

# The **tableof** package

JEAN-FRANÇOIS BURNOL

jfbu (at) free (dot) fr

Package version: v1.3 (2015/02/11)

Documentation generated from the source file

with timestamp "11-02-2015 at 23:14:33 CET"

## Abstract

The commands `\toftagstart`, `\toftagstop`, `\toftagthis`, `\tofuntagthis` are used to tag chapters or sections. Then:

```
\nexttocwithtags{required1, requ.2, ...}{excluded1, 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 document with classes where `\tableofcontents` is only single-use, the package provides:

```
\tableoftaggedcontents{required1, requ.2, ...}{excluded1, excl.2, ...}
```

which does not have this restriction.

## Contents

<b>1</b>	<b>Tagging commands</b>	<b>2</b>
1.1	<code>\toftagstart {tag1, tag2, ...}</code> and <code>\toftagstop {tag1, tag2, ...}</code>	2
1.2	<code>\toftagthis {tag1, tag2, ...}</code> and <code>\tofuntagthis {tag1, tag2, ...}</code>	2
1.3	Spaces	2
<b>2</b>	<b>Table of contents commands</b>	<b>3</b>
2.1	<code>\nexttocwithtags {required tag1, required tag2, ...}{excluded tag1, excluded tag2, ...}</code>	3
2.1.1	starred variants	3
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>	4
2.5	<code>\tofOpenTocFileForWrite</code>	4
2.6	Compatibility with other packages	4
2.7	(old) TODO (never to be done ?)	4
2.8	Version History	5
<b>3</b>	<b>Generating the package file and the test file</b>	<b>5</b>
<b>4</b>	<b>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 `\tofuntagthis {tag1, tag2, ...}`

The `\toftagthis` command flags with the comma separated values from its list argument only the *next* sectioning command. The `\tofuntagthis` command similarly untags only the *next* entry.

```
\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. (*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 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> This is only of relevance if the `\tableoftaggedcontents`, `\tableof` or `\tablenotof` are made use of in the document.

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 can be used also in the preamble.

Again: 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`.

## 2.6 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.

`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 *crazy* interference of `biblatex` which adds commands to the `.toc` line on every other line (thus, potentially right in the middle of multi-line commands `:-C`).

No testing has actually been done of compatibility with packages manipulating tables of contents, apart from `etoc`.<sup>2</sup> It went fine.<sup>3</sup>

## 2.7 (old) TODO (never to be done ?)

Create a parser for arbitrary iterated boolean combinations of tags . . . but this kind of thing must have been done a zillion times already, as it belongs to the basics in computer science! ... and it surely is no easy task in  $\text{\TeX}$ ! so I cautiously retreated from re-doing something difficult which has already been done somewhere by someone ...

---

<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.

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

<sup>3</sup>but that was a long time ago...

## 2.8 Version History

- 2015/02/11 v1.3: i. comma separated tags now allow spaces (they are removed).  
 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.

## 3 Generating the package file and the test file

Running latex (or pdflatex) on tableof.dtx generates: 1) the package file tableof.sty (which should be moved to a suitable location within the T<sub>E</sub>X installation), 2) tableof.ins for T<sub>E</sub>X 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 (run pdflatex twice to get the bookmarks.)

## 4 Implementation

Writing-up source code comments is hopefully for a future release.

```

1 \ProvidesPackage{tableof}
2 [2015/02/11 v1.3 Tables of tagged contents (jfb)]
3 \NeedsTeXFormat{LaTeX2e}
4 \DeclareOption*{\PackageWarning{tableof}{Option ‘\CurrentOption’ is unknown.}}
5 \ProcessOptions\relax

6 \newtoks\tof@toctoks

1.3 codes this \tof@readtoc slightly better (copied from etoc.dtx).
7 \def\tof@readtoc {%
8   \ifeof \tof@tf
9   \else
10    \read\tof@tf to \tof@buffer
11    \tof@toctoks\expandafter\expandafter\expandafter
12      {\expandafter\the\expandafter\tof@toctoks\tof@buffer}%
13    \expandafter\tof@readtoc
14  \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

## 4 Implementation

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.

```

15 \AtBeginDocument{\IfFileExists{\jobname.toc}
16   {\endlinechar-1
17    \makeatletter
18    \newread\tof@tf
19    \openin\tof@tf\@filef@und
20    \tof@readtoc
21    \global\tof@toctoks=\expandafter{\the\tof@toctoks}%
22    \closein\tof@tf}}{}

```

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.

```

23 \AtBeginDocument{
24   \addtocontents{toc}{\string\@ifundefined{tof@begin}%
25    {\let\string\tof@begin\relax
26     \let\string\tof@finish\relax
27     \let\string\tof@starttags\string@gobble
28     \let\string\tof@stoptags\string@gobble
29     \let\string\tof@tagthis\string@gobble
30     \let\string\tof@untagthis\string@gobble}}{}
31   \addtocontents{toc}{\string\tof@begin}
32   \ifpackageloaded{hyperref}
33     {\def\tof@gobblethree@orfour#1#2#3#4{}%
34      \ifx\hyper@last\@undefined\tof@toctoks{}\fi}
35     {\def\tof@gobblethree@orfour#1#2#3{}%
36 \AtEndDocument{\addtocontents{toc}{\string\tof@finish}}
37 \newcommand\tofOpenTocFileForWrite{%
38   \if@filesw
39   \newwrite \tf@toc
40   \immediate \openout \tf@toc \jobname.toc\relax
41   \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.

```

42 \def\tof@tags{}
43 \def\tof@tags@tmp{}
44 \def\tof@untags@tmp{}
45 \def\tof@true #1{\expandafter\let\csname tofsw@#1\endcsname\iftrue}
46 \def\tof@false#1{\expandafter\let\csname tofsw@#1\endcsname\iffalse}
47 \def\tof@secondiftrue#1%
48   {\csname tofsw@#1\endcsname \let\tof@next\@secondoftwo\fi}
49 \def\tof@secondiffalse#1%
50   {\csname tofsw@#1\endcsname\else\let\tof@next\@secondoftwo\fi}
51 \def\tof@setflags #1{\let\tof@next\@firstoftwo
52   \@for\@tempa:=#1\do {\tof@true {\@tempa}}%
53   \@for\@tempa:=\tof@tags\do {\tof@false{\@tempa}}%
54   \@for\@tempa:=\tof@tags@tmp\do {\tof@false{\@tempa}}%
55   \@for\@tempa:=\tof@untags@tmp\do{\tof@true {\@tempa}}}

```

## 4 Implementation

```

56 \def\tof@filter#1#2{\ifx#1#2\else
57     \ifx\tof@tmptags\@empty\edef\tof@tmptags{#2}\else
58     \edef\tof@tmptags{\tof@tmptags,#2}\fi\fi}
59 \def\tof@starttags#1{%
60     \ifx\tof@tags\@empty\edef\tof@tags{#1}\else
61     \edef\tof@tags{\tof@tags,#1}\fi}
62 \def\tof@stoptags#1{%
63     \@for\@tempa:=#1\do{\def\tof@tmptags{}}%
64     \@for\@tempb:=\tof@tags\do{\tof@filter\@tempa\@tempb}%
65     \edef\tof@tags{\tof@tmptags}}
66 \def\tof@init#1{%
67     \def\tof@begin{%
68         \begingroup
69         \def\tof@finish{\endgroup\global\let\tof@begin\relax}%
70         \let\tof@savedcontentsline\contentsline
71         \let\tof@starttags\tof@starttags
72         \let\tof@stoptags\tof@stoptags
73         \def\tof@tagthis ####1{\def\tof@tags@tmp{####1}}%
74         \def\tof@untagthis ####1{\def\tof@untags@tmp{####1}}%
75         \def\contentsline {#1}}

```

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.

```

76 \def\tof@zapspaces #1 #2{#1#2\tof@zapspaces }%
77 \def\tof@cleanspaces #1#2{\edef\tof@tmp {\tof@zapspaces #2 \@gobble}}%
78     \expandafter #1\tof@tmp }
79 \def\tof@and #1{%
80     \tof@init{\tof@setflags{#1}\def\tof@tags@tmp{\def\tof@untags@tmp{}}%
81     \@for\@tempa:=#1\do{\tof@secondifftrue{\@tempa}}%
82     \tof@next\tof@savedcontentsline\tof@gobblethree@orfour}%
83     \the\tof@toctoks }
84 \def\tof@or #1{%
85     \tof@init{\tof@setflags{#1}\def\tof@tags@tmp{\def\tof@untags@tmp{}}%
86     \@for\@tempa:=#1\do{\tof@secondiffalse{\@tempa}}%
87     \tof@next\tof@gobblethree@orfour\tof@savedcontentsline}%
88     \the\tof@toctoks }
89 \def\tof@nor #1{%
90     \tof@init{\tof@setflags{#1}\def\tof@tags@tmp{\def\tof@untags@tmp{}}%
91     \@for\@tempa:=#1\do{\tof@secondiffalse{\@tempa}}%
92     \tof@next\tof@savedcontentsline\tof@gobblethree@orfour}%
93     \the\tof@toctoks }
94 \def\tof@nand #1{%
95     \tof@init{\tof@setflags{#1}\def\tof@tags@tmp{\def\tof@untags@tmp{}}%
96     \@for\@tempa:=#1\do{\tof@secondifftrue{\@tempa}}%
97     \tof@next\tof@gobblethree@orfour\tof@savedcontentsline}%
98     \the\tof@toctoks }
99 \newcommand*\tableof{\@ifstar{\tof@cleanspaces\tof@or}
100     {\tof@cleanspaces\tof@and}}
101 \newcommand*\tablenotof{\@ifstar{\tof@cleanspaces\tof@nand}
102     {\tof@cleanspaces\tof@nor}}

```

## 4 Implementation

```

103 \def\tof@nextof@or #1{\toks@{\tof@setflags{#1}%
104     \@for\@tempa:=#1\do{\tof@secondiffalse{\@tempa}}}%
105     \tof@next
106     {\def\tof@tags@tmp{}\def\tof@untags@tmp{}\tof@gobblethree@orfour}}%
107     \@ifstar{\tof@cleanspaces\tof@nextof@nand}
108         {\tof@cleanspaces\tof@nextof@nor}}
109 \def\tof@nextof@and #1{\toks@{\tof@setflags{#1}%
110     \@for\@tempa:=#1\do{\tof@secondifftrue{\@tempa}}}%
111     \tof@next\@secondoftwo\@firstoftwo
112     {\def\tof@tags@tmp{}\def\tof@untags@tmp{}\tof@gobblethree@orfour}}%
113     \@ifstar{\tof@cleanspaces\tof@nextof@nand}
114         {\tof@cleanspaces\tof@nextof@nor}}
115 \def\tof@nextof@nor #1{%
116     \toks@\expandafter{\the\toks@
117         {\tof@setflags{#1}\def\tof@tags@tmp{}\def\tof@untags@tmp{}%
118             \@for\@tempa:=#1\do{\tof@secondiffalse{\@tempa}}}%
119             \tof@next\tof@saveditocline\tof@gobblethree@orfour}}}%
120     \expandafter\tof@init\expandafter{\the\toks@}%
121     \tof@printtoc }
122 \def\tof@nextof@nand #1{%
123     \toks@\expandafter{\the\toks@
124         {\tof@setflags{#1}\def\tof@tags@tmp{}\def\tof@untags@tmp{}%
125             \@for\@tempa:=#1\do{\tof@secondifftrue{\@tempa}}}%
126             \tof@next\tof@gobblethree@orfour\tof@saveditocline}}}%
127     \expandafter\tof@init\expandafter{\the\toks@}%
128     \tof@printtoc }
129 \newcommand*\nexttocwithtags{\let\tof@printtoc\relax
130     \@ifstar{\tof@cleanspaces\tof@nextof@or}
131         {\tof@cleanspaces\tof@nextof@and}}
132 \newcommand*\tableoftaggedcontents{\def\tof@printtoc{\the\tof@toctoks}%
133     \@ifstar{\tof@cleanspaces\tof@nextof@or}
134         {\tof@cleanspaces\tof@nextof@and}}

135 \newcommand*\toftagthis[1]
136     {\addtocontents{toc}{\string\tof@tagthis {\tof@zapspace #1 \@gobble }}}
137 \newcommand*\tofuntagthis[1]
138     {\addtocontents{toc}{\string\tof@untagthis{\tof@zapspace #1 \@gobble }}}
139 \newcommand*\toftagstart[1]
140     {\addtocontents{toc}{\string\tof@starttags{\tof@zapspace #1 \@gobble }}}
141 \newcommand*\toftagstop[1]
142     {\addtocontents{toc}{\string\tof@stoptags {\tof@zapspace #1 \@gobble }}}
143 \endinput

```