiffalse{\tempa}}%
111     \tof@next\tof@gobblethree@orfour\tof@savedcontentsline}%
112   \the\tof@toctoks }

113 \def\tof@nor #1{%
114   \tof@init{\tof@setflags{#1}\def\tof@tags@tmp{}\def\tof@untags@tmp{}%
115     \@for@tempa:=#1\do{\tof@secondiffalse{\tempa}}%
116     \tof@next\tof@savedcontentsline\tof@gobblethree@orfour}%
117   \the\tof@toctoks }

118 \def\tof@nand #1{%
119   \tof@init{\tof@setflags{#1}\def\tof@tags@tmp{}\def\tof@untags@tmp{}%
120     \@for@tempa:=#1\do{\tof@secondiftrue{\tempa}}%
121     \tof@next\tof@gobblethree@orfour\tof@savedcontentsline}%
122   \the\tof@toctoks }

123 \newcommand*\tableof{\@ifstar{\tof@cleanspaces\tof@or}%
124   {\tof@cleanspaces\tof@and}}
125 \newcommand*\tablenotof{\@ifstar{\tof@cleanspaces\tof@nand}%
126   {\tof@cleanspaces\tof@nor}}

placeholder for comments

127 \def\tof@nextof@or #1{\toks@{\tof@setflags{#1}}%
128   \@for@tempa:=#1\do{\tof@secondiffalse{\tempa}}%
129   \tof@next
130   {\def\tof@tags@tmp{}\def\tof@untags@tmp{}\tof@gobblethree@orfour}%
131   \ifstar{\tof@cleanspaces\tof@nextof@nand}%
132     {\tof@cleanspaces\tof@nextof@nor}}
133 \def\tof@nextof@and #1{\toks@{\tof@setflags{#1}}%
134   \@for@tempa:=#1\do{\tof@secondiftrue{\tempa}}%
135   \tof@next\secondeoftwo\firsofttwo%
136   {\def\tof@tags@tmp{}\def\tof@untags@tmp{}\tof@gobblethree@orfour}%

```

## 6 Implementation

```
137     \@ifstar{\tof@cleanspaces\tof@nextof@nand}
138             {\tof@cleanspaces\tof@nextof@nor}}
139 \def\tof@nextof@nor #1{%
140     \toks@\expandafter{\the\toks@
141         {\tof@setflags{#1}\def\tof@tags@tmp{}\def\tof@untags@tmp{}%
142             \@for\@tempa:=#1\do{\tof@secondiffalse{@tempa}}%
143             \tof@next\tof@savedcontentsline\tof@gobblethree@orfour}%
144     \expandafter\tof@init\expandafter{\the\toks@}%
145     \tof@printtoc }
146 \def\tof@nextof@nand #1{%
147     \toks@\expandafter{\the\toks@
148         {\tof@setflags{#1}\def\tof@tags@tmp{}\def\tof@untags@tmp{}%
149             \@for\@tempa:=#1\do{\tof@secondiftrue{@tempa}}%
150             \tof@next\tof@gobblethree@orfour\tof@savedcontentsline}%
151     \expandafter\tof@init\expandafter{\the\toks@}%
152     \tof@printtoc }
153 \newcommand*{\nexttocwithtags}{\let\tof@printtoc\relax
154     \@ifstar{\tof@cleanspaces\tof@nextof@or}
155             {\tof@cleanspaces\tof@nextof@and}}
156 \newcommand*{\tableoftaggedcontents}{\def\tof@printtoc{\the\tof@toctoks}%
157     \@ifstar{\tof@cleanspaces\tof@nextof@or}
158             {\tof@cleanspaces\tof@nextof@and}}
placeholder for comments
159 \newcommand*\toftagthis[1]
160   {\addtocontents{toc}{\string\tof@tagthis {\tof@zapspaces #1 \@gobble }}}
161 \newcommand*\tofuntagthis[1]
162   {\addtocontents{toc}{\string\tof@untagthis{\tof@zapspaces #1 \@gobble }}}
163 \newcommand*\toftagstart[1]
164   {\addtocontents{toc}{\string\tof@starttags{\tof@zapspaces #1 \@gobble }}}
165 \newcommand*\toftagstop[1]
166   {\addtocontents{toc}{\string\tof@stoptags {\tof@zapspaces #1 \@gobble }}}
167 \endinput
```