

The `tocloft` package^{*}

Author: Peter Wilson, Herries Press
Maintainer: L^AT_EX Project, Will Robertson

2026/08/12

Abstract

The `tocloft` package provides means of controlling the typographic design of the Table of Contents, List of Figures and List of Tables. New kinds of ‘List of ...’ can be defined.

The package has been tested with the `tocbibind`, `minitoc`, `ccaption`, `subfigure`, `float`, `fncychap`, and `hyperref` packages.

Contents

1	Introduction	2
1.1	Compatibility with <code>\DocumentMetadata</code> and the tagging support	2
1.2	Legacy L ^A T _E X’s methods	3
2	The <code>tocloft</code> package	7
2.1	Package options	7
2.2	Changing the titles	7
2.3	Typesetting the entries	8
2.4	New list of...	13
2.5	Experimental utilities	16
2.6	Usage with other packages	17
3	The package code	18
3.1	Support for the <code>subfigure</code> package	42
3.2	New list of...	45
3.3	Switching page numbering	50
3.4	Experimental utilities	52
	Index	56

List of Figures

1	Layout of a ToC (LoF, LoT) entry	4
---	--	---

^{*}This file (`tocloft.dtx`) has version number v3.0a, last revised 2026/08/12. Issues can be reported at <https://github.com/LaTeX-Package-Repositories/herries-press>

List of Tables

1	Indents and Numwidths	5
---	-----------------------	---

1 Introduction

In the standard classes the typographic design of the Table of Contents (ToC), the List of Figures (LoF) and List of Tables (LoT) is fixed or, more precisely, it is buried within the class definitions. The `tocloft` package provides handles for an author to change the design to meet the needs of the particular document.

Elements of the package were developed as part of a class and package bundle for typesetting ISO standards [Wil96b]. This manual is typeset according to the conventions of the $\text{\LaTeX}$ `DOCSTRIP` utility which enables the automatic extraction of the $\text{\LaTeX}$ macro source files [GMS94].

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

The package has been tested in combination with at least the `tocbibind` package [Wil00], the `minitoc` package [Dru99], the `ccaption` package [Wil01], the `subfigure` package [Coc95] (versions 2.0 and 2.1), the `algorithm` package [Wil96a] (which, in turn, calls the `float` package [Lin95]) and the `fncychap` package [Lin97]. It also works with the `hyperref` package. Please send me any comments as to how you think that the package can be improved, or of any interesting examples of how you have used it.¹

The package is not compatible with the KOMA-class. The memoir class suppresses the package and emulates it.

Version 3 adds support for `\DocumentMetadata` and tagging. It requires a current $\text{\LaTeX}$ format. The previous version can be loaded with

```
\usepackage{tocloft}[v2]
```

1.1 Compatibility with `\DocumentMetadata` and the tagging support

To enable tagging of the various lists like the Table of Contents, the code loaded by `\DocumentMetadata` in $\text{\LaTeX}$ since 2025-11-01 changes various internals of the ToC processing and this clashes with the changes made by `tocloft` in various ways: In some cases `\DocumentMetadata` overwrites changes by `tocloft` and destroys its features, in other `tocloft` destroys tagging.

Version 3 of the package tries to make the `tocloft` package compatible with `\DocumentMetadata`. It is recommended to use this together with a current $\text{\LaTeX}$.

Note that the code loaded by `\DocumentMetadata` is still changing. At some time ToC and similar lists will be reimplemented with templates and then more adaptations will be needed to keep `tocloft` compatible.

The following problems are handled by the new version if `\DocumentMetadata` is detected:

¹Thanks to Rowland (rebecca@astrid.u-net.com), John Foster (john@isjf.demon.co.uk), Kasper (kbg@dkik.dk), Lee Nave (nave@math.washington.edu), and Andrew Thurber (athurber@emba.uvm.edu) for their suggestions.

`\@starttoc` `tocloft` patches the `\@starttoc` command to insert hooks like `\cfttocprehook` and `\cfttocposthook` but it does it only if `\@starttoc` has its original definition. As `\DocumentMetadata` changes `\@starttoc` the patch is lost. The new code reinstates the patch. Note that the hooks are before and after (outside) the tagging sockets that create the TOC-structure. They should not contain code that creates structures that are not allowed in this place.

Tagging of list entries In standard L^AT_EX the tagging support for the list entries is mostly handled through a redefinition of `\@dottedtocline`. As `tocloft` doesn't use `\@dottedtocline` (and also overwrites `\l@part` and `\l@chapter`) none of the entries were correctly tagged. Version 3.0 now add hooks and tagging sockets to all `\l@X` (`\l@part`, `\l@chapter` etc), to all `\cftXfillnum` commands and to the `\numberline` command to enable the tagging support.

Tagging of list headings If the package option `titles` is used, `tocloft` doesn't redefine the list commands and the list headings are tagged as in standard L^AT_EX. The various commands offered by the package to configure the look of the heading then do nothing.

If the package option `titles` is *not* used, `tocloft` doesn't use heading commands to typeset the headings of lists, instead it uses a combination of spacing and font commands to get the right look and the headings are not properly tagged as headings.

There are theoretically two options to add the proper tagging support: suitable structures could be added manually, e.g. with a tagging socket, or the headings could be typeset with standard heading commands always. The new code uses the second approach and passes the various `tocloft` commands as keys to the heading commands to emulate the configuration options.

This means that the headings can be configured by using the legacy `tocloft` commands, but also react (to some extent) if the template instances are edited. It also leads to small incompatible changes: As a heading command finishes with a `\par` the command `\cftafterZtitle` should not start with a newline command (and if used together with tagging, the tagging should be carefully checked).

1.2 Legacy L^AT_EX's methods

This is a general description of how L^AT_EX does the processing for a Table of Contents. As the processing for List of Figures and List of Tables is similar I will, without loss of generality, just discuss the ToC.

`\addcontentsline` L^AT_EX generates a `.toc` file if the document contains a `\tableofcontents` command. The sectioning commands² put entries into the `.toc` file by calling the L^AT_EX `\addcontentsline{<file>}{<kind>}{<title>}` command, where `<file>` is the file extension (e.g., `toc`), `<kind>` is the kind of entry (e.g., `section` or `subsection`), and `<title>` is the (numbered) title text. In the cases where there is a number, the `<title>` argument is given in the form `{\numberline{number} title-text}`.

NOTE: The `hyperref` package dislikes authors using `\addcontentsline`. To get it to work properly with `hyperref` you normally have to put `\phantomsection` (a macro defined within the `hyperref` package) immediately before `\addcontentsline`.

`\contentsline` The `\addcontentsline` command writes an entry to the given file in the form

²For figures and tables it is the `\caption` command that populates the `.lof` and `.lot` files.

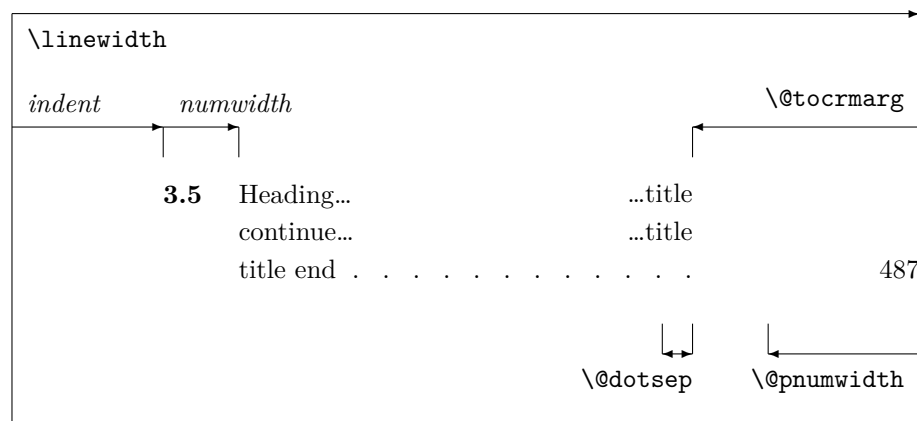


Figure 1: Layout of a ToC (LoF, LoT) entry

`\contentsline{<kind>}{<title>}{<page>}{<target>}` where `<page>` is the page number and `<destination>` is the anchor name (if `hyperref` is used). For each `<kind>`, `LATEX` provides a command `\l@kind{<title>}{<page>}` which performs the actual typesetting of the `\contentsline` entry.

The general layout of a typeset entry is illustrated in Figure 1. There are three internal `LATEX` commands that are used in the typesetting. The page number is typeset flushright in a box of width `\@pnumwidth`, and the box is at the righthand margin. If the page number is too long to fit into the box it will stick out into the righthand margin. The title text is indented from the righthand margin by an amount given by `\@tocrmarg`. Note that `\@tocrmarg` should be greater than `\@pnumwidth`. Some entries are typeset with a dotted leader between the end of the title text and the righthand margin indentation. The distance, in *math units*³ between the dots in the leader is given by the value of `\@dotsep`. In the standard classes the same values are used for the ToC, LoF and the LoT.

The standard values for these internal commands are:

- `\@pnumwidth = 1.55em`
- `\@tocrmarg = 2.55em`
- `\@dotsep = 4.5`

The values can be changed by using `\renewcommand`, in spite of the fact that the first two appear to be lengths.

Dotted leaders are not available for Part and Chapter ToC entries (nor for Section entries in the `article` class and its derivatives).

Each `\l@kind` macro is responsible for setting the general *indent* from the lefthand margin, and the *numwidth*. The `\numberline{<number>}` macro is responsible for typesetting the number flushleft in a box of width *numwidth*. If the number is too long for the box then it will protrude into the title text. The title text is indented by $(indent + numwidth)$ from the lefthand margin. That is, the title text is typeset in a block of width $(\linewidth - indent - numwidth - \@tocrmarg)$.

³There are 18mu to 1em.

Table 1: Indents and Numwidths (in ems)

Entry	Level	Chaptered		Otherwise	
		indent	numwidth	indent	numwidth
part	-1	0	—	0	—
chapter	0	0	1.5		
section	1	1.5	2.3	0	1.5
subsection	2	3.8	3.2	1.5	2.3
subsubsection	3	7.0	4.1	3.8	3.2
paragraph	4	10.0	5.0	7.0	4.1
subparagraph	5	12.0	6.0	10.0	5.0
figure/table	(1)	1.5	2.3	1.5	2.3

Table 1 lists the standard values for the *indent* and *numwidth*. There is no explicit *numwidth* for a part; instead a gap of 1em is put between the number and the title text. Note that for a sectioning command the values depend on whether or not the document class provides the `\chapter` command. Also, which somewhat surprises me, the table and figure entries are all indented.

`\@dottedtocline` Most of the `\l@kind` commands are defined in terms of the `\@dottedtocline` command. This command takes three arguments:

`\@dottedtocline{<seclevel>}{<indent>}{<numwidth>}`.

For example, one definition of the `\l@section` command is:

`\newcommand*{\l@section}{\@dottedtocline{1}{1.5em}{2.3em}}`

If it is necessary to change the default typesetting of the entries, then it is usually necessary to change these definitions (but the `tocloft` package gives you handles to easily alter things without having to know the L^AT_EX internals).

You can use the `\addcontentsline` command to add `\contentsline` commands to a file.

`\addtocontents` L^AT_EX also provides the `\addtocontents{<file>}{<text>}` command that will insert `<text>` into `<file>`. You can use this for adding extra text and/or macros into the file, for processing when the file is typeset by `\tableofcontents` (or whatever other command is used for `<file>` processing, such as `\listoftables` for a .lot file).

As `\addcontentsline` and `\addtocontents` write their arguments to a file, any fragile commands used in their arguments must be `\protected`.

You can make certain adjustments to the ToC etc., layout without using any package. Some examples are:

- If your page numbers stick out into the righthand margin

`\renewcommand{\@pnumwidth}{3em} \renewcommand{\@tocrmarg}{4em}`

but using lengths appropriate to your document.

- To have the (sectional) titles in the ToC, etc., typeset ragged right with no hyphenation

`\renewcommand{\@tocrmarg}{2.55em plus1fil}`

where the value 2.55em can be changed for whatever margin space you want.

- The dots in the leaders can be eliminated by increasing `\@dotsep` to a large value:

```
\renewcommand{\@dotsep}{10000}
```

- To have dotted leaders in your ToC and LoF but not in your LoT:

```
...
\tableofcontents
\makeatletter \renewcommand{\@dotsep}{10000} \makeatother
\listoftables
\makeatletter \renewcommand{\@dotsep}{4.5} \makeatother
\listoffigures
...
```

For this document I used this method to double the dot spacing for the LoF with respect to that for the ToC. As you can see, it is much better that all dot leaders have the same spacing.

- To add a horizontal line across the whole width of the ToC below an entry for a Part:

```
\part{Part title}
\addtocontents{toc}{\protect\mbox{\protect\hrulefill}\par}
```

Note that as both `\addtocontents` and `\addcontentsline` write their arguments to a file, it means that any *fragile* commands in their arguments must be protected by preceding each fragile command with `\protect`. The result of the example above would be the following two lines in the `.toc` file (assuming that it is the second Part and is on page 34):

```
\contentsline {part}{II\hspace {1em}Part title}{34}{}%
\mbox {} \hrulefill \par
```

If the `\protect`s were not used, then the second line would instead be:

```
\unhbox \voidb@x \hbox {} \unhbox \voidb@x \leaders \hrule \hfill \kern \z@ \par
```

- You may get undesired page breaks in the ToC. For example you may have a long multiline section title and in the ToC there is a page break between the lines. After your document is stable you can use `\addtocontents` at appropriate places in the body of the document to adjust the page breaking in the ToC. As examples:
 - `\addtocontents{toc}{\protect\newpage}` to force a page break.
 - `\addtocontents{toc}{\protect\enlargethispage{2\baselineskip}}` to make the page longer.
 - `\addtocontents{toc}{\protect\needspace{2\baselineskip}}` to specify that if there is not a vertical space of two baselines left on the page then start a new page (the `\needspace` macro is defined in the `needspace` package).

Remember, if you are modifying any command that includes an `@` sign then this must be done in either a `.sty` file or if in the document itself it must be surrounded by `\makeatletter` and `\makeatother`. For example, if you want to modify `\@dotsep` in the preamble to your document you have to do it like this:

```
\makeatletter
\renewcommand{\@dotsep}{9.0}
\makeatother
```

2 The `tocloft` package

The `tocloft` package provides means of specifying the typography of the Table of Contents (ToC), the List of Figures (LoF) and the List of Tables (LoT).

`\tableofcontents` The ToC, LoF, and LoT are printed at the point in the document where these commands are called, as per normal $\LaTeX$. However, there is one difference between the standard $\LaTeX$ behaviour and the behaviour with the `tocloft` package. In the standard $\LaTeX$ classes that have `\chapter` headings, the ToC, LoF and LoT each appear on a new page. With the `tocloft` package they do not necessarily start new pages; if you want them to be on new pages you may have to specifically issue an appropriate command beforehand. For example:

```
...
\clearpage
\tableofcontents
\clearpage
\listoftables
...
```

`\tocloftpagestyle` The `\thispagestyle` page style of the ToC, LoF and/or LoT is set by the command `\tocloftpagestyle{<style>}`, where `<style>` is one of the available page styles. The package initially sets `\tocloftpagestyle{plain}`.

2.1 Package options

The package takes the following options:

`subfigure` This option is required if, and only if, the `tocloft` and `subfigure` packages are being used together. The two packages can be specified in any order. Note that the `subfigure` is obsolete.

`titles` The `titles` option causes the titles of the ToC, LoF, and LoT lists to be typeset using the default $\LaTeX$ methods. This can be useful, for example, when the `tocloft` and `fncychap` packages are used together and the ‘fancy’ chapter styles should be used for the ToC, etc., titles.

If you use the `titles` option you can ignore the next section and continue reading at section 2.3.

2.2 Changing the titles

Commands are provided for controlling the appearance of the titles. Following $\LaTeX$ custom, the title texts are the values of the `\contentsname`, `\listfigurename` and `\listtablename` commands (stored inside `\cfttoccitlename`, `\cftloftitlename`, `\cftlottitlename`).

Similar sets of commands are provided for ToC, LoF and LoT title typesetting control. For convenience (certainly mine, and hopefully yours) in the following descriptions I will use Z to stand for ‘toc’ or ‘lof’ or ‘lot’. For example, `\cftmarkZ` stands for `\cftmarktoc` or `\cftmarklof` or `\cftmarklot`.

`\cftmarkZ` These macros set the appearance of the running heads on the ToC, LoF, and LoT pages. You probably don’t need to change these.

`\cftbeforeZtitleskip` These lengths control the vertical spacing before and after the titles. You can change
`\cftafterZtitleskip`

them from their default values by using `\setlength`.

`\cftZtitlefont` The code used for typesetting the ToC title looks like

`\cftZtitlename` `{\cfttoctitlefont \cftZtitlename}{\cftaftertoctitle}\par`

`\cftafterZtitle`

By default, `\cftZtitlefont` is defined as a font specification (e.g., `\Large\bfseries`), and `\cftafterZtitle` is empty. These commands can be changed (via `\renewcommand`) to change the typesetting. As examples:

- `\renewcommand{\cftZtitlefont}{\hfill\Large\itshape}` will result in a Large italic title typeset flushright.
- `\renewcommand{\cftZtitlefont}{\hfill\Large\bfseries}` together with `\renewcommand{\cftafterZtitle}{\hfill}` will give a centered Large bold title.
- Doing


```
\renewcommand{\cftafterZtitle}{%
  \[\baselineskip]\mbox{}\hfill{\normalfont Page}}
```

 will put the word ‘Page’ flushright on the line following the title. (If you do this, then you may need to decrease `\cftafterZtitleskip`). Note that if `\DocumentMetadata` is used, the command should not start with a `\`, as it is will be inserted after the heading command.
- `\renewcommand{\cftafterZtitle}{\thispagestyle{empty}}` will make the page with the title empty (i.e., the page number will not be printed).

`\cfttitlecommand` `\cfttitlecommand` is used to typeset the heading of a list if `\DocumentMetadata`

`\cftchapter` *is* used *and* the `titles` option is *not* used. It has one argument and by default it calls

`\cftsection` `\cftchapter` or `\cftsection` depending on if the class knows chapters or not.

2.3 Typesetting the entries

Commands are also provided to enable finer control over the typesetting of the different kinds of entries. The parameters defining the default layout of the entries are illustrated as part of the `layouts` package or in [GMS94, page 34], and are repeated in Figure 1.

`\Zdepth` The command `\Zdepth{<number>}` is analogous to the standard `\tocdepth{<number>}` command, in that it specifies that entries in the new listing should not be typeset if their numbering level is greater than `<number>`. The default definition is `\setcounter{Zdepth}{1}`. These commands are needed, for instance by users of packages such as `subcaption`, which will generate subfigure and subtable captions corresponding to a `lofdepth` and `lotdepth` of 2.

`\cftdot` In the default ToC typesetting only the more minor entries have dotted leader lines between the sectioning title and the page number. The `tocloft` package provides for general leaders for all entries. The ‘dot’ in a leader is given by the value of `\cftdot`. Its default definition is `\newcommand{\cftdot}{.}` which gives the default dotted leader. By changing `\cftdot` you can use symbols other than a period in the leader. For example

```
\renewcommand{\cftdot}{\ensuremath{\ast}}
```

will result in a dotted leader using asterisks as the symbol.

`\cftdotsep` Each kind of entry can control the separation between the dots in its leader (see below). For consistency though, all dotted leaders should use the same spacing. The macro `\cftdotsep` specifies the default spacing. Its value is a number. However, if the separation is too large then no dots will be actually typeset. The macro `\cftnodots` is a separation value that is ‘too large’.

`\cftsetpnumwidth` The page numbers are typeset in a fixed width box. The command `\cftsetpnumwidth{<length>}` can be used to change the width of the box (L^AT_EX’s internal `\@pnumwidth`). The title texts will end before reaching the righthand margin. `\cftsetrmarg{<length>}` can be used to set this distance (L^AT_EX’s internal `\@tocrmarg`). Note that the length used in `\cftsetrmarg` should be greater than the length set in `\cftsetpnumwidth`. These values should remain constant in any given document.

`\cftpnumalign` The page numbers are typeset in a box as described above. By default they are right-aligned which is suitable when the page numbers are aligned vertically on the page so their digits line up. For a design with fixed width between a ToC entry and its page number, say, a left alignment may be more suitable. This can be controlled by setting the `\cftpnumalign` macro to `l`, `c`, or `r` (just like `\makebox`):

```
\renewcommand{\cftpnumalign}{l}
```

`\cftparskip` Normally the `\parskip` in the ToC, etc., is zero. This may be changed by changing the `\cftparskip` length. Note that the current value of `\cftparskip` is used for the ToC, LoF and LoT, but you can change the value before calling `\tableofcontents` or `\listoffigures` or `\listoftables` if one or other of these should have different values (which is not a good idea).

In the following I will use **X** to stand for the following:

- `part` for `\part` titles
- `chap` for `\chapter` titles
- `sec` for `\section` titles
- `subsec` for `\subsection` titles
- `subsubsec` for `\subsubsection` titles
- `para` for `\paragraph` titles
- `subpara` for `\subparagraph` titles
- `fig` for figure `\caption` titles
- `subfig` for subfigure `\caption` titles
- `tab` for table `\caption` titles
- `subtab` for subtable `\caption` titles

`\cftbeforeXskip` This controls the vertical space before an entry. It can be changed by using `\setlength`.

`\cftXindent` This controls the indentation of an entry from the left margin (*indent* in Figure 1). It can be changed using `\setlength`.

`\cftXnumwidth` This controls the space allowed for typesetting title numbers (*numwidth* in Figure 1).

It can be changed using `\setlength`. Second and subsequent lines of a multiline title will be indented by this amount.

The remaining commands are related to the specifics of typesetting an entry. This is a simplified pseudo-code version for the typesetting of numbered and unnumbered entries.

```
{\cftXfont {\cftXpresnum SNUM\cftXaftersnum\hfil} \cftXaftersnumb TITLE}%
  {\cftXleader}{\cftXpagefont PAGE}\cftXafterpnum\cftXpnumfinal\par
```

```
{\cftXfont TITLE}{\cftXleader}{\cftXpagefont PAGE}\cftXafterpnum\cftXpnumfinal\par
```

where SNUM is the section number, TITLE is the title text and PAGE is the page number.

In the numbered entry the pseudo-code

```
{\cftXpresnum SNUM\cftXaftersnum\hfil}
```

is typeset within a box of width `\cftXnumwidth`.

`\cftXfont` This controls the appearance of the title (and its preceding number, if any). It may be changed using `\renewcommand`.

`\cftXpresnum` Normally the section number is typeset within a box of width `\cftXnumwidth`.

`\cftXaftersnum` Within the box the macro `\cftXpresnum` is first called, then the number is typeset, and

`\cftXaftersnumb` next the `\cftXaftersnum` macro is called after the number is typeset. The last command

`\cftpartinitnumwidth` within the box is `\hfil` to make the box contents flushleft. After the box is typeset the `\cftXaftersnumb` macro is called before typesetting the title text. All three of these can be changed by `\renewcommand`. By default they are defined to do nothing.

In the standard classes the ToC entry for a `\part` is just typeset as the number and title, followed by the page number, with the `\cftpartpresnum` macro being called before typesetting the number and title. Due to L^AT_EX idiosyncracies, `\cftpartpresnum` may become doubled in the output if a third-party package behaves differently to that of the default internal L^AT_EX commands. The `tocloft` package contains specific code to prevent this in the case of the KOMA Script classes and for the `titlesec` package; please contact the maintainer to add further corrections if you discover other packages which also exhibit this mis-behaviour.

When a standard class is used the entry for `\part` doesn't use `\numberline` and so `\cftpartaftersnum` and `\cftpartaftersnumb` macros have no effect (but they may do something if a non-standard class is used). This changes if `\DocumentMetadata` is used: then also `\part` uses `\numberline`. To allow to adapt the formatting to the one used in the standard classes, the command uses `\cftpartinitnumwidth` to reset the width of the box with the number to the natural width. Its default definition is

```
\newcommand{\cftpartinitnumwidth}[1]{\settowidth\@tempdima{#1}}
```

`\cftXleader` `\cftXleader` defines the leader between the title and the page number; it can be changed by `\renewcommand`. The spacing between any dots in the leader is controlled by `\cftXdotsep` (`\@dotsep` in Figure 1). It can be changed by `\renewcommand` and its value must be either a number (e.g., 6.6 or `\cftdotsep`) or `\cftnodots` (to disable the dots). The spacing is in terms of *math units* where there are 18mu to 1em.

`\cftXfillnum` This commands typesets the leader. It contains the sockets for tagging support and should normally not be redefined by a user.

`\cftXpagefont` This defines the font to be used for typesetting the page number. It can be changed by `\renewcommand`.

`\cftXafterpnum` This macro is called after the page number has been typeset. Its default is to do nothing. It can be changed by `\renewcommand`.

`\cftXpnumfinal` This macro is called after `\cftXafterpnum`. By default it contains a pair of kerns

(stored in the command `\cftpnumfinaldefault`) that with `microtype` suppresses protrusion in this place. Such kerns are also used in the L^AT_EX kernel and normally there should be no need to remove or change them, but with this command it is possible. It can be changed by `\renewcommand`.

`\cftpnumfinaldefault` The default kerns used in `\cftXpnumfinal`.

`\cftsetindents` The command `\cftsetindents{⟨entry⟩}{⟨indent⟩}{⟨numwidth⟩}` sets the `⟨entry⟩`'s *indent* to the length `⟨indent⟩` and its *numwidth* to the length `⟨numwidth⟩`. The `⟨entry⟩` argument is the name of one of the standard entries (e.g., `subsection`) or the name of entry that has been defined with the `tocloft` package. For example `\cftsetindents{figure}{0em}{1.5em}` will make figure entries left justified.

Various effects can be achieved by changing the definitions of `\cftXfont`, `\cftXaftersnum`, `\cftXaftersnumb`, `\cftXleader` and `\cftXafterpnum`, either singly or in combination. For the sake of some examples, assume that we have the following initial definitions

```
\newcommand{\cftXfont}{}
\newcommand{\cftXaftersnum}{}
\newcommand{\cftXaftersnumb}{}
\newcommand{\cftXleader}{\cftdotfill{\cftXdotsep}}
\newcommand{\cftXdotsep}{\cftdotsep}
\newcommand{\cftXpagefont}{}
\newcommand{\cftXafterpnum}{}

```

(Note that the same font should be used for the title, leader and page number to provide a coherent appearance).

- To eliminate the dots in the leader:

```
\renewcommand{\cftXdotsep}{\cftnodots}

```

- To put something (e.g., a name) before the title (number):

```
\renewcommand{\cftXpresnum}{SOMETHING }

```

- To add a colon after the section number:

```
\renewcommand{\cftXaftersnum}{:}

```

- To put something before the title number, add a colon after the the title number, set everything in bold font, and start the title text on the following line:

```
\renewcommand{\cftXfont}{\bfseries}
\renewcommand{\cftXleader}{\bfseries\cftdotfill{\cftXdotsep}}
\renewcommand{\cftXpagefont}{\bfseries}
\renewcommand{\cftXpresnum}{SOMETHING }
\renewcommand{\cftXaftersnum}{:}
\renewcommand{\cftXaftersnumb}{\}

```

If you are adding text in the number box in addition to the number, then you will probably have to increase the width of the box so that multiline titles have a neat vertical alignment; changing box widths usually implies that the indents will require modification as well.⁴ One possible method of adjusting the box width for the above example is:

⁴Lyndon Dudding (lyndon.dudding@totalise.co.uk) discovered this.

```

\newlength{\mylen} % a "scratch" length
\settowidth{\mylen}{\bfseries\cftXpresnum\cftXaftersnum} % extra space
\addtolength{\cftXnumwidth}{\mylen} % add the extra space

```

- To set the section numbers flushright:⁵

```

\setlength{\mylen}{0.5em} % need some extra space at end of number
\renewcommand{\cftXpresnum}{\hfill} % note the double `l'
\renewcommand{\cftXaftersnum}{\hspace*{\mylen}}
\addtolength{\cftXnumwidth}{\mylen}

```

In the above, the added initial `\hfill` in the box overrides the final `\hfil` in the box, thus shifting everything to the right hand end of the box. The extra space is so that the number is not typeset immediately at the left of the title text.

- To set the entry ragged left (but this only looks good for single line titles):

```

\renewcommand{\cftXfont}{\hfill\bfseries}
\renewcommand{\cftXleader}{\hspace*{\mylen}}

```

- To set the page number immediately after the entry text instead of at the righthand margin:

```

\renewcommand{\cftXleader}{\hspace*{\mylen}}
\renewcommand{\cftXafterpnum}{\cftparskip}
\renewcommand{\cftpnumalign}{l}

```

`\cftparskip`

By default the `\parskip` value is locally set to fill up the last line of a paragraph. Just changing `\cftXleader` puts horrible interword spaces into the last line of the title. The `\cftparskip` command is part of the `tocloft` package and is provided just so that the above effect can be achieved. In addition, this is a good example of when it would be suitable to change the alignment of the page number box.

- To remove the space inserted between table and figure caption entries between chapters:

```

\begingroup
\renewcommand*{\addvspace}[1]{\hspace*{\mylen}}
\listoftables
\listoffigures
\endgroup

```

`\cftpagenumbersoff`

The command `\cftpagenumbersoff{<entry>}` will eliminate the page numbers for `<entry>` in the listing, where `<entry>` is the name of one of the standard kinds of entries (e.g., `subsection`, or `figure` — including `subfigure` if the `subfigure` package is used — etc.), or the name of a new entry defined with the `tocloft` package.

`\cftpagenumberson`

The command `\cftpagenumberson{<entry>}` reverses the effect of a corresponding `\cftpagenumbersoff`.

One question that appeared on the `comp.text.tex` newsgroup asked how to get the titles of Appendices list in the ToC *without* page numbers. Here is a simple way of doing it, assuming the document has chapters

⁵With thanks to David Holz (1bda@earthlink.net) for requesting this.

```
...
\appendix
\addtocontents{toc}{\cftpagenumbersoff{chapter}}
\chapter{First appendix}
```

If there are other chaptered headings to go into the ToC after the appendices, then it will be necessary to do a similar

```
\addtocontents{toc}{\cftpagenumberon{chapter}}
```

to restore the page numbering in the ToC.

Similarly, if you are using the `subfigure` package you may want to eliminate the page numbers for the subfigure captions. This can be accomplished by:

```
\cftpagenumbersoff{subfigure}
```

At this point, I leave it up to your ingenuity as to other effects that you can achieve. However, if you come up with further examples, let me know for possible inclusion in a later version of this document.

2.4 New list of...

`\newlistof` The command `\newlistof[⟨within⟩]{⟨entry⟩}{⟨ext⟩}{⟨listofname⟩}` creates a new List of ..., and assorted commands to go along with it.

The first required argument, `⟨entry⟩` is used to define a new counter called `entry`. The optional `⟨within⟩` argument can be used so that `entry` gets reset to one every time the counter called `within` is changed. That is, the first two arguments are equivalent to calling `\newcounter{⟨entry⟩}[⟨within⟩]`.

The next argument, `⟨ext⟩`, is the file extension for the new List of. The last argument, `⟨listofname⟩`, is the text for the heading of the new List of. As an example:

```
\newcommand{\listanswername}{List of Answers}
\newlistof[chapter]{answer}{ans}{\listanswername}
```

will create a new `answer` counter that will be reset at the start of each `\chapter{...}`. Any answer titles will be written to the file `jobname.ans` and `\listanswername` will be used as the list heading. A command `\listofanswer` is created which can be used just like the `\listoftables` or `tableofcontents` commands to generate a listing. It is up to you to specify how the entries are put into the new List of Answers. Here is a very simple example, remembering that an `answer` counter has been created.

```
\newcommand{\answer}[1]{%
  \refstepcounter{answer}
  \par\noindent\textbf{Answer \theanswer. #1}
  \addcontentsline{ans}{answer}{\protect\numberline{\theanswer}#1}\par}
```

which, when used like:

```
\answer{Hard} The \ldots will print as:
```

Answer 1. Hard

The ...

As mentioned above, the `\newlistof` command creates several new commands, most of which you should now be familiar with. For convenience, assume that `\newlistof{X}{ZZ}{...}` has been issued; so `X` is the name of the new counter and corresponds to the `X` in section 2.3, and `ZZ` is the new file extension and corresponds to the `Z` in section 2.2. Then, among others, the following new commands will be made available.

The five commands, `\cftmarkZZ`, `\cftbeforeZZtitleskip`, `\cftafterZZtitleskip`, `\cftZZtitlefont`, and `\cftafterZZtitle`, are analogous to the commands of the same names described in section 2.2.

- `\listofX` The command `\listofX` is similar to `\listoftables`, etc., in that it typesets the new listing at the point where it is called.
- `\ZZdepth` The command `\ZZdepth{<number>}` is analogous to the standard `\tocdepth{<number>}` command, in that it specifies that entries in the new listing should not be typeset if their numbering level is greater than `<number>`. The default definition is `\setcounter{ZZdepth}{1}`.
- `\newlistentry` The command `\newlistentry[<within>]{<entry>}{<ext>}{<level-1>}` creates new commands for typesetting a new kind of entry in a listing. It is used internally by the `\newlistof` command but may be used independently.
 The first required argument, `<entry>` is used to define a new counter called `entry`. The optional `<within>` argument can be used so that `entry` gets reset to one every time the counter called `within` is changed. That is, the first two arguments are equivalent to calling `\newcounter{<entry>}[<within>]`. The second required argument, `<ext>`, is the file extension for the entry listing. The last argument, `<level-1>`, is a number specifying the numbering level minus one, of the entry in a listing. For example, the command `\newlistof[chapter]{answer}{ans}{\listanswername}` will call the command:
`\newlistentry[chapter]{answer}{ans}{0}`

Calling `\newlistentry` creates several new commands. Assuming that it is called as `\newlistentry[within]{X}{ZZ}{N}`, where `X` and `ZZ` are similar to the previous uses of them, and `N` is an integer number, then the following commands are made available.

The set of commands `\cftbeforeXskip`, `\cftXfont`, `\cftXpresnum`, `\cftXaftersnum`, `\cftXaftersnumb`, `\cftXleader`, `\cftXdotsep`, `\cftXpagefont`, `\cftXafterpnum`, and `\cftXpnumfinal` are analogous to the commands of the same names described in section 2.3. Their default values are also as described earlier.

The default values of `\cftXindent` and `\cftXnumwidth` are set according to the value of the `<level-1>` argument (i.e., `N` in this example). For `N=0` the settings correspond to those for sections in non-chaptered documents, as listed in Table 1. For `N=4` the settings correspond to subparagraphs in non-chaptered documents, and for intermediate values correspond to the matching sectional division in chaptered documents. For values of `N` less than zero or greater than four, or for non-default values, use the `\cftsetindents` command to set the values.

- `\l@X` `\l@X` is an internal command that typesets an entry in the list, and is defined in terms of the above `\cft*X*` commands. It will not typeset an entry if `\ZZdepth` is `N` or less, where `ZZ` is the listing's file extension.
- `\theX` The command `\theX` prints the value of the `X` counter. It is initially defined so that it prints arabic numerals. If the optional `<within>` argument is used, `\theX` is defined as `\renewcommand{\theX}{\thewithin.\arabic{X}}` otherwise as `\renewcommand{\theX}{\arabic{X}}`.

As an example of the independent use of `\newlistentry`, the following will set up for sub-answers.

```
\newlistentry[answer]{subanswer}{1}
\cftsetindents{subanswer}{1.5em}{3.0em}
\renewcommand{\thesubanswer}{\theanswer.\alph{subanswer}}
\newcommand{\subanswer}[1]{%
```

```

\refstepcounter{subanswer}
\par\textbf{\thesubanswer} #1}
\addcontentsline{ans}{subanswer}{\protect\numberline{\thesubanswer}#1}}
\setcounter{ansdepth}{2}

```

And then:

```

\answer{Harder} The \ldots
\subanswer{Reformulate the problem} It assists \ldots

```

will be typeset as:

Answer 2. Harder

The ...

2.a) Reformulate the problem It assists ...

By default the answer entries will appear in the List of Answers listing (typeset by the `\listofanswer` command). In order to get the subanswers to appear, the `\setcounter{ansdepth}{2}` command was used above.

To turn off page numbering for the subanswers, do

```
\cftpagenumbersoff{subanswer}
```

As another example of `\newlistentry`, suppose that an extra sectioning division below `subparagraph` is required, called `subsubpara`. The `\subsubpara` command itself can be defined via the LaTeX kernel `\@startsection` command. Also it is necessary to define a `\subsubparamark` macro, a new `subsubpara` counter, a `\thesubsubpara` macro and a `\l@subsubpara` macro. Using the `tocloft` package's `\newlistentry` takes care of most of these as shown below (remember the caveats about commands with `@` signs in them).

```

\newcommand{\subsubpara}{\@startsection{subpara}%
{6}%
level
{\parindent}%
indent from left margin
{3.25ex \@plus1ex \@minus .2ex}%
skip above heading
{-1em}%
runin heading with 1em between title & text
{\normalfont\normalsize\itshape}% italic number and title
}
\newlistentry[subparagraph]{subsubpara}{toc}{5}
\cftsetindents{subsubpara}{14.0em}{7.0em}
\newcommand*{\subsubparamark}[1]{ % gobble heading mark

```

Each List of...uses a file to store the list entries, and these files must remain open for writing throughout the document processing. TeX has only a limited number of files that it can keep open, and this puts a limit on the number of listings that can be used. For a document that includes a ToC but no other extra ancillary files (e.g., no index or bibliography output files) the maximum number of LoX's, including a LoF and LoT, is no more than about eleven. If you try and create too many new listings LaTeX will respond with the error message:

No room for a new write

If you get such a message the only recourse is to redesign your document.

The `tocloft` package does not provide a simple means of specifying new Lists of Floats or float environments. For those, I recommend the `ccaption` package [Wil01].

2.5 Experimental utilities

The macros described in this section are even more experimental than those described previously.

`\cftchapterprecis` Some old style novels, and even some modern text books,⁶ include a short synopsis of the contents of the chapter either immediately after the chapter heading or in the Toc, or in both places.

The command `\cftchapterprecis{<text>}` prints its argument both at the point in the document where it is called, and also adds it to the `.toc` file. For example:

```
...
\chapter{} % first chapter
\cftchapterprecis{Our hero is introduced; family tree; early days.}
...
```

If tagging is enabled, the toc entry is currently tagged as an individual toc entry.

`\cftchapterprecishere` The `\cftchapterprecis` command calls these two commands to print the text in the document (the `\...here{<text>}` command) and to put it into the ToC (the `\...toc{<text>}` command). These can be used individually if required.

Sometimes it may be desirable to make a change to the global parameters for an individual entry. For example, a figure might be placed on the end paper of a book (the inside of the front or back cover), and this needs to be placed in a LoF with the page number set as, say ‘inside front cover’. If ‘inside front cover’ is typeset as an ordinary page number it will stick out into the margin. Therefore, the parameters for this particular entry need to be changed.

`\cftlocalchange` The command `\cftlocalchange{<file>}{<pnumwidth>}{<tocrmarg>}` will write an entry into `<file>` to reset the global parameters. The command should be called again after any special entry to reset the parameters back to their usual values. Any fragile commands used in the arguments must be protected.

`\cftaddtitleline` The command `\cftaddtitleline{<file>}{<kind>}{<title>}{<page>}` will write a `\contentsline` entry into `<file>` for a `<kind>` entry with title `<title>` and page number `<page>`. That is, an entry is made of the form:

```
\contentsline{kind}{title}{page}{target}
```

Any fragile commands used in the arguments must be protected.

`\cftaddnumtitleline` The command `\cftaddnumtitleline{<file>}{<kind>}{<num>}{<title>}{<page>}` is similar except that it also includes `<num>` as the argument to the `\numberline`. That is, an entry is made of the form:

```
\contentsline{kind}{\numberline{num} title}{page}{target}
```

Any fragile commands used in the arguments must be protected.

As an example of the use of these commands, noting that the default $\text{\LaTeX}$ values for `\@pnumwidth` and `\@tocrmarg` are 1.55em and 2.55em respectively, one might do the following for a figure on the frontispiece page.

```
...
% this is the frontispiece page with no number
% draw or import the picture (with no \caption)
\cftlocalchange{lof}{4em}{5em} % make pnumwidth big enough for
                                % frontispiece and change margin to suit
\cftaddtitleline{lof}{figure}{The title}{frontispiece}
```

⁶For example, Robert Sedgewick, *Algorithms*, Addison-Wesley, 1983.

```
\cftlocalchange{lof}{1.55em}{2.55em} % return to normal settings
...
```

Recall that a `\caption` command will put an entry in the `.lof` file, which is not wanted here. If a caption is required, then you can either craft one yourself or, assuming that your general captions are not too exotic, use the `\legend` command from the `ccaption` package. If the illustration is numbered, use the `\cftaddnumtitleline` command instead of `\cftaddtitleline`.

`\cftZprehook` It's surprisingly difficult to achieve multicolumn ToCs; can you guess what the prob-
`\cftZposthook` lem is to write the following?

```
\begin{multicols}{2}
\tableofcontents
\end{multicols}
```

Probably the easiest way to do it in regular L^AT_EX is something like

```
\RequirePackage{multicol}
\AtBeginDocument{\addtocontents{toc}{\protect\begin{multicols}{2}}}
\AtEndDocument {\addtocontents{toc}{\protect\end {multicols}}}
```

This method of writing to the `.toc` file is most flexible for trying to control the typesetting output within the table of contents.

To make this *slightly* easier with `tocloft`, the following macros are available: `\cftZprehook` and `\cftZposthook`, where Z is `toc`, `lof`, `lot`, etc. If these are defined, they insert material just before the actual typesetting of the entries of the table of contents and so on. A multicolumn ToC can therefore be achieved with this:

```
\RequirePackage{multicol}
\renewcommand\cfttocprehook{\begin{multicols}{2}}
\renewcommand\cfttocposthook{\end{multicols}}
```

2.6 Usage with other packages

The `tocloft` and `tocbibind` packages can be used together in the same document. The `tocbibind` package provides easy means of adding document elements like the bibliography or the index to the Table of Contents. However there is one known potential problem:

- If the argument to the `\tocotherhead` command is other than one of the normal sectioning divisions (i.e., part through to sub-paragraph) such as `\tocotherhead{clause}`, then this will almost certainly cause a problem (as the `tocloft` package will not know how to define the corresponding `\l@clause` command). In such a case you will have to supply the appropriate macros yourself.

`\@cftbsnum` Some packages, like the `float` package by Anselm Lingnau, enable the creation of
`\@cftasnum` other kinds of *List of ...*. The `tocloft` package is only minimally able to change the for-
`\@cftasnumb` matting of these, principally because the packages are independent of each other and, in
the case of the `float` package, new kinds of float environments and their associated lists
can be created on the fly at any point in a document. Some aspects of the typesetting are
controlled by `\@cftbsnum`, `\@cftasnum` and `\@cftasnumb` commands. These are equiva-
lent to the `\cftXpresnum`, `\cftXaftersnum` and `\cftXaftersnumb` commands described
earlier. By default they are defined to do nothing, but may be renewed to do something.

The `tocloft` and `minitoc` packages have an unfortunate interaction,⁷ which fortunately can be fixed. In the normal course of events, when `minitoc` is used in a chaptered document it will typeset section entries in the minitocs in bold font. If `tocloft` is used in conjunction with `minitoc`, then the minitoc section entries are typeset in the normal font, except for the page numbers which are in bold font, while the ToC section entries are all in normal font.

One cure, if you want the minitoc section entries to be all in normal font is to put:

```
\renewcommand{\mtcSfont}{\small\normalfont}
```

in the preamble.

Otherwise, the cure is the following incantation:

```
\renewcommand{\cftsecfont}{\bfseries}
\renewcommand{\cftsecleader}{\bfseries\cftdotfill{\cftdotsep}}
\renewcommand{\cftsecpagefont}{\bfseries}
```

To have the section entries in both the ToC and the minitocs in bold then put the incantation in the preamble. To have only the minitoc section entries in bold while the ToC entries are in the normal font, put the incantation between the `\tableofcontents` command and the first `\chapter` command.

In general, use with other packages that redefine any of the macros that `tocloft` also modifies is likely to be problematic.

3 The package code

```
1 \usc
2 \RequirePackage{etoolbox}
```

In order to try and avoid name clashes with other packages, each internal name will include the character string `@cft`.

```
\@cftifundefined Due to a conflict with how this package and fancyhdr checked for undefinedness.
3 \newcommand\@cftifundefined[1]{%
4 \begingroup\expandafter\expandafter\expandafter\endgroup
5 \expandafter\ifx\csname #1\endcsname\relax
6 \expandafter\@firstoftwo
7 \else
8 \expandafter\@secondoftwo
9 \fi}
```

(End of definition for \@cftifundefined.)

```
\@cftquit We will be using either chapter or section type headings for the ToC, etc., so we need to
\ifcfthaschapter know which of these the document class supports.
```

```
10 \newcommand{\@cftquit}{}
11 \newif\ifcfthaschapter
```

(End of definition for \@cftquit and \ifcfthaschapter.)

⁷Discovered by Lyndon Dudding (lyndon.dudding@totalise.co.uk).

`\if@cftkoma` The koma classes have different defaults than the standard classes, so we need to know if a koma class has been loaded.

```

12 \newif\if@cftkoma
13 \cftkomafalse
14 \@ifclassloaded{scrartcl}{\@cftkomatrue}{}
15 \@ifclassloaded{scrreprt}{\@cftkomatrue}{}
16 \@ifclassloaded{scrbook}{\@cftkomatrue}{}

```

(End of definition for \if@cftkoma.)

`\if@cfttitlesec`

```

17 \newif\if@cfttitlesec
18 \AtBeginDocument{\@ifpackageloaded{titlesec}{\@cfttitlesectrue}{} }

```

(End of definition for \if@cfttitlesec.)

Issue a warning if there are no recognised sectional divisions and then skip the rest of the package code.

```

19 \@cftifundefined{chapter}{%
20   \cftaschapterfalse
21   \@cftifundefined{section}{%
22     \PackageWarning{tocloft}{%
23       {I don't recognize any sectional divisions so I'll do nothing}
24     \renewcommand{\@cftquit}{\endinput}
25     }\PackageInfo{tocloft}{The document has section divisions}}
26   }\cftaschaptertrue
27   \PackageInfo{tocloft}{The document has chapter divisions}}

```

Perhaps quit now.

```

28 \cftquit

```

Use chapter style if `\if@cftaschapter` is TRUE, otherwise section style.

`\if@cfttocbibind` A flag that is set TRUE iff the tocbibind package has been loaded. The 1998/11/15 version of tocbibind does not necessarily work well with tocloft.

```

29 \newif\if@cfttocbibind
30 \AtBeginDocument{%
31   \@ifpackageloaded{tocbibind}{\@cfttocbibindtrue}{\@cfttocbibindfalse}
32   \if@cfttocbibind
33     \@ifpackagelater{tocbibind}{1998/11/16}{\PackageWarning{tocloft}{%
34       You are using a version of the tocbibind package\MessageBreak
35       that is not compatible with tocloft.\MessageBreak
36       The results may be surprising.\MessageBreak
37       Consider installing the current version of tocbibind.}}
38     \fi
39   }
40 }

```

(End of definition for \if@cfttocbibind.)

`\if@cftnctoc` A boolean used to implement the titles option. It is TRUE if the ToC, LoT, LoF titles should use the default styles.

```

41 \newif\if@cftnctoc\cftnctocfalse
42 \DeclareOption{titles}{\@cftnctoctrue}
43 %% \ProcessOptions\relax

```

(End of definition for `\if@cftnctoc`.)

`\if@cftsubfigopt` A boolean used to implement the subfigure option.

```
44 \newif\if@cftsubfigopt\@cftsubfigoptfalse
45 \DeclareOption{subfigure}{\@cftsubfigopttrue}
```

(End of definition for `\if@cftsubfigopt`.)

Process the options.

```
46 \ProcessOptions\relax
47
```

`\tocloftpagestyle` A user-level macro to set the pagestyle for the first page of the ToC, etc. The default is the plain pagestyle.

```
48 \newcommand{\tocloftpagestyle}[1]{%
49   \def\@cftpagestyle{\thispagestyle{#1}}
50   \tocloftpagestyle{plain}
51
```

(End of definition for `\tocloftpagestyle` and `\@cftpagestyle`. These functions are documented on page 7.)

`\cftmarktoc` These three macros set the style for running heads. They are initialised to give the default appearance.

```
\cftmarklof
\cftmarklot
52 \newcommand{\cftmarktoc}{%
53   \@mkboth{\MakeUppercase\cfttoctitlename}{\MakeUppercase\cfttoctitlename}}
54 \newcommand{\cftmarklof}{%
55   \@mkboth{\MakeUppercase\cftloftitlename}{\MakeUppercase\cftloftitlename}}
56 \newcommand{\cftmarklot}{%
57   \@mkboth{\MakeUppercase\cftlottitlename}{\MakeUppercase\cftlottitlename}}
58 \if@cftkoma
59   \renewcommand{\cftmarktoc}{%
60     \@mkboth{\cfttoctitlename}{\cfttoctitlename}}
61   \renewcommand{\cftmarklof}{%
62     \@mkboth{\cftloftitlename}{\cftloftitlename}}
63   \renewcommand{\cftmarklot}{%
64     \@mkboth{\cftlottitlename}{\cftlottitlename}}
65 \fi
```

(End of definition for `\cftmarktoc`, `\cftmarklof`, and `\cftmarklot`. These functions are documented on page 7.)

`\@cfttocstart` Two macros to perform the actions at the beginning and end of the `\tableofcontents` command (and friends). `\@cfttocstart` deals with chaptered documents, ensuring that the ToC is typeset in a single column (see `classes.dtx` for the original code). These macros are also provided by the `ccaption` package.

```
\@cfttocfinish
66 \providecommand{\@cfttocstart}{%
67   \if@cftthaschapter
68     \if@twocolumn
69       \@restonecoltrue\onecolumn
70     \else
71       \@restonecolfalse
72     \fi
73   \fi}
```

`\cfttocfinish` resets, if required, twocolumn typesetting.

```

74 \providecommand{\cfttocfinish}{%
75   \if@cfthaschapter
76     \if@restonecol\twocolumn\fi
77   \fi}

```

(End of definition for `\cfttocstart` and `\cfttocfinish`.)

`\phantomsection` This is provided because the `hyperref` package screws with `\addcontentsline`.

```

78 \providecommand{\phantomsection}{}
79

```

(End of definition for `\phantomsection`.)

`\@cftdobibtoc` If the `tocbibind` package has been used and it has redefined `\tableofcontents` we need to cater for that. The contents of the definition are defined in `tocbibind`.

```

80 \newcommand{\@cftdobibtoc}{%
81   \if@dotoc
82     \if@bibchapter
83       \phantomsection
84       \addcontentsline{toc}{chapter}{\cfttoctitlename}
85     \else
86       \phantomsection
87       \addcontentsline{toc}{\@tocextra}{\cfttoctitlename}
88     \fi
89   \fi}
90

```

(End of definition for `\@cftdobibtoc`.)

`\cftparskip` The `\parskip` local to the ToC, etc., is set to the length `\cftparskip`.

```

91 \newlength{\cftparskip}
92 \setlength{\cftparskip}{0pt}
93

```

(End of definition for `\cftparskip`. This function is documented on page 9.)

`\cftpnumfinaldefault` A kern pair used at the end of entries to avoid problems with protrusion.

```

94 \newcommand\cftpnumfinaldefault{\kern-\p@\kern\p@}

```

(End of definition for `\cftpnumfinaldefault`. This function is documented on page 11.)

`\cftZtitlename`

```

95 \newcommand\cfttoctitlename{\contentsname}
96 \newcommand\cftloftitlename{\listfigurename}
97 \newcommand\cftlottitlename{\listtablename}

```

(End of definition for `\cftZtitlename`. This function is documented on page 8.)

`\cfttitlecommand` This command is used to typeset the title if `\DocumentMetadata` is used and the `titles` option is not used.

```

98 \if@cfthaschapter
99   \newcommand\cfttitlecommand[1]{\cftchapter{#1}}
100 \else
101   \newcommand\cfttitlecommand[1]{\cftsection{#1}}
102 \fi

```

(End of definition for \cfttitlecommand. This function is documented on page 8.)

\cftchapter
\cftsection

```

103 \IfDocumentMetadataTF
104 {
105   \DeclareDocumentCommand \cftchapter {0}{m}
106     {%
107       \chapter
108       [
109         toc={},
110         placement=normal,
111         unnumbered,
112         after-penalty-vspace=\dimeval{\UseName{cftbefore#2titleskip}}-\topskip},
113         after-vspace=\UseName{cftafter#2titleskip},
114         final-code=\@cftpagestyle\@afterheading,
115         title-decls=\UseName{cft#2titlefont},
116         #1
117       ] {\UseName{cft#2titlename}}}%
118   }
119   \DeclareDocumentCommand \cftsection {0}{m}
120     {%
121       \section[
122         toc={},
123         placement=normal,
124         unnumbered,
125         after-penalty-vspace=\UseName{cftbefore#2titleskip},
126         after-vspace=\UseName{cftafter#2titleskip},
127         final-code=\@cftpagestyle\@afterheading,
128         title-decls=\UseName{cft#2titlefont},
129         #1
130       ] {\UseName{cft#2titlename}}}%
131   }
132 }
133 {
134   \DeclareDocumentCommand \cftchapter {0}{m}
135     {%
136       \addpenalty\@secpenalty
137       \vspace*{\UseName{cftbefore#2titleskip}}}%
138       \@cftpagestyle
139       {\interlinepenalty\@M
140       {\UseName{cft#2titlefont}}\UseName{cft#2titlename}}}%
141       {\UseName{cftafter#2title}}}%
142       \UseName{cftmark#2}}%
143       \par\nobreak
144       \vskip \UseName{cftafter#2titleskip}}%
145       \@afterheading}%
146   }
147   \DeclareDocumentCommand \cftsection {0}{m}
148     {%
149       \addpenalty\@secpenalty
150       \vspace{\UseName{cftbefore#2titleskip}}}%
151       \@cftpagestyle
152       {\interlinepenalty\@M
153       {\UseName{cft#2titlefont}}\UseName{cft#2titlename}}}%

```

```

154     {\UseName{cftafter#2title}}}%
155     \UseName{cftmark#2}}%
156     \par\nobreak
157     \vskip \UseName{cftafter#2titleskip}%
158     \@afterheading}%
159   }
160 }

```

(End of definition for `\cftchapter` and `\cftsection`. These functions are documented on page 8.)

`\tableofcontents` This is a parameterised version of the default `\tableofcontents` command. Each class has its own definition, but we have to cater for all classes in one definition, hence some of the checks. The definition is modified after all packages have been loaded. If the titles option has been used, then the command is not modified.

```

161 \AtBeginDocument{%
162   \if@cftnctoc
163     % ensure \cftparskip is still set properly
164     \let\OLD@starttoc\@starttoc
165     \renewcommand{\@starttoc}[1]{%
166       \begingroup
167         \parskip=\cftparskip
168         \OLD@starttoc{#1}%
169       \endgroup
170     }
171   \else
172     \renewcommand{\tableofcontents}{%
173       \@cfttocstart

```

Ensure that any previous paragraph has been finished. Within a group set the local paragraphing style and typeset the title.

```

174     \par
175     \begingroup
176     \parindent\z@ \parskip\cftparskip
177     \@cftmaketoctitle

```

If `tocbibind` has been used, then add the ToC name to the ToC.

```

178     \if@cfttocbibind
179       \@cftdobibtoc
180     \fi

```

Finally, read the `.toc` file and finish up.

```

181     \@starttoc{toc}%
182   \endgroup
183   \@cfttocfinish}
184 \fi
185 }

```

(End of definition for `\tableofcontents`. This function is documented on page 7.)

`\@cftmaketoctitle` This command typesets the title for the ToC.

```

186 \newcommand{\@cftmaketoctitle}{\cfttitlecommand{toc}}%

```

(End of definition for `\@cftmaketoctitle`.)

`\cftbeforetoctitleskip` These two lengths control the vertical spacing before and after the ToC title.

`\cftaftertoctitleskip`

```

187 \newlength{\cftbeforetoctitleskip}
188 \newlength{\cftaftertoctitleskip}

```

Their values depend on whether the document has chapters or not. In chaptered documents the default ToC title is typeset as a `\chapter*`, otherwise as a `\section*`.

```

189 \if@cfthaschapter
190   \setlength{\cftbeforetoctitleskip}{50pt}
191   \setlength{\cftaftertoctitleskip}{40pt}
192 \else
193   \setlength{\cftbeforetoctitleskip}{3.5ex \@plus 1ex \@minus .2ex}
194   \setlength{\cftaftertoctitleskip}{2.3ex \@plus .2ex}
195 \fi

```

(End of definition for `\cftbeforetoctitleskip` and `\cftaftertoctitleskip`. These functions are documented on page 7.)

`\cfttoctitlefont` The ToC title is typeset in the style given by `\cfttoctitlefont`. The macro
`\cftaftertoctitle` `\cftaftertoctitle` is called after typesetting the title. This is initialised to do nothing. Both these macros can be redefined to do other things (e.g., adding an `\hfill` to `\cfttoctitlefont` will make the title flushright).

```

196 \if@cfthaschapter
197   \newcommand{\cfttoctitlefont}{\normalfont\Huge\bfseries}
198   \if@cfkoma\renewcommand{\cfttoctitlefont}{\size@chapter\sectfont}\fi
199 \else
200   \newcommand{\cfttoctitlefont}{\normalfont\Large\bfseries}
201   \if@cfkoma\renewcommand{\cfttoctitlefont}{\size@section\sectfont}\fi
202 \fi
203 \newcommand{\cftaftertoctitle}{}

```

(End of definition for `\cfttoctitlefont` and `\cftaftertoctitle`. These functions are documented on page 8.)

`\cftsetpnumwidth` Users commands for setting `\@pnumwidth` and `\@tocrmarg`.

`\cftsetrmarg`

```

204 \newcommand{\cftsetpnumwidth}[1]{\renewcommand{\@pnumwidth}{#1}}
205 \newcommand{\cftsetrmarg}[1]{\renewcommand{\@tocrmarg}{#1}}

```

(End of definition for `\cftsetpnumwidth` and `\cftsetrmarg`. These functions are documented on page 9.)

`\cftpnumalign` Alignment string (as input to `\makebox` for the page number box.

```

206 \newcommand{\cftpnumalign}{r}

```

(End of definition for `\cftpnumalign`. This function is documented on page 9.)

`\cftdot` In the default ToC, a dotted line can be used to provide a leader between a title and the
`\cftdotfill` page number. The definition of this leader is buried in the `\@dottedtocline` command. The `\cftdotfill{<sep>}` command provides a parameterised version of the leader code, where `<sep>` is the separation between the dots in mu units. The symbol used for the ‘dots’ in the leader is given by the value of `\cftdot`. These macros are also provided by the `caption` package.

```

207 \providecommand{\cftdot}{.}
208 \providecommand{\cftdotfill}[1]{%
209   \def\@tempa{#1}%
210   \def\@tempb{\cftnodots}%

```

```

211 \ifx\@tempa\@tempb
212   \hfill
213 \else
214   \leaders\hbox{$\m@th\mkern #1 mu\hbox{\cftdot}\mkern #1 mu$}\hfill
215 \fi
216 }

```

(End of definition for `\cftdot` and `\cftdotfill`. These functions are documented on page 8.)

`\cftdotsep` `\cftdotsep` holds the default dot separation, and is also provided by the `ccaption` package. If the kerns in `\cftdotfill` are large enough, then no dots will be printed. `\cftnodots` should be ‘large enough’. (Actually, `\cftnodots` is now used as a flag for a conditional branch, so its numerical value isn’t as important now.)

```

217 \providecommand{\cftdotsep}{4.5}
218 \newcommand{\cftnodots}{5000}

```

(End of definition for `\cftdotsep` and `\cftnodots`. These functions are documented on page 9.)

Now for the trickier bits regarding the typesetting of the ToC entries.

A `.toc` (also `.lof` and `.lot`) file consists of a list of

`\contentsline{<kind>}{<title>}{<page>}{<target>}` commands, where `<kind>` is the kind of heading (e.g., `part` or `section` or `figure`), `<title>` is the title text (including the number), and `<page>` is the page number. `<target>` is set by `hyperref` or by the tagging code for links and structure references. The entries are inserted into the file by calling the `\addcontentsline{<file>}{<kind>}{<title>}` command, where `<file>` is the file extension (e.g., `toc`, `lot`) and the other arguments are the same as for the `\contentsline` command. (Arbitrary stuff may also be put into the file via the `\addtocontents{<file>}{<text>}` command). The typesetting of the `\contentsline` entries is performed by commands of the form `\l@kind`. The sectioning and captioning commands call `\addcontentsline` to insert their titles into the `.toc` etc., files.

For the purposes at hand it is generally impossible to treat the typesetting of a title and its number separately, as both are bundled into the `<title>` argument within `\contentsline`. They could be handled separately if the `\contentsline` command was suitably modified. If this was done, then the `\addtocontentsline` command would also need to be changed which would then require the sectioning and captioning commands to be modified as well. This is certainly possible, but would cause problems if any other package also modified the sectioning or captioning commands, and there are several packages which do this.

Having said this, for all but Part entries, the sectional number is typeset via the `\numberline` command. We can take advantage of this fact.

I have taken the decision to not touch the `\contentsline` macro and instead to do what can be done with it as it exists. That is, I will modify the `\l@kind` commands. Essentially, my new definitions consist of inlined versions of the code for `\dottedtocline`.

`\cftparfillskip` The `\l@kind` commands modify (locally) the value of `\parfillskip`. `\cftparfillskip` is a copy of the default *TEXbook* `\parfillskip` definition.

```

219 \newcommand{\cftparfillskip}{\parfillskip=0pt plus1fil}

```

(End of definition for `\cftparfillskip`. This function is documented on page 12.)

`\numberline` The purpose of the `\numberline{<secnum>}` command is to typeset `<secnum>` left justified in a box of width `\@tempdima`. I redefine it to add three additional parameters, namely `\@cftbsnum`, `\@cftasnum` and `\@cftasnumb` (see `ltsect.dtx` for the original definition). `\@cftinitnumwidth` can be used to change `\@tempdima` based on the number e.g. in `\part`.

```

220 \newcommand\@cftinitnumwidth[1]{}
221 \renewcommand{\numberline}[1]{%
222   \@cftinitnumwidth{#1}%
223   \hb@xt@\@tempdima{%
224     \UseHookWithArguments{contentsline/number/before}{1}{#1}%
225     \@cftbsnum #1\@cftasnum\hfil
226     \UseHookWithArguments{contentsline/number/after}{1}{#1}%
227     \@cftasnumb
228   }

```

(End of definition for `\numberline` and `\@cftinitnumwidth`. These functions are documented on page 4.)

`\@cftbsnum` Originally these were not defined but were `\let` to appropriate commands in the `\l@...`
`\@cftasnum` commands, but they have to be defined in case something unexpected calls `\numberline`,
`\@cftasnumb` for example through use of the float package.⁸

```

229 \newcommand{\@cftbsnum}{}
230 \newcommand{\@cftasnum}{}
231 \newcommand{\@cftasnumb}{}

```

(End of definition for `\@cftbsnum`, `\@cftasnum`, and `\@cftasnumb`. These functions are documented on page 17.)

`\l@part` `\l@part{<title>}{<page>}` typesets the ToC entry for a `part` heading. It is a parameterised copy of the default `\l@part` (see `classes.dtx` for the original definition and the code below for `\l@section` for an explanation of most of this code). By default, Parts (and Chapters) do not have dotted leaders. This package provides for all entries to have dotted leaders.

`\ifcftdopart`

```

232 \newif\ifcftdopart
233 \newif\ifcfthaspart
234 \cftifundefined{part}{\cfthaspartfalse}{\cfthasparttrue}
235 \ifcfthaspart
236 \renewcommand*{\l@part}[2]{%
237   \@cftdopartfalse
238   \ifnum \c@tocdepth >-2\relax
239     \ifcfthaschapter
240       \cftdoparttrue
241     \fi
242     \ifnum \c@tocdepth >\m@ne
243       \ifcfthaschapter\else
244         \cftdoparttrue
245       \fi
246     \fi
247   \fi
248   \ifcftdopart
249     \ifcfthaschapter
250       \addpenalty{-\@highpenalty}%

```

⁸This bug was discovered by Andrew Thurber when using the `tocloft` and `algorithm` packages together.

```

251 \else
252 \addpenalty\@secpenalty
253 \fi
254 \addvspace{\cftbeforepartskip}%
255 \begingroup
256 {\leftskip \cftpartindent\relax
257 \rightskip \@tocrmarg
258 \parfillskip -\rightskip
259 \parindent \cftpartindent\relax\@afterindenttrue
260 \interlinepenalty\@M
261 \leavevmode
262 \@tempdima \cftpartnumwidth\relax
263 \let\@cftinitnumwidth\cftpartinitnumwidth
264 \let\@cftbsnum \cftpartpresnum
265 \let\@cftasnum \cftpartaftersnum
266 \let\@cftasnumb \cftpartaftersnumb
267 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip

```

In default L^AT_EX, the part ToC entry is written without `\numberline` and hence the ‘presnum’ needs to be inserted manually. In Koma-Script and titlesec (and probably others—let me know!), however, this is not the case.

```

268 \UseHookWithArguments{contentsline/text/before}{4}
269 {-1}{#1}{#2}{\@contentsline@destination}%
270 {\cftpartfont \if@cftkoma\else\if@cfttitlesec\else\cftpartpresnum\fi\fi #1}%
271 \UseHookWithArguments{contentsline/text/after}{4}
272 {-1}{#1}{#2}{\@contentsline@destination}%
273 \cftpartfillnum{#2}}
274 \nobreak
275 \if@cfthaschapter
276 \global\@nobreaktrue
277 \everypar{\global\@nobreakfalse\everypar{}}%
278 \else
279 \if@compatibility
280 \global\@nobreaktrue
281 \everypar{\global\@nobreakfalse\everypar{}}%
282 \fi
283 \fi
284 \endgroup
285 \fi}
286 \fi

```

(End of definition for `\l@part` and `\if@cftdopart`.)

`\cftbeforepartskip`
`\cftpartnumwidth`
`\cftpartfont`
`\cftpartpresnum`
`\cftpartaftersnum`
`\cftpartaftersnumb`
`\cftpartleader`
`\cftpartdotsep`
`\cftpartpagefont`
`\cftpartafterpnum`
`\cftpartpnumfinal`
`\cftpartindent`
`\cftpartfillnum`

These are the user commands to control the typesetting of Part entries. They are initialised to give the standard appearance.

```

287 \if@cfthaspart
288 \newlength{\cftbeforepartskip}
289 \setlength{\cftbeforepartskip}{2.25em \@plus\p@}
290 \newlength{\cftpartnumwidth}
291 \setlength{\cftpartnumwidth}{3em}
292 \newcommand{\cftpartfont}{\large\bfseries}
293 \newcommand{\cftpartinitnumwidth}[1]{\settowidth\@tempdima{#1}}
294 \newcommand{\cftpartpresnum}{}
295 \newcommand{\cftpartaftersnum}{}

```

```

296 \newcommand{\cftpartaftersnumb}{\hspace{1em}}
297 \newcommand{\cftpартleader}{\large\bfseries\cftdotfill{\cftpартdotsep}}
298 \newcommand{\cftpартdotsep}{\cftnodots}
299 \newcommand{\cftpартpagefont}{\large\bfseries}
300 \newcommand{\cftpартafterpnum}{\}
301 \newcommand{\cftpартpnumfinal}{\cftpnumfinaldefault}
302 \newlength{\cftpартindent}
303 \setlength{\cftpартindent}{0em}
304 \newcommand{\cftpартfillnum}[1]{%
305   {%
306     \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
307     \cftpартleader
308     \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
309   }%
310   {\UseHookWithArguments{contentsline/page/before}{4}
311     {-1}{\#1}{\@contentsline@destination}%
312     \makebox[\@pnumwidth][\cftpnumalign]{\cftpартpagefont \#1}%
313     \UseHookWithArguments{contentsline/page/after}{4}
314     {-1}{\#1}{\@contentsline@destination}%
315     \cftpartafterpnum\cftpартpnumfinal\par}%
316 }

```

koma classes use some different settings.

```

317 \if@cftkoma
318   \setlength{\cftpартnumwidth}{2em}
319   \renewcommand{\cftpартfont}{\sectfont\large}
320   \renewcommand{\cftpартpagefont}{\sectfont\large}
321 \fi
322 \fi

```

(End of definition for \cftbeforepartskip and others. These functions are documented on page 9.)

\l@chapter `\l@chapter{<title>}{<page>}` typesets the ToC entry for a chapter heading. It is a parameterised copy of the default `\l@chapter` (see `classes.dtx` for the original definition). This only applies to chaptered documents.

```

323 \if@cftchapter
324 \renewcommand*{\l@chapter}[2]{%
325   \ifnum \c@tocdepth > \m@ne
326     \addpenalty{-\@highpenalty}%
327     \vskip \cftbeforechapskip
328     {\leftskip \cftchapindent\relax
329       \rightskip \@tocrmarg
330       \parfillskip -\rightskip
331       \parindent \cftchapindent\relax\@afterindenttrue
332       \interlinepenalty\@M
333       \leavevmode
334       \@tempdima \cftchapnumwidth\relax
335       \let\@cftbsnum \cftchappresnum
336       \let\@cftasnum \cftchapaftersnum
337       \let\@cftasnumb \cftchapaftersnumb
338       \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
339       \UseHookWithArguments{contentsline/text/before}{4}
340       {0}{\#1}{\#2}{\@contentsline@destination}%
341       {\cftchapfont \#1}\nobreak

```

```

342     \UseHookWithArguments{contentsline/text/after}{4}
343     {0}{#1}{#2}{\@contentsline@destination}%
344     \cftchapfillnum{#2}}%
345     \fi}%
346 \fi

```

(End of definition for \l@chapter.)

`\cftbeforechapskip` `\cftchapindent` `\cftchapnumwidth` `\cftchapfont` `\cftchappresnum` `\cftchapaftersnum` `\cftchapaftersnumb` `\cftchapleader` `\cftchapdotsep` `\cftchappagefont` `\cftchapafterpnum` `\cftchappnumfinal` `\cftchapfillnum` These are the user commands to control the typesetting of Chapter entries. They are initialised to give the standard appearance.

```

347 \if@cfthaschapter
348   \newlength{\cftbeforechapskip}
349   \setlength{\cftbeforechapskip}{1.0em \@plus\p@}
350   \newlength{\cftchapindent}
351   \setlength{\cftchapindent}{0em}
352   \newlength{\cftchapnumwidth}
353   \setlength{\cftchapnumwidth}{1.5em}
354   \newcommand{\cftchapfont}{\bfseries}
355   \newcommand{\cftchappresnum}{}
356   \newcommand{\cftchapaftersnum}{}
357   \newcommand{\cftchapaftersnumb}{}
358   \newcommand{\cftchapleader}{\bfseries\cftdotfill{\cftchapdotsep}}
359   \newcommand{\cftchapdotsep}{\cftnodots}
360   \newcommand{\cftchappagefont}{\bfseries}
361   \newcommand{\cftchapafterpnum}{}
362   \newcommand{\cftchappnumfinal}{\cftpnumfinaldefault}
363   \newcommand{\cftchapfillnum}[1]{%
364     {\UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
365       \cftchapleader
366       \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
367     }\nobreak
368     \UseHookWithArguments{contentsline/page/before}{4}
369     {0}{#1}{\@contentsline@destination}%
370     \makebox[\cftchapnumwidth][\cftpnumalign]{\cftchappagefont #1}%
371     \UseHookWithArguments{contentsline/page/after}{4}
372     {0}{#1}{\@contentsline@destination}%
373     \cftchapafterpnum\cftchappnumfinal\par
374   }

```

koma classes have different chapter settings.

```

375 \if@cftkoma
376   \renewcommand{\cftchapfont}{\sectfont}
377   \fi
378 \fi
379

```

(End of definition for `\cftbeforechapskip` and others. These functions are documented on page 9.)

`\l@section` `\l@section{<title>}{<page>}` typesets the ToC entry for a section heading. It is a parameterised copy of the default `\l@section` (see `classes.dtx` for the original definition).

```

380 \renewcommand*{\l@section}[2]{%
381   \ifnum \c@tocdepth >\z@
382     \if@cfthaschapter
383       \vskip \cftbeforesecskip

```

```

384 \else
385 \addpenalty\@secpenalty
386 \advspace{\cftbeforesecskip}
387 \fi
388 {\leftskip \cftsecindent\relax
389 \rightskip \@tocrmarg
390 \parfillskip -\rightskip
391 \parindent \cftsecindent\relax\@afterindenttrue
392 \interlinepenalty\@M
393 \leavevmode
394 \@tempdima \cftsecnumwidth\relax
395 \let\@cftbsnum \cftsecpresnum
396 \let\@cftasnum \cftsecaftersnum
397 \let\@cftasnumb \cftsecaftersnumb
398 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
399 \UseHookWithArguments{contentsline/text/before}{4}
400 {1}{#1}{#2}{\@contentsline@destination}%
401 {\cftsecfont #1}\nobreak
402 \UseHookWithArguments{contentsline/text/after}{4}
403 {1}{#1}{#2}{\@contentsline@destination}%
404 \cftsecfillnum{#2}}%
405 \fi}

```

(End of definition for \l@section.)

`\cftbeforesecskip` These are the user commands to control the typesetting of Section entries. They are initialised to give the standard appearance.

```

\cftsecindent
\cftsecnumwidth 406 \newlength{\cftbeforesecskip}
\cftsecfont      407 \newlength{\cftsecindent}
\cftsecpresnum   408 \newlength{\cftsecnumwidth}
\cftsecaftersnum 409 \newcommand{\cftsecpresnum}{}
\cftsecaftersnumb 410 \newcommand{\cftsecaftersnum}{}
\cftsecleader    411 \newcommand{\cftsecaftersnumb}{}
\cftsecdotsep    412 \if@cfhaschapter
\cftsecpagefont  413 \setlength{\cftbeforesecskip}{1.0em \@plus\p@}
\cftsecafterpnum 414 \setlength{\cftsecindent}{1.5em}
\cftsecnumfinal  415 \setlength{\cftsecnumwidth}{2.3em}
\cftsecfillnum   416 \newcommand{\cftsecfont}{\normalfont}
417 \newcommand{\cftsecleader}{\normalfont\cftdotfill{\cftsecdotsep}}
418 \newcommand{\cftsecdotsep}{\cftdotsep}
419 \newcommand{\cftsecpagefont}{\normalfont}
420 \else
421 \setlength{\cftbeforesecskip}{1.0em \@plus\p@}
422 \setlength{\cftsecindent}{0em}
423 \setlength{\cftsecnumwidth}{1.5em}
424 \newcommand{\cftsecfont}{\bfseries}
425 \newcommand{\cftsecleader}{\bfseries\cftdotfill{\cftsecdotsep}}
426 \newcommand{\cftsecdotsep}{\cftnodots}
427 \newcommand{\cftsecpagefont}{\bfseries}
428 \fi
429 \newcommand{\cftsecafterpnum}{}
430 \newcommand{\cftsecnumfinal}{\cftpnumfinaldefault}
431 \newcommand{\cftsecfillnum}[1]{%
432 {%

```

```

433 \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
434 \cftsecleader
435 \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
436 }\nobreak
437 \UseHookWithArguments{contentsline/page/before}{4}
438 {1}{#1}{\@contentsline@destination}%
439 \makebox[\@pnumwidth][\cftpnumalign]{\cftsecpagefont #1}%
440 \UseHookWithArguments{contentsline/page/after}{4}
441 {1}{#1}{\@contentsline@destination}%
442 \cftsecafterpnum\cftsecpnumfinal\par
443 }

```

(End of definition for `\cftbeforesecskip` and others. These functions are documented on page 9.)

`\l@section` `\l@section{<title>}{<page>}` typesets the ToC entry for a subsection heading. It is a parameterised copy of the default `\l@section` (see `classes.dtx` for the original definition).

```

444 \renewcommand*{\l@section}[2]{%

```

Only typeset the entry if it falls within the `tocdepth`.

```

445 \ifnum \c@tocdepth >\@ne

```

Add some vertical space.

```

446 \vskip \cftbeforesubsecskip

```

Start a group to keep paragraphing changes local. Set the `\leftskip` to the entry's indentation.

```

447 {\leftskip \cftsubsecindent\relax

```

Set the `\rightskip` to `\@tocrmarg` to leave room for the page number.

```

448 \rightskip \@tocrmarg

```

Ensure that the last line of the entry will be filled. Setting `\parfillskip` to a negative number prevents any overfull box messages.

```

449 \parfillskip -\rightskip

```

Set the paragraph indent to the entry's indentation.

```

450 \parindent \cftsubsecindent\relax\@afterindenttrue

```

Try and prevent breaks between lines in a multiple line entry.

```

451 \interlinepenalty\@M

```

Make sure that we have left vertical mode.

```

452 \leavevmode

```

Our version of `\numberline` expects that the width of the number box is in `\@tempdima`, and that the three macros `\cftbsnum`, `\cftasnum` and `\cftasnumb` are defined. We set all these to the values for this entry.

```

453 \@tempdima \cftsubsecnumwidth\relax
454 \let\cftbsnum \cftsubsecpresnum
455 \let\cftasnum \cftsubsecaftersnum
456 \let\cftasnumb \cftsubsecaftersnumb

```

Arrange that the (section number and) first line of the title is set at the current indent, and any further lines are further indented.

```
457      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
```

Print the (number and) title, prohibiting any breaking.

```
458      \UseHookWithArguments{contentsline/text/before}{4}{2}{#1}{#2}{\@contentsline@destination}%
459      {\cftsubsecfont #1}\nobreak
460      \UseHookWithArguments{contentsline/text/after}{4}{2}{#1}{#2}{\@contentsline@destination}%
```

Print the leader and the page number, and close the group.

```
461      \cftsubsecfillnum{#2}}%
462      \fi}
```

(End of definition for \l@section.)

\cftbeforesubsecskip These are the user commands to control the typesetting of Sub-section entries. They are initialised to give the standard appearance.

```
\cftsubsecindent
\cftsubsecnumwidth
\cftsubsecfont
\cftsubsecpresnum
\cftsubsecaftersnum
\cftsubsecaftersnumb
\cftsubsecleader
\cftsubsecdotsep
\cftsubsecpagefont
\cftsubsecafterpnum
\cftsubsecpnumfinal
463 \newlength{\cftbeforesubsecskip}
464 \setlength{\cftbeforesubsecskip}{\z@ \@plus.2\p@}
465 \newlength{\cftsubsecindent}
466 \newlength{\cftsubsecnumwidth}
467 \if@cfthaschapter
468 \setlength{\cftsubsecindent}{3.8em}
469 \setlength{\cftsubsecnumwidth}{3.2em}
470 \else
471 \setlength{\cftsubsecindent}{1.5em}
472 \setlength{\cftsubsecnumwidth}{2.3em}
473 \fi
474 \newcommand{\cftsubsecfont}{\normalfont}
475 \newcommand{\cftsubsecpresnum}{\@empty}
476 \newcommand{\cftsubsecaftersnum}{\@empty}
477 \newcommand{\cftsubsecaftersnumb}{\@empty}
478 \newcommand{\cftsubsecleader}{\normalfont\cftdotfill{\cftsubsecdotsep}}
479 \newcommand{\cftsubsecdotsep}{\cftdotsep}
480 \newcommand{\cftsubsecpagefont}{\normalfont}
481 \newcommand{\cftsubsecafterpnum}{\@empty}
482 \newcommand{\cftsubsecpnumfinal}{\cftpnumfinaldefault}
```

(End of definition for \cftbeforesubsecskip and others. These functions are documented on page 9.)

\cftsubsecfillnum `\cftsubsecfillnum{⟨page⟩}` typesets the leader and the ⟨page⟩ number of a subsection entry. First print the leader and then, with no break, set the page number flushright in a box of width `\@pnumwidth`, not forgetting to finish the paragraph.

```
483 \newcommand{\cftsubsecfillnum}[1]{%
484   {%
485     \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
486     \cftsubsecleader
487     \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
488   }\nobreak
489   \UseHookWithArguments{contentsline/page/before}{4}
490   {2}{#1}{\@contentsline@destination}%
491   \makebox[\@pnumwidth]{\cftpnumalign{\cftsubsecpagefont #1}%
492   \UseHookWithArguments{contentsline/page/after}{4}
493   {2}{#1}{\@contentsline@destination}%
```

```

494 \cftsubsecafterpnum\cftsubsecpnumfinal\par
495 }

```

(End of definition for \cftsubsecfillnum. This function is documented on page 10.)

`\l@subsubsection` `\l@subsubsection{<title>}{<page>}` typesets the ToC entry for a subsubsection heading. It is a parameterised copy of the default `\l@subsubsection` (see `classes.dtx` for the original definition).

```

496 \renewcommand*{\l@subsubsection}[2]{%
497 \ifnum \c@tocdepth >\tw@
498 \vskip \cftbeforesubsubsecskip
499 {\leftskip \cftsubsubsecindent\relax
500 \rightskip \@tocrmarg
501 \parfillskip -\rightskip
502 \parindent \cftsubsubsecindent\relax\@afterindenttrue
503 \interlinepenalty\@M
504 \leavevmode
505 \@tempdima \cftsubsubsecnumwidth\relax
506 \let\@cftbsnum \cftsubsubsecpresnum
507 \let\@cftasnum \cftsubsubsecaftersnum
508 \let\@cftasnumb \cftsubsubsecaftersnumb
509 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
510 \UseHookWithArguments{contentsline/text/before}{4}{3}{#1}{#2}{\@contentsline@destination}
511 {\cftsubsubsecfont #1}\nobreak
512 \UseHookWithArguments{contentsline/text/after}{4}{3}{#1}{#2}{\@contentsline@destination}
513 \cftsubsubsecfillnum{#2}}%
514 \fi}

```

(End of definition for \l@subsubsection.)

`\cftbeforesubsubsecskip` These are the user commands to control the typesetting of Sub-sub-section entries. They are initialised to give the standard appearance.

```

\cftsubsubsecindent
\cftsubsubsecnumwidth
\cftsubsubsecfont
\cftsubsubsecpresnum
\cftsubsubsecaftersnum
\cftsubsubsecaftersnumb
\cftsubsubsecleader
\cftsubsubsecdotsep
\cftsubsubsecpagefont
\cftsubsubsecafterpnum
\cftsubsubsecpnumfinal
\cftsubsubsecfillnum
515 \newlength{\cftbeforesubsubsecskip}
516 \setlength{\cftbeforesubsubsecskip}{\z@ \@plus.2\p@}
517 \newlength{\cftsubsubsecindent}
518 \newlength{\cftsubsubsecnumwidth}
519 \if@cfthaschapter
520 \setlength{\cftsubsubsecindent}{7.0em}
521 \setlength{\cftsubsubsecnumwidth}{4.1em}
522 \else
523 \setlength{\cftsubsubsecindent}{3.8em}
524 \setlength{\cftsubsubsecnumwidth}{3.2em}
525 \fi
526 \newcommand{\cftsubsubsecfont}{\normalfont}
527 \newcommand{\cftsubsubsecpresnum}{}
528 \newcommand{\cftsubsubsecaftersnum}{}
529 \newcommand{\cftsubsubsecaftersnumb}{}
530 \newcommand{\cftsubsubsecleader}{\normalfont\cftdotfill{\cftsubsubsecdotsep}}
531 \newcommand{\cftsubsubsecdotsep}{\cftdotsep}
532 \newcommand{\cftsubsubsecpagefont}{\normalfont}
533 \newcommand{\cftsubsubsecafterpnum}{}
534 \newcommand{\cftsubsubsecpnumfinal}{\cftpnumfinaldefault}
535 \newcommand{\cftsubsubsecfillnum}[1]{%
536 {%

```

```

537 \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
538 \cftsubsubsecleader
539 \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
540 }\nobreak
541 \UseHookWithArguments{contentsline/page/before}{4}
542 {3}{\#1}{\@contentsline@destination}%
543 \makebox[\@pnumwidth][\cftpnumalign]{\cftsubsubsecpagefont #1}%
544 \UseHookWithArguments{contentsline/page/after}{4}
545 {3}{\#1}{\@contentsline@destination}%
546 \cftsubsubsecafterpnum\cftsubsubsecpnumfinal\par
547 }

```

(End of definition for `\cftbeforesubsubsecskip` and others. These functions are documented on page 9.)

`\l@paragraph` `\l@paragraph{<title>}{<page>}` typesets the ToC entry for a paragraph heading. It is a parameterised copy of the default `\l@paragraph` (see `classes.dtx` for the original definition).

```

548 \renewcommand*{\l@paragraph}[2]{%
549 \ifnum \c@tocdepth >3\relax
550 \vskip \cftbeforeparaskip
551 {\leftskip \cftparaindent\relax
552 \rightskip \@tocrmarg
553 \parfillskip -\rightskip
554 \parindent \cftparaindent\relax\@afterindenttrue
555 \interlinepenalty\@M
556 \leavevmode
557 \@tempdima \cftparanumwidth\relax
558 \let\@cftbsnum \cftparapresnum
559 \let\@cftasnum \cftparaaftersnum
560 \let\@cftasnumb \cftparaaftersnumb
561 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
562 \UseHookWithArguments{contentsline/text/before}{4}
563 {4}{\#1}{\#2}{\@contentsline@destination}%
564 {\cftparafont #1}\nobreak
565 \UseHookWithArguments{contentsline/text/after}{4}
566 {4}{\#1}{\#2}{\@contentsline@destination}%
567 \cftparafillnum{\#2}}%
568 \fi}

```

(End of definition for `\l@paragraph`.)

`\cftbeforeparaskip` `\cftparaindent` These are the user commands to control the typesetting of Paragraph entries. They are initialised to give the standard appearance.

```

\cftparanumwidth
\cftparafont
\cftparapresnum
\cftparaaftersnum
\cftparaaftersnumb
\cftparaleader
\cftparadotsep
\cftparapagefont
\cftparaafterpnum
\cftparapnumfinal
\cftparafillnum
569 \newlength{\cftbeforeparaskip}
570 \setlength{\cftbeforeparaskip}{\z@ \@plus.2\p@}
571 \newlength{\cftparaindent}
572 \newlength{\cftparanumwidth}
573 \if@cfthaschapter
574 \setlength{\cftparaindent}{10em}
575 \setlength{\cftparanumwidth}{5em}
576 \else
577 \setlength{\cftparaindent}{7.0em}
578 \setlength{\cftparanumwidth}{4.1em}
579 \fi

```

```

580 \newcommand{\cftparafont}{\normalfont}
581 \newcommand{\cftparapresnum}{\cftparapresnum}
582 \newcommand{\cftparaaftersnum}{\cftparaaftersnum}
583 \newcommand{\cftparaaftersnumb}{\cftparaaftersnumb}
584 \newcommand{\cftparaleader}{\normalfont\cftdotfill{\cftparadotsep}}
585 \newcommand{\cftparadotsep}{\cftdotsep}
586 \newcommand{\cftparapagefont}{\normalfont}
587 \newcommand{\cftparaafterpnum}{\cftparaafterpnum}
588 \newcommand{\cftparapnumfinal}{\cftpnumfinaldefault}
589 \newcommand{\cftparafillnum}[1]{%
590   {%
591     \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
592     \cftparaleader
593     \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
594   }\nobreak
595   \UseHookWithArguments{contentsline/page/before}{4}
596     {4}{\cftparafont}{\cftparapresnum}{\cftparaaftersnum}{\cftparaaftersnumb}
597     \makebox[\cftparapnumwidth][\cftpnumalign]{\cftparapagefont #1}%
598   \UseHookWithArguments{contentsline/page/after}{4}
599     {4}{\cftparafont}{\cftparapresnum}{\cftparaaftersnum}{\cftparaaftersnumb}
600   \cftparaafterpnum\cftparapnumfinal\par
601 }

```

(End of definition for \cftbeforeparaskip and others. These functions are documented on page 9.)

`\l@subparagraph` `\l@subparagraph{<title>}{<page>}` typesets the ToC entry for a subparagraph heading. It is a parameterised copy of the default `\l@subparagraph` (see `classes.dtx` for the original definition).

```

602 \renewcommand*\l@subparagraph[2]{%
603   \ifnum \c@tocdepth >4\relax
604     \vskip \cftbeforesubparaskip
605     {\leftskip \cftsubparaindent\relax
606       \rightskip \@tocrmarg
607       \parfillskip -\rightskip
608       \parindent \cftsubparaindent\relax\@afterindenttrue
609       \interlinepenalty\@M
610       \leavevmode
611       \@tempdima \cftsubparanumwidth\relax
612       \let\@cftbsnum \cftsubparapresnum
613       \let\@cftasnum \cftsubparaaftersnum
614       \let\@cftasnumb \cftsubparaaftersnumb
615       \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
616       \UseHookWithArguments{contentsline/text/before}{4}
617         {5}{\cftbsnum}{\cftasnum}{\cftasnumb}{\cftparafont}{\cftparapresnum}{\cftparaaftersnum}{\cftparaaftersnumb}
618       {\cftsubparafont #1}\nobreak
619       \UseHookWithArguments{contentsline/text/after}{4}
620         {5}{\cftbsnum}{\cftasnum}{\cftasnumb}{\cftparafont}{\cftparapresnum}{\cftparaaftersnum}{\cftparaaftersnumb}
621       \cftsubparafillnum{#2}}%
622   \fi}

```

(End of definition for \l@subparagraph.)

`\cftbeforesubparaskip` `\cftsubparaindent` `\cftsubparanumwidth` `\cftsubparafont` `\cftsubparapresnum` `\cftsubparaaftersnum` `\cftsubparaaftersnumb` `\cftsubparaleader` `\cftsubparadotsep` `\cftsubparapagefont` `\cftsubparaafterpnum` `\cftsubparapnumfinal` `\cftsubparafillnum` These are the user commands to control the typesetting of Sub-paragraph entries. They are initialised to give the standard appearance.

```

623 \newlength{\cftbeforesubparaskip}

```

```

624 \setlength{\cftbeforesubparaskip}{\z@ \@plus.2\p@}
625 \newlength{\cftsubparaindent}
626 \newlength{\cftsubparanumwidth}
627 \if@cfthaschapter
628 \setlength{\cftsubparaindent}{12em}
629 \setlength{\cftsubparanumwidth}{6em}
630 \else
631 \setlength{\cftsubparaindent}{10em}
632 \setlength{\cftsubparanumwidth}{5em}
633 \fi
634 \newcommand{\cftsubparafont}{\normalfont}
635 \newcommand{\cftsubparapresnum}{}
636 \newcommand{\cftsubparaaftersnum}{}
637 \newcommand{\cftsubparaaftersnumb}{}
638 \newcommand{\cftsubparaaleader}{\normalfont\cftdotfill{\cftsubparadotsep}}
639 \newcommand{\cftsubparadotsep}{\cftdotsep}
640 \newcommand{\cftsubparapagefont}{\normalfont}
641 \newcommand{\cftsubparaafterpnum}{}
642 \newcommand{\cftsubparapnumfinal}{\cftpnumfinaldefault}
643 \newcommand{\cftsubparafillnum}[1]{%
644   {%
645     \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
646     \cftsubparaaleader
647     \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
648   }\nobreak
649   \UseHookWithArguments{contentsline/page/before}{4}
650   {5}{\cftsubparafillnum}{\cftsubparanumwidth}{\cftsubparapagefont #1}%
651   \makebox[\cftsubparanumwidth]{\cftsubparapagefont #1}%
652   \UseHookWithArguments{contentsline/page/after}{4}
653   {5}{\cftsubparafillnum}{\cftsubparanumwidth}{\cftsubparapagefont #1}%
654   \cftsubparaafterpnum\cftsubparapnumfinal\par
655 }

```

(End of definition for \cftbeforesubparaskip and others. These functions are documented on page 9.)

\cftdobiblof If the tocibind package has been used and it has redefined \listoffigures we need to cater for that. The contents of the definition are defined in tocibind.

```

656 \newcommand{\cftdobiblof}{%
657 \if@dotoclof
658 \if@bibchapter
659 \phantomsection
660 \addcontentsline{toc}{chapter}{\cftloftitlename}
661 \else
662 \phantomsection
663 \addcontentsline{toc}{\@tocextra}{\cftloftitlename}
664 \fi
665 \fi}

```

(End of definition for \cftdobiblof.)

\listoffigures This is a parameterised version of the default \listoffigures command. The changes are postponed until after all packages have been loaded. Each class has its own definition, but we have to cater for all classes in one definition, hence some of the checks. First, perform the default checks for multicolumns. (Do nothing if titles option is used).

```

666 \AtBeginDocument{%

```

```

667 \if@cftnctoc\else
668 \renewcommand{\listoffigures}{%
669 \cfttocstart

```

Ensure that any previous paragraph has been finished. Within a group set the local paragraphing style. Typeset the title and then do the contents of the .lof file.

```

670 \par
671 \begingroup
672 \parindent\z@ \parskip\cftparskip
673 \@cftmakeloftitle
674 \if@cfttocbibind
675 \cftdobiblof
676 \fi
677 \@starttoc{lof}%
678 \endgroup

```

Finally, restore any multicolumn typesetting.

```

679 \@cfttocfinish}%
680 \fi
681 }
682

```

(End of definition for \listoffigures. This function is documented on page 7.)

`\@cftmakeloftitle` This command typesets the title for the LoF.

```

683 \newcommand{\@cftmakeloftitle}{\cfttitlecommand{lof}}%

```

(End of definition for \@cftmakeloftitle.)

`\cftbeforeloftitleskip` These two lengths control the vertical spacing before and after the LoF title.

```

\cftafterloftitleskip
684 \newlength{\cftbeforeloftitleskip}
685 \newlength{\cftafterloftitleskip}

```

Their values depend on whether the document has chapters or not. In chaptered documents the default LoF title is typeset as a `\chapter*`, otherwise as a `\section*`.

```

686 \if@cfthaschapter
687 \setlength{\cftbeforeloftitleskip}{50pt}
688 \setlength{\cftafterloftitleskip}{40pt}
689 \else
690 \setlength{\cftbeforeloftitleskip}{3.5ex \@plus 1ex \@minus .2ex}
691 \setlength{\cftafterloftitleskip}{2.3ex \@plus .2ex}
692 \fi

```

(End of definition for \cftbeforeloftitleskip and \cftafterloftitleskip. These functions are documented on page 7.)

`\cftloftitlefont` The LoF title is typeset in the style given by `\cftloftitlefont`. The macro
`\cftafterloftitle` `\cftafterloftitle` is called after typesetting the title. This is initialised to do nothing. Both these macros can be redefined to do other things (e.g., adding an `\hfill` to `\cftloftitlefont` will make the title flushright).

```

693 \if@cfthaschapter
694 \newcommand{\cftloftitlefont}{\normalfont\Huge\bfseries}
695 \if@cftekoma\renewcommand{\cftloftitlefont}{\size@chapter\sectfont}\fi
696 \else
697 \newcommand{\cftloftitlefont}{\normalfont\Large\bfseries}

```

```

698 \if@ctkoma\renewcommand{\cftloftitlefont}{\size@section\sectfont}\fi
699 \fi
700 \newcommand{\cftafterloftitle}{}
701

```

(End of definition for `\cftloftitlefont` and `\cftafterloftitle`. These functions are documented on page 8.)

`\l@figure` `\l@figure{<title>}{<page>}` typesets the LoF entry for a figure caption heading. It is a parameterised copy of the default `\l@figure` (see `classes.dtx` for the original definition).

```

702 \renewcommand*{\l@figure}[2]{%
703   \ifnum \c@lofdepth >\z@
704     \vskip \cftbeforefigskip
705     {\leftskip \cftfigindent\relax
706      \rightskip \@tocrmarg
707      \parfillskip -\rightskip
708      \parindent \cftfigindent\relax\@afterindenttrue
709      \interlinepenalty\@M
710      \leavevmode
711      \@tempdima \cftfignumwidth\relax
712      \let\@cftbsnum \cftfigpresnum
713      \let\@cftasnum \cftfigaftersnum
714      \let\@cftasnumb \cftfigaftersnumb
715      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
716      \UseHookWithArguments{contentsline/text/before}{4}
717       {1}{#1}{#2}{\@contentsline@destination}%
718       {\cftfigfont #1}\nobreak
719      \UseHookWithArguments{contentsline/text/after}{4}
720       {1}{#1}{#2}{\@contentsline@destination}%
721      \cftfigfillnum{#2}}%
722   \fi
723 }

```

(End of definition for `\l@figure`.)

`\cftbeforefigskip` `\cftfigindent` `\cftfignumwidth` `\cftfigfont` `\cftfigpresnum` `\cftfigaftersnum` `\cftfigaftersnumb` `\cftfigleader` `\cftfigdotsep` `\cftfigpagefont` `\cftfigafterpnum` `\cftfigpnumfinal` `\cftfigfillnum` These are the user commands to control the typesetting of Figure caption entries. They are initialised to give the standard appearance.

```

724 \newlength{\cftbeforefigskip}
725 \setlength{\cftbeforefigskip}{\z@ \@plus.2\p@}
726 \newlength{\cftfigindent}
727 \setlength{\cftfigindent}{1.5em}
728 \newlength{\cftfignumwidth}
729 \setlength{\cftfignumwidth}{2.3em}
730 \newcommand{\cftfigfont}{\normalfont}
731 \newcommand{\cftfigpresnum}{}
732 \newcommand{\cftfigaftersnum}{}
733 \newcommand{\cftfigaftersnumb}{}
734 \newcommand{\cftfigleader}{\normalfont\cftdotfill{\cftfigdotsep}}
735 \newcommand{\cftfigdotsep}{\cftdotsep}
736 \newcommand{\cftfigpagefont}{\normalfont}
737 \newcommand{\cftfigafterpnum}{}
738 \newcommand{\cftfigpnumfinal}{\cftpnumfinaldefault}
739 \newcommand{\cftfigfillnum}[1]{%
740   {%

```

```

741 \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
742 \cftfigleader
743 \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
744 }\nobreak
745 \UseHookWithArguments{contentsline/page/before}{4}
746 {1}{#1}{\@contentsline@destination}%
747 \makebox[\@pnumwidth][\cftpnumalign]{\cftfigpagefont #1}%
748 \UseHookWithArguments{contentsline/page/after}{4}
749 {1}{#1}{\@contentsline@destination}%
750 \cftfigafterpnum\cftfigpnumfinal\par
751 }

```

(End of definition for \cftbeforefigskip and others. These functions are documented on page 9.)

lofdepth The counters lofdepth and lotdepth are defined by the subfigure package. Define them here if that package is not used.

```

752 \if@cftsubfigopt\else
753 \newcounter{lofdepth}\setcounter{lofdepth}{1}
754 \newcounter{lotdepth}\setcounter{lotdepth}{1}
755 \fi
756

```

(End of definition for lofdepth and lotdepth. These functions are documented on page 8.)

\cftdobiblot If the tocbibind package has been used and it has redefined \listoftables we need to cater for that. The contents of the definition are defined in tocbibind.

```

757 \newcommand{\cftdobiblot}{%
758 \if@dotoclot
759 \if@bibchapter
760 \phantomsection
761 \addcontentsline{toc}{chapter}{\cftlottitlename}
762 \else
763 \phantomsection
764 \addcontentsline{toc}{\@tocextra}{\cftlottitlename}
765 \fi
766 \fi}
767

```

(End of definition for \cftdobiblot.)

\listoftables This is a parameterised version of the default \listoftables command. The changes are postponed until after all packages have been loaded. Each class has its own definition, but we have to cater for all classes in one definition, hence some of the checks. First, perform the default checks for multicolumns. (Do nothing if the titles option has been used).

```

768 \AtBeginDocument{%
769 \if@cftnctoc\else
770 \renewcommand{\listoftables}{%
771 \cfttocstart

```

Ensure that any previous paragraph has been finished. Within a group set the local paragraphing style. Typeset the title and then do the contents of the .lot file.

```

772 \par
773 \begingroup

```

```

774 \parindent\z@ \parskip\cftparskip
775 \@cftmakelottitle
776 \if@cfttocbibind
777 \cftdobiblot
778 \fi
779 \@starttoc{lot}%
780 \endgroup

```

Finally, restore any multicolumn typesetting.

```

781 \cfttocfinish}%
782 \fi
783 }
784

```

(End of definition for \listoftables. This function is documented on page 7.)

```

\@cftmakelottitle This command typesets the title for the LoT.
785 \newcommand{\@cftmakelottitle}{\cfttitlecommand{lot}}%

```

(End of definition for \@cftmakelottitle.)

```

\cftbeforelottitleskip These two lengths control the vertical spacing before and after the LoT title.
\cftafterlottitleskip
786 \newlength{\cftbeforelottitleskip}
787 \newlength{\cftafterlottitleskip}

```

Their values depend on whether the document has chapters or not. In chaptered documents the default LoT title is typeset as a `\chapter*`, otherwise as a `\section*`.

```

788 \if@cftaschapter
789 \setlength{\cftbeforelottitleskip}{50pt}
790 \setlength{\cftafterlottitleskip}{40pt}
791 \else
792 \setlength{\cftbeforelottitleskip}{3.5ex \@plus 1ex \@minus .2ex}
793 \setlength{\cftafterlottitleskip}{2.3ex \@plus .2ex}
794 \fi

```

(End of definition for \cftbeforelottitleskip and \cftafterlottitleskip. These functions are documented on page 7.)

```

\cftlottitlefont The LoT title is typeset in the style given by \cftlottitlefont. The macro
\cftafterlottitle \cftafterlottitle is called after typesetting the title. This is initialised to do nothing. Both these macros can be redefined to do other things (e.g., adding an \hfill to \cftlottitlefont will make the title flushright).

```

```

795 \if@cftaschapter
796 \newcommand{\cftlottitlefont}{\normalfont\Huge\bfseries}
797 \if@cftkoma\renewcommand{\cftlottitlefont}{\size@chapter\sectfont}\fi
798 \else
799 \newcommand{\cftlottitlefont}{\normalfont\Large\bfseries}
800 \if@cftkoma\renewcommand{\cftlottitlefont}{\size@section\sectfont}\fi
801 \fi
802 \newcommand{\cftafterlottitle}{}
803

```

(End of definition for \cftlottitlefont and \cftafterlottitle. These functions are documented on page 8.)

`\l@table` `\l@table{<title>}{<page>}` typesets the LoT entry for a table caption heading. It is a parameterised copy of the default `\l@table` (see `classes.dtx` for the original definition).

```

804 \renewcommand*{\l@table}[2]{%
805   \ifnum\c@lotdepth >\z@
806     \vskip \cftbeforetabskip
807     {\leftskip \cfttabindent\relax
808      \rightskip \@tocrmarg
809      \parfillskip -\rightskip
810      \parindent \cfttabindent\relax\@afterindenttrue
811      \interlinepenalty\@M
812      \leavevmode
813      \@tempdima \cfttabnumwidth\relax
814      \let\@cftbsnum \cfttabpresnum
815      \let\@cftasnum \cfttabaftersnum
816      \let\@cftasnumb \cfttabaftersnumb
817      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
818      \UseHookWithArguments{contentsline/text/before}{4}
819       {1}{#1}{#2}{\@contentsline@destination}%
820       {\cfttabfont #1}\nobreak
821      \UseHookWithArguments{contentsline/text/after}{4}
822       {1}{#1}{#2}{\@contentsline@destination}%
823      \cfttabfillnum{#2}}%
824   \fi
825 }
```

(End of definition for `\l@table`.)

`\cftbeforetabskip` `\cfttabindent` `\cfttabnumwidth` `\cfttabfont` `\cfttabpresnum` `\cfttabaftersnum` `\cfttabaftersnumb` `\cfttableader` `\cfttabdotsep` `\cfttabpagefont` `\cfttabafterpnum` `\cfttabpnumfinal` `\cfttabfillnum` These are the user commands to control the typesetting of Table caption entries. They are initialised to give the standard appearance.

```

826 \newlength{\cftbeforetabskip}
827 \setlength{\cftbeforetabskip}{\z@ \@plus.2\p@}
828 \newlength{\cfttabindent}
829 \setlength{\cfttabindent}{1.5em}
830 \newlength{\cfttabnumwidth}
831 \setlength{\cfttabnumwidth}{2.3em}
832 \newcommand{\cfttabfont}{\normalfont}
833 \newcommand{\cfttabpresnum}{}
834 \newcommand{\cfttabaftersnum}{}
835 \newcommand{\cfttabaftersnumb}{}
836 \newcommand{\cfttableader}{\normalfont\cftdotfill{\cfttabdotsep}}
837 \newcommand{\cfttabdotsep}{\cftdotsep}
838 \newcommand{\cfttabpagefont}{\normalfont}
839 \newcommand{\cfttabafterpnum}{}
840 \newcommand{\cfttabpnumfinal}{\cftpnumfinaldefault}
841 \newcommand{\cfttabfillnum}[1]{%
842   {%
843     \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
844     \cfttableader
845     \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
846   }\nobreak
847   \UseHookWithArguments{contentsline/page/before}{4}
848    {1}{#1}{\@contentsline@destination}%
849   \makebox[\@pnumwidth]{\cftpnumalign}{\cfttabpagefont #1}%
850   \UseHookWithArguments{contentsline/page/after}{4}
```

```

851 {1}{1}{\@contentsline@destination}%
852 \cfttabafterpnum\cfttabpnumfinal\par
853 }

```

(End of definition for `\cftbeforetabskip` and others. These functions are documented on page 9.)

3.1 Support for the subfigure package

The code for supporting the subfigure package is, in all essentials, the same as that for the figure and table captions; only the names are changed. However, the code need only be executed if the subfigure package is actually loaded.

`\@cftl@subfig` This command redefines the `\l@subfigure` command.

```

854 \newcommand{\@cftl@subfig}{%
\l@subfigure{<title>}{<page>}} typesets the LoF entry for a subfigure caption head-
\l@subfigure ing. It is essentially the same as the parameterised code for \l@figure except that
account has to be taken of lofdepth.
855 \renewcommand*{\l@subfigure}[2]{%
856 \ifnum \toclevel@subfigure > \c@lofdepth
857 \else
858 \vskip \cftbeforesubfigskip
859 {\leftskip \cftsubfigindent\relax
860 \rightskip \@tocrmarg
861 \parfillskip -\rightskip
862 \parindent \cftsubfigindent\relax\@afterindenttrue
863 \interlinepenalty\@M
864 \leavevmode
865 \@tempdima \cftsubfignumwidth\relax
866 \let\@cftbsnum \cftsubfigpresnum
867 \let\@cftasnum \cftsubfigaftersnum
868 \let\@cftasnumb \cftsubfigaftersnumb
869 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
870 \UseHookWithArguments{contentsline/text/before}{4}
871 {\toclevel@subfigure}{##1}{##2}{\@contentsline@destination}%
872 {\cftsubfigfont ##1}\nobreak
873 \UseHookWithArguments{contentsline/text/after}{4}
874 {\toclevel@subfigure}{##1}{##2}{\@contentsline@destination}%
875 \cftsubfigfillnum{##2}}%
876 \fi
877 }%
878 }
879

```

(End of definition for `\@cftl@subfig` and `\l@subfigure`.)

`\@cftsetsubfig` This command initialises the setup for subfigure captions in the LoF.

```

880 \newcommand{\@cftsetsubfig}{%
\cftbeforesubfigskip 881 \newlength{\cftbeforesubfigskip}
\cftsubfigindent 882 \setlength{\cftbeforesubfigskip}{\z@ \@plus.2\p@}
\cftsubfignumwidth 883 \newlength{\cftsubfigindent}
\cftsubfigfont 884 \setlength{\cftsubfigindent}{3.8em}
\cftsubfigpresnum
\cftsubfigaftersnum
\cftsubfigaftersnumb
\cftsubfigleader
\cftsubfigdotsep
\cftsubfigpagefont
\cftsubfigafterpnum
\cftsubfigpnumfinal
\toclevel@subfig
\cftsubfigfillnum

```

```

885 \newlength{\cftsubfignumwidth}
886 \setlength{\cftsubfignumwidth}{2.5em}
887 \newcommand{\cftsubfigfont}{\normalfont}
888 \newcommand{\cftsubfigpresnum}{\}
889 \newcommand{\cftsubfigaftersnum}{\}
890 \newcommand{\cftsubfigaftersnumb}{\}
891 \newcommand{\cftsubfigleader}{\normalfont\cftdotfill{\cftsubtabdotsep}}
892 \newcommand{\cftsubfigdotsep}{\cftdotsep}
893 \newcommand{\cftsubfigpagefont}{\normalfont}
894 \newcommand{\cftsubfigafterpnum}{\}
895 \newcommand{\cftsubfigpnumfinal}{\cftpnumfinaldefault}
896 \providecommand{\toclevel@subfigure}{1}
897 \newcommand{\cftsubfigfillnum}[1]{%
898   {%
899     \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
900     \cftsubfigleader
901     \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
902   }\nobreak
903   \UseHookWithArguments{contentsline/page/before}{4}
904     {\toclevel@subfigure}{\}{{#1}}{\@contentsline@destination}%
905   \makebox[\@pnumwidth]{\cftpnumalign}{\cftsubfigpagefont #1}%
906   \UseHookWithArguments{contentsline/page/after}{4}
907     {\toclevel@subfigure}{\}{{#1}}{\@contentsline@destination}%
908   \cftsubfigafterpnum\cftsubfigpnumfinal\par
909 }

```

This is the end of \cftsetsubfig.

```

910 }
911

```

(End of definition for \cftsetsubfig and others.)

\cftl@subtab This code redefines the code for \l@subtable.

```

912 \newcommand{\cftl@subtab}{%

```

\l@subtable{<title>}{<page>} typesets the LoT entry for a subtable caption heading.

\l@subtable It is essentially the same as the parameterised code for \l@table except account has to be taken of lotdepth.

```

913 \renewcommand*{\l@subtable}[2]{%
914   \ifnum \toclevel@subtable > \c@lofdepth
915   \else
916     \vskip \cftbeforesubtabskip
917     {\leftskip \cftsubtabindent\relax
918       \rightskip \@tocrmarg
919       \parfillskip -\rightskip
920       \parindent \cftsubtabindent\relax\@afterindenttrue
921       \interlinepenalty\@M
922       \leavevmode
923       \@tempdima \cftsubtabnumwidth\relax
924       \let\@cftbsnum \cftsubtabpresnum
925       \let\@cftasnum \cftsubtabaftersnum
926       \let\@cftasnumb \cftsubtabaftersnumb
927       \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
928       \UseHookWithArguments{contentsline/text/before}{4}

```

```

929     {\toclevel@subtable}{##1}{##2}{\@contentsline@destination}%
930     {\cftsubtabfont ##1}\nobreak
931     \UseHookWithArguments{contentsline/text/after}{4}
932     {\toclevel@subtable}{##1}{##2}{\@contentsline@destination}%
933     \cftsubtabfillnum{##2}}%
934   \fi
935   }%
936 }

```

(End of definition for \@cftl@subtab and \l@subtable.)

\@cftsetsubtab This command sets up the defaults for subtable entries in the LoT.

```

937 \newcommand{\@cftsetsubtab}{%

```

These are the user commands to control the typesetting of Subtable caption entries. They are initialised to give the standard appearance.

```

\cftbeforesubtabskip 938 \newlength{\cftbeforesubtabskip}
\cftsubtabindent      939 \setlength{\cftbeforesubtabskip}{\z@ \@plus.2\p@}
\cftsubtabnumwidth    940 \newlength{\cftsubtabindent}
\cftsubtabfont        941 \setlength{\cftsubtabindent}{3.8em}
\cftsubtabpresnum     942 \newlength{\cftsubtabnumwidth}
\cftsubtabaftersnum   943 \setlength{\cftsubtabnumwidth}{2.5em}
\cftsubtabaftersnumb  944 \newcommand{\cftsubtabfont}{\normalfont}
\cftsubtableader      945 \newcommand{\cftsubtabpresnum}{\cftsubtabfont}
\cftsubtabdotsep      946 \newcommand{\cftsubtabaftersnum}{\cftsubtabfont}
\cftsubtabpagefont    947 \newcommand{\cftsubtabaftersnumb}{\cftsubtabfont}
\cftsubtabafterpnum   948 \newcommand{\cftsubtableader}{\normalfont\cftdotfill{\cftsubtabdotsep}}
\cftsubtabpnumfinal   949 \newcommand{\cftsubtabdotsep}{\cftdotsep}
\toclevel@subtable    950 \newcommand{\cftsubtabpagefont}{\normalfont}
\cftsubtabfillnum     951 \newcommand{\cftsubtabafterpnum}{\cftsubtabfont}
                      952 \newcommand{\cftsubtabpnumfinal}{\cftpnumfinaldefault}
                      953 \providecommand{\toclevel@subtable}{1}
                      954 \newcommand{\cftsubtagfillnum}[1]{%
                      955   {%
                      956     \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
                      957     \cftsubtagleader
                      958     \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
                      959   }\nobreak
                      960   \UseHookWithArguments{contentsline/page/before}{4}
                      961     {\toclevel@subtable}{\cftsubtagfillnum}{\@contentsline@destination}%
                      962   \makebox[\cftsubtabnumwidth][\cftpnumalign]{\cftsubtagpagefont ##1}%
                      963   \UseHookWithArguments{contentsline/page/after}{4}
                      964     {\toclevel@subtable}{\cftsubtagfillnum}{\@contentsline@destination}%
                      965   \cftsubtagafterpnum\cftsubtagpnumfinal\par
                      966 }

```

This is the end of \@cftsetsubtab.

```

967 }
968

```

(End of definition for \@cftsetsubtab and others.)

Call the subfigure package setup code only if the subfigure option is specified. The \l@... redefinitions have to come after the subfigure package is loaded.

```

969 \if@cftsubfigopt

```

```

970 \cftsetsubfig\cftsetsubtab
971 \AtBeginDocument{\cftl@subfig\cftl@subtab}
972 \fi
973 %% \AtBeginDocument{\ifcftsubfigopt
974 %% \cftsetsubfig\cftsetsubtab
975 %% \cftl@subfig\cftl@subtab
976 %% \fi}
977

```

3.2 New list of...

\newlistentry `\newlistentry[<within>]{<counter>}{<ext>}{<level-1>}` creates a set of commands for a new kind of entry into a List of.

```

978 \newcommand{\newlistentry}[4][\@empty]{%

```

Check if *<within>* and *<counter>* have been defined. It is an error if *<within>* has not been defined, and an error if *<counter>* has been defined. Set the default counter values.

```

\c@X \theX
979 \ifundefined{c@#2}{%      check & set the counter
980 \ifx \@empty#1\relax
981 \newcounter{#2}
982 \else
983 \ifundefined{c@#1}{\PackageWarning{tocloft}%
984 {#1 has no counter for use as a `within'}
985 \newcounter{#2}}%
986 {\newcounter{#2}[#1}%
987 \expandafter\edef\csname the#2\endcsname{%
988 \expandafter\noexpand\csname the#1\endcsname.\noexpand\arabic{#2}}}%
989 \fi
990 \setcounter{#2}{0}
991 }
992 {\PackageError{tocloft}{#2 has been previously defined}{\@eha}}
993

```

That finishes off the error checking. No matter what the result, the rest of the new commands are defined.

`\l@X{<title>}{<page>}` typesets the entry.

```

\l@X
994 \@namedef{l@#2}##1##2{%

```

Only typeset if the `\Zdepth` is greater than *<level-1>*.

```

995 \ifnum \@nameuse{c@#3depth} > #4\relax
996 \vskip \@nameuse{cftbefore#2skip}
997 {\leftskip \@nameuse{cft#2indent}\relax
998 \rightskip \@tocrmarg
999 \parfillskip -\rightskip
1000 \parindent \@nameuse{cft#2indent}\relax\@afterindenttrue
1001 \interlinepenalty\@M
1002 \leavevmode
1003 \@tempdima \@nameuse{cft#2numwidth}\relax
1004 \expandafter\let\expandafter\cftbsnum\csname cft#2presnum\endcsname
1005 \expandafter\let\expandafter\cftasnum\csname cft#2aftersnum\endcsname
1006 \expandafter\let\expandafter\cftasnumb\csname cft#2aftersnumb\endcsname
1007 \advance\leftskip\@tempdima \null\nobreak\hskip -\leftskip
1008 \UseHookWithArguments{contentsline/text/before}{4}

```

```

1009         {\@nameuse{toclevel@#2}}{##1}{##2}{\@contentsline@destination}%
1010         {\@nameuse{cft#2font} ##1}\nobreak
1011         \UseHookWithArguments{contentsline/text/after}{4}
1012         {\@nameuse{toclevel@#2}}{##1}{##2}{\@contentsline@destination}%
1013         \@nameuse{cft#2fillnum}{##2}}%
1014     \fi
1015 } % end of \l@#2
1016

```

Now define all the layout commands used by \l@X. The default values of these correspond to those for section entries in non-chaptered documents.

```

1017 \expandafter\newlength\csname cftbefore#2skip\endcsname
1018 \setlength{\@nameuse{cftbefore#2skip}}{\z@ \@plus .2\p@}

\cftXindent 1019 \expandafter\newlength\csname cft#2indent\endcsname
\cftXnumwidth 1020 \expandafter\newlength\csname cft#2numwidth\endcsname

```

Set the default values for the indent and numwidth depending on the entry's level. A level of 1 corresponds to a figure entry.

```

1021 \ifcase #4\relax % 0
1022     \setlength{\@nameuse{cft#2indent}}{0em}
1023     \setlength{\@nameuse{cft#2numwidth}}{1.5em}
1024 \or % 1
1025     \setlength{\@nameuse{cft#2indent}}{1.5em}
1026     \setlength{\@nameuse{cft#2numwidth}}{2.3em}
1027 \or % 2
1028     \setlength{\@nameuse{cft#2indent}}{3.8em}
1029     \setlength{\@nameuse{cft#2numwidth}}{3.2em}
1030 \or % 3
1031     \setlength{\@nameuse{cft#2indent}}{7.0em}
1032     \setlength{\@nameuse{cft#2numwidth}}{4.1em}
1033 \else % anything else
1034     \setlength{\@nameuse{cft#2indent}}{10.0em}
1035     \setlength{\@nameuse{cft#2numwidth}}{5.0em}
1036 \fi

```

And the remaining commands.

```

\cftXfont 1037 \@namedef{cft#2font}{\normalfont}
\cftXpresnum 1038 \@namedef{cft#2presnum}{\}
\cftXaftersnum 1039 \@namedef{cft#2aftersnum}{\}
\cftXaftersnumb 1040 \@namedef{cft#2aftersnumb}{\}
\cftXdotsep 1041 \@namedef{cft#2dotsep}{\cftdotsep}
\cftXleader 1042 \@namedef{cft#2leader}{\normalfont\cftdotfill{\@nameuse{cft#2dotsep}}}
\cftXpagefont 1043 \@namedef{cft#2pagefont}{\normalfont}
\cftXafterpnum 1044 \@namedef{cft#2afterpnum}{\}
\cftXpnumfinal 1045 \@namedef{cft#2pnumfinal}{\cftpnumfinaldefault}

```

The hyperref package needs a command \toclevel@X, holding the *<level>* value.

```

\toclevel@X 1046 \ExpandArgs{ce}\def{toclevel@#2}{\interval{#4+1}}

```

Typeset the leader and page number.

```

\cftXfillnum 1047 \@namedef{cft#2fillnum}##1{%
1048     {\UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%

```

```

1049     \@nameuse{cft#2leader}%
1050     \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
1051   }\nobreak
1052   \UseHookWithArguments{contentsline/page/before}{4}
1053     {\@nameuse{toclevel@#2}}{\@##1}{\@contentsline@destination}
1054   \makebox[\@pnumwidth][\cftpnumalign]{\@nameuse{cft#2pagefont}\@##1}%
1055   \UseHookWithArguments{contentsline/page/after}{4}
1056     {\@nameuse{toclevel@#2}}{\@##1}{\@contentsline@destination}%
1057   \@nameuse{cft#2afterpnum}\@nameuse{cft#2pnumfinal}\par
1058 }

```

This ends the definition of `\newlistentry`.

```

1059 }

```

(End of definition for `\newlistentry` and others. These functions are documented on page 14.)

`\newlistof` `\newlistof[⟨within⟩]{⟨entry⟩}{⟨ext⟩}{⟨listofname⟩}` creates the commands for a new List of.

```

1060 \newcommand{\newlistof}[4][\@empty]{%

```

Call `\newlistentry` to set up the first level entry.

```

1061   \ifx \@empty#1\relax
1062     \newlistentry{#2}{#3}{0}
1063   \else
1064     \newlistentry[#1]{#2}{#3}{0}
1065   \fi

```

(End of definition for `\newlistof`. This function is documented on page 13.)

`\ext@Z` The file extension and listing depth.

```

1066 \Zdepth
1067   \@namedef{ext@#2}{#3}
1068   \newcounter{#3depth}
1069   \setcounter{#3depth}{1}

```

(End of definition for `\ext@Z` and `\Zdepth`.)

`\cftZtitlename`

```

1069   \@namedef{cft#3titlename}{#4}

```

(End of definition for `\cftZtitlename`. This function is documented on page 8.)

`\cftmarkZ` The heading marks for the listing.

```

1070   \if@cftkoma
1071     \@namedef{cftmark#3}{%
1072       \@mkboth{\UseName{cft#3titlename}}{\UseName{cft#3titlename}}}
1073   \else
1074     \@namedef{cftmark#3}{%
1075       \@mkboth{\MakeUppercase{\UseName{cft#3titlename}}}
1076               {\MakeUppercase{\UseName{cft#3titlename}}}}
1077   \fi

```

(End of definition for `\cftmarkZ`. This function is documented on page 7.)

`\listofX` Typeset the listing title and entries.

```

1078   \if@cftnctoc

```

For the titles option, basically copy the code from the standard `\tableofcontents` command.

```

1079 \namedef{listof#2}{%
1080   \cfttocstart
1081   \if@cfthaschapter
1082     \chapter*{\UseName{cft#3titlename}}
1083   \else
1084     \section*{\UseName{cft#3titlename}}
1085   \fi
1086   \nameuse{cftmark#3}
1087   \starttoc{#3}%
1088   \cfttocfinish}
1089 \else

```

Otherwise use the fully parameterised definition.

```

1090 \namedef{listof#2}{%
1091   \cfttocstart
1092   \par
1093   \begingroup
1094     \parindent\z@ \parskip\cftparskip
1095     \nameuse{@cftmake#3title}
1096     \starttoc{#3}%
1097   \endgroup
1098   \cfttocfinish}
1099 \fi
1100

```

(End of definition for `\listofX`. This function is documented on page 14.)

`\@cftmakeZtitle` Typeset the title.

```

1101 \namedef{@cftmake#3title}{\cfttitlecommand{#3}}%

```

(End of definition for `\@cftmakeZtitle`.)

`\cftbeforeZtitleskip` The skips before and after the title heading, and the title font. The default values depend on whether or not the document class has chapters.

`\cftafterZtitleskip`

`\cftZtitlefont`

```

1102 \expandafter\newlength\csname cftbefore#3titleskip\endcsname
1103 \expandafter\newlength\csname cftafter#3titleskip\endcsname
1104 \if@cfthaschapter
1105   \setlength{\@nameuse{cftbefore#3titleskip}}{50pt}
1106   \setlength{\@nameuse{cftafter#3titleskip}}{40pt}
1107   \if@cftkoma
1108     \namedef{cft#3titlefont}{\size@chapter\sectfont}
1109   \else
1110     \namedef{cft#3titlefont}{\normalfont\Huge\bfseries}
1111   \fi
1112 \else
1113   \setlength{\@nameuse{cftbefore#3titleskip}}{3.5ex \@plus 1ex \@minus .2ex}
1114   \setlength{\@nameuse{cftafter#3titleskip}}{2.3ex \@plus .2ex}
1115   \if@cftkoma
1116     \namedef{cft#3titlefont}{\size@section\sectfont}
1117   \else
1118     \namedef{cft#3titlefont}{\normalfont\Huge\bfseries}
1119   \fi
1120 \fi

```

(End of definition for `\cftbeforeZtitleskip`, `\cftafterZtitleskip`, and `\cftZtitlefont`. These functions are documented on page 7.)

`\cftafterZtitle` Something to go after the title.

```
1121 \namedef{cftafter#3title}{}

```

(End of definition for `\cftafterZtitle`. This function is documented on page 8.)

`\cftZprehook` Hooks before and after the list of entries.

```
1122 \namedef{cft#3prehook}{}
1123 \namedef{cft#3posthook}{}

```

This is the end of the definition of `\newlistof`.

```
1124 }

```

(End of definition for `\cftZprehook` and `\cftZposthook`. These functions are documented on page 17.)

`\cftsetindents` `\cftsetindents{<entry>}{<indent>}{<numwidth>}` sets the *indent* and *numwidth* for entry `<entry>`. The macro has to map between the external entry name and the internal shorthand.

```
1125 \newcommand{\cftsetindents}[3]{%
1126   \def\@cftemp{#1}
1127   \ifx\@cftemp\cftchapname
1128     \cftsetindents{chap}{#2}{#3}
1129   \else
1130     \ifx\@cftemp\cftsecname \cftsetindents{sec}{#2}{#3}
1131     \else
1132       \ifx\@cftemp\cftsubsecname \cftsetindents{subsec}{#2}{#3}
1133       \else
1134         \ifx\@cftemp\cftsubsubsecname \cftsetindents{subsubsec}{#2}{#3}
1135         \else
1136           \ifx\@cftemp\cftparaname \cftsetindents{para}{#2}{#3}
1137           \else
1138             \ifx\@cftemp\cftsubparaname \cftsetindents{subpara}{#2}{#3}
1139             \else
1140               \ifx\@cftemp\cftfigname \cftsetindents{fig}{#2}{#3}
1141               \else
1142                 \ifx\@cftemp\cftsubfigname \cftsetindents{subfig}{#2}{#3}
1143                 \else
1144                   \ifx\@cftemp\cfttabname \cftsetindents{tab}{#2}{#3}
1145                   \else
1146                     \ifx\@cftemp\cftsubtabname \cftsetindents{subtab}{#2}{#3}
1147                     \else
1148                       \cftsetindents{#1}{#2}{#3}
1149                     \fi
1150                   \fi
1151                 \fi
1152               \fi
1153             \fi
1154           \fi
1155         \fi
1156       \fi
1157     \fi
1158   \fi
1159 }
1160

```

(End of definition for `\cftsetindents`. This function is documented on page 11.)

`\@cftsetindents` `\@cftsetindents{⟨X⟩}{⟨indent⟩}{⟨numwidth⟩}` is the internal version of `\cftsetindents`, where in this case `⟨X⟩` is the internal (shorthand) name of the entry.

```
1161 \newcommand{\@cftsetindents}[3]{%
1162   \setlength{\@nameuse{cft#1indent}}{#2}
1163   \setlength{\@nameuse{cft#1numwidth}}{#3}
1164 }
1165
```

(End of definition for `\@cftsetindents`.)

3.3 Switching page numbering

`\@cftpnumoff` `\@cftpnumoff{⟨shorthand⟩}` is the workhorse for switching page numbering off. The `⟨shorthand⟩` argument is the shorthand name of the entry (e.g. `subsec` for `subsection`). The macro redefines the `\cftXnumfill` command so that there is no leader and the page number is ignored.

```
1166 \newcommand{\@cftpnumoff}[1]{%
1167   \@namedef{cft#1fillnum}##1{\UseTaggingSocket{struct/end}%
1168     \cftpnumfillskip\@nameuse{cft#1afterpnum}\@nameuse{cft#1pnumfinal}\par}}
1169
```

(End of definition for `\@cftpnumoff`.)

`\cftchapname` Unfortunately an early design decision was the use shorthands like `sec` for `section`. For the page switching I need to be able to correlate the shorthands and longhands.

`\cftsecname`

`\cftsubsecname` 1170 `\newcommand*{\cftchapname}{chapter}`

`\cftsubsubsecname` 1171 `\newcommand*{\cftsecname}{section}`

`\cftparaname` 1172 `\newcommand*{\cftsubsecname}{subsection}`

`\cftsubparaname` 1173 `\newcommand*{\cftsubsubsecname}{subsubsection}`

`\cftfigname` 1174 `\newcommand*{\cftparaname}{paragraph}`

`\cftsubfigname` 1175 `\newcommand*{\cftsubparaname}{subparagraph}`

`\cfttabname` 1176 `\newcommand*{\cftfigname}{figure}`

`\cftsubtabname` 1177 `\newcommand*{\cftsubfigname}{subfigure}`

1178 `\newcommand*{\cfttabname}{table}`

1179 `\newcommand*{\cftsubtabname}{subtable}`

1180

(End of definition for `\cftchapname` and others.)

`\cftpagenumbersoff` The user level command for switching off page numbers is `\cftpagenumbersoff{⟨entry⟩}` where `⟨entry⟩` is the longhand name of the entry. The principal task of this macro is to determine the corresponding shorthand name of the `⟨entry⟩` and then call `\@cftpnumoff` to do the work. For `part` and user-defined entries the long- and short-hand entry names are identical.

```
1181 \DeclareRobustCommand{\cftpagenumbersoff}[1]{%
1182   \def\@cftemp{#1}
1183   \ifx\@cftemp\cftchapname
1184     \@cftpnumoff{chap}
1185   \else
1186     \ifx\@cftemp\cftsecname \@cftpnumoff{sec}
1187     \else
1188       \ifx\@cftemp\cftsubsecname \@cftpnumoff{subsec}

```

```

1189 \else
1190 \ifx\@cfttemp\cftsubsubsecname \@cftpnumoff{subsubsec}
1191 \else
1192 \ifx\@cfttemp\cftparaname \@cftpnumoff{para}
1193 \else
1194 \ifx\@cfttemp\cftsubparaname \@cftpnumoff{subpara}
1195 \else
1196 \ifx\@cfttemp\cftfigname \@cftpnumoff{fig}
1197 \else
1198 \ifx\@cfttemp\cftsubfigname \@cftpnumoff{subfig}
1199 \else
1200 \ifx\@cfttemp\cfttabname \@cftpnumoff{tab}
1201 \else
1202 \ifx\@cfttemp\cftsubtabname \@cftpnumoff{subtab}
1203 \else
1204 \@cftpnumoff{#1}
1205 \fi
1206 \fi
1207 \fi
1208 \fi
1209 \fi
1210 \fi
1211 \fi
1212 \fi
1213 \fi
1214 \fi
1215 }
1216

```

(End of definition for \cftpagenumbersoff. This function is documented on page 12.)

\cftpagenumberon \cftpagenumberon{<entry>} is the user level command for reversing the corresponding \cftpagenumbersoff.

```

1217 \DeclareRobustCommand{\cftpagenumberon}[1]{%
1218 \def\@cfttemp{#1}
1219 \ifx\@cfttemp\cftchapname
1220 \cftpnumon{chap}
1221 \else
1222 \ifx\@cfttemp\cftsecname \@cftpnumon{sec}
1223 \else
1224 \ifx\@cfttemp\cftsubsecname \@cftpnumon{subsec}
1225 \else
1226 \ifx\@cfttemp\cftsubsubsecname \@cftpnumon{subsubsec}
1227 \else
1228 \ifx\@cfttemp\cftparaname \@cftpnumon{para}
1229 \else
1230 \ifx\@cfttemp\cftsubparaname \@cftpnumon{subpara}
1231 \else
1232 \ifx\@cfttemp\cftfigname \@cftpnumon{fig}
1233 \else
1234 \ifx\@cfttemp\cftsubfigname \@cftpnumon{subfig}
1235 \else
1236 \ifx\@cfttemp\cfttabname \@cftpnumon{tab}
1237 \else

```

```

1238             \ifx\@cfttemp\cftsubtabname \@cftpnumon{subtab}
1239             \else
1240               \@cftpnumon{#1}
1241             \fi
1242           \fi
1243         \fi
1244       \fi
1245     \fi
1246   \fi
1247 \fi
1248 \fi
1249 \fi
1250 \fi
1251 }
1252

```

(End of definition for \cftpagenumerson. This function is documented on page 12.)

\cftpnumon **\@cftpnumon{<shorthand>}** is the workhorse for switching page numbering off. The **<shorthand>** argument is the shorthand name of the entry (e.g. **subsec** for **subsection**). The macro defines the **\cftXnumfill** command to correspond to the default definition.

```

1253 \newcommand{\@cftpnumon}[1]{%
1254   \@namedef{cft#1fillnum}##1{%
1255     {\@nameuse{cft#1leader}}\nobreak
1256     \makebox[\@pnumwidth][\cftpnumalign]{\@nameuse{cft#1pagefont}##1}%
1257     \@nameuse{cft#1afterpnum}\@nameuse{cft#1pnumfinal}\par
1258   }%
1259 }

```

(End of definition for \@cftpnumon.)

3.4 Experimental utilities

The code in this section is experimental but in the sense that the capabilities might be modified in the future rather than that the code does not work.

\cftchapterprecis This is experimental. **\cftchapterprecis{<text>}** typesets **<text>** at the point where it is called, and also adds **<text>** to the **.toc** file. It expects to be called immediately after a **\chapter** command.

```

1260 \newcommand{\cftchapterprecis}[1]{%
1261   \cftchapterprecishere{#1}
1262   \cftchapterprecistoc{#1}}

```

(End of definition for \cftchapterprecis. This function is documented on page 16.)

\cftchapterprecishere **\cftchapterprecishere{<text>}** typesets **<text>**. It expects to be called immediately after a **\chapter** command. First add some negative vertical space to move it closer to the chapter heading.

```

1263 \newcommand{\cftchapterprecishere}[1]{%
1264   \vspace*{-2\baselineskip}

```

Typeset its argument using italic font in a quote environment.

```

1265   \begin{quote}\textit{#1}\end{quote}}

```

(End of definition for \cftchapterprecishere. This function is documented on page 16.)

`\cftchapterprecistoc` `\cftchapterprecistoc{<text>}` adds `<text>` to the `.toc` file. The `<text>` will be typeset within the same margins as the the title text of a `\chapter` heading, using an italic font.

```
1266 \newcommand{\cftchapterprecistoc}[1]{\addtocontents{toc}{%
```

Start a group to localize changes to the paragraphing. Set the left margin to the chapter indent plus the chapter number width.

```
1267 {\UseTaggingSocket{struct-mc}{tag=\protect\UseStructureName{toc/item}}{}%
1268 \leftskip \cftchapindent\relax
1269 \advance\leftskip \cftchapnumwidth\relax
```

Set the right hand margin to `\@tocrmarg`.

```
1270 \rightskip \@tocrmarg\relax
```

Typeset `<text>` using an italic font, then ensure that the paragraph is finished (to use the local skips). Finally close the group and we are done.

```
1271 \textit{#1}\protect\par}}}}
1272
```

(End of definition for `\cftchapterprecistoc`. This function is documented on page 16.)

`\cftlocalchange` `\cftmakelocalchange{<file>}{<pnumwidth>}{<tocrmarg>}` makes an entry into `<file>` to change the `\@pnumwidth` and the `\@tocrmarg` values.

```
1273 \newcommand{\cftlocalchange}[3]{%
1274 \addtocontents{#1}{\protect\cftsetpnumwidth{#2} \protect\cftsetrmarg{#3}}}
```

(End of definition for `\cftlocalchange`. This function is documented on page 16.)

`\cftaddtitleline` `\cftaddtitleline{<file>}{<kind>}{<title>}{<page>}` adds a `\contentsline` entry to `<file>` with the given information.

```
1275 \newcommand{\cftaddtitleline}[4]{\addtocontents{#1}{%
1276 \protect\contentsline{#2}{#3}{#4}{\@currentHref}\protected@file@percent}}
```

(End of definition for `\cftaddtitleline`. This function is documented on page 16.)

`\cftaddnumtitleline` `\cftaddnumtitleline{<file>}{<kind>}{<num>}{<title>}{<page>}` adds a `\contentsline` entry to `<file>` with the given information.

```
1277 \newcommand{\cftaddnumtitleline}[5]{\addtocontents{#1}{%
1278 \protect\contentsline{#2}{\protect\numberline{#3}{#4}{#5}{\@currentHref}}%
1279 \protected@file@percent}}
```

(End of definition for `\cftaddnumtitleline`. This function is documented on page 16.)

And, if dear old `hyperref` has been used, we have to fix up these two macros. . . Argument number is identical now.

`\@starttoc`

Okay, here's a roughly-comprehensive list where `\@starttoc` is redefined in T_EX Live 2014.

- `amsart`, `amsbook`, `amsdtx`, `amsproc`
- `asect`
- `latex.ltx` (of course)

- newfloat
- flowfram
- gmampulex (?), gmtypos
- hyperref, memhfixc
- ijmart
- scrartcl, scrbook, screprt
- scrwtfle
- tocbasic
- tocstyle
- memoir
- multitoc
- nccsect
- notoccite
- artikel3, boek3, rapport3,
- rerunfilecheck
- parskip
- pdfwin
- revtex4
- devanagari

This makes things difficult if we want to redefine `\@starttoc` here. Many of the packages/classes above will not be used in conjunction with `tocloft`; on the other hand, we don't want to trample too much on others' code.

So for our own work here, let's be extra conservative, at least for now, and only hook into `\starttoc` if it's the standard L^AT_EX version.

```

1280 \def\@starttoc@latex@orig#1{%
1281   \begingroup
1282     \makeatletter
1283     \@input{\jobname.#1}%
1284     \if@filesw
1285       \expandafter\newwrite\csname tf@#1\endcsname
1286       \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
1287     \fi
1288     \@nobreakfalse
1289   \endgroup}
1290 \newif\if@cft@patch@starttoc
1291 \IfDocumentMetadataTF
1292 {
1293   \@cft@patch@starttoctrue

```

```

1294 }
1295 {
1296   \ifx\@starttoc\@starttoc@latex@orig
1297   \@cft@patch@starttoctrue
1298   \fi
1299 }
1300 \if@cft@patch@starttoc
1301   \patchcmd\@starttoc
1302     {\begingroup}
1303     {\csname cft#1prehook\endcsname\begingroup}
1304     {}{\@cft@fail}
1305   \patchcmd\@starttoc
1306     {\endgroup}
1307     {\endgroup\csname cft#1posthook\endcsname}
1308     {}{\@cft@fail}
1309   \newcommand\cfttocprehook{}
1310   \newcommand\cftlofprehook{}
1311   \newcommand\cftlotprehook{}
1312   \newcommand\cfttocposthook{}
1313   \newcommand\cftlofposthook{}
1314   \newcommand\cftlotposthook{}
1315 \else
1316   \PackageWarning{tocloft}{\string\@starttoc\space has already been redefined;
1317   \string\cfttocprehook\space et al. not installed.}
1318 \fi

```

(End of definition for \@starttoc.)

The end of this package.

```
1319 </usc>
```

References

- [Coc95] Steven Douglas Cochran. *The subfigure package*. March 1995. (Available from CTAN as file `subfigure.dtx`)
- [Dru99] Jean-Pierre Drucbert. *The minitoc package*. August 1999. (Available from CTAN in subdirectory `/minitoc`)
- [GMS94] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley Publishing Company, 1994.
- [Lin97] Ulf A. Lindgren. *FncyChap V1.11*. April 1997. (Available from CTAN in subdirectory `/fncychap`)
- [Lin95] Anselm Lingnau. *An Improved Environment for Floats*. March 1995. (Available from CTAN in subdirectory `/float`)
- [Wil96a] Peter Williams. *Algorithms*. April 1996. (Available from CTAN in subdirectory `/algorithm`)
- [Wil96b] Peter R. Wilson. *LaTeX for standards: The LaTeX package files user manual*. NIST Report NISTIR, June 1996.

[Wil00] Peter R. Wilson. *The tocbibind package*. March 2000. (Available from CTAN as file tocbibind.dtx)

[Wil01] Peter R. Wilson. *The ccaption package*. March 2001. (Available from CTAN as file ccaption.dtx)

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.

Symbols	
\@cftasnum	17
\@cftasnumb	17
\@cftbsnum	17
\@dotsep	4
\@dottedtocline	5
\@pnumwidth	4
\@tocrmarg	4
A	
\addcontentsline	3
\addcontentsline	84, 87, 660, 663, 761, 764
\addtocontents	5
\addtocontents	1266, 1274, 1275, 1277
\AtBeginDocument	18, 30, 161, 666, 768, 971, 973
B	
\baselineskip	1264
C	
\cftaddnumtitleline	16
\cftaddnumtitleline	1277
\cftaddtitleline	16
\cftaddtitleline	1275
\cftafterloftitle	693
\cftafterloftitleskip	684
\cftafterlottitle	795
\cftafterlottitleskip	786
\cftaftertoctitle	196
\cftaftertoctitleskip	187
\cftafterZtitle	8
\cftafterZtitle	3, 1121
\cftafterZtitleskip	7
\cftafterZtitleskip	1102
\cftbeforechapskip	327, 347
\cftbeforefigskip	704, 724
\cftbeforeloftitleskip	684
\cftbeforelottitleskip	786
\cftbeforeparaskip	550, 569
\cftbeforepartskip	254, 287
\cftbeforesecskip	383, 386, 406
\cftbeforesubfigskip	858, 881
\cftbeforesubparaskip	604, 623
\cftbeforesubsecskip	446, 463
\cftbeforesubsubsecskip	498, 515
\cftbeforesubtabskip	916, 938
\cftbeforetabskip	806, 826
\cftbeforetoctitleskip	187
\cftbeforeXskip	9
\cftbeforeXskip	1017
\cftbeforeZtitleskip	7
\cftbeforeZtitleskip	1102
\cftchapafterpnum	347
\cftchapaftersnum	336, 347
\cftchapaftersnumb	337, 347
\cftchapdotsep	347
\cftchapfillnum	344, 347
\cftchapfont	341, 347
\cftchapindent	328, 331, 347, 1268
\cftchapleader	347
\cftchapname	1127, 1170, 1183, 1219
\cftchapnumwidth	334, 347, 1269
\cftchappagefont	347
\cftchappnumfinal	347
\cftchappresnum	335, 347
\cftchapter	8
\cftchapter	99, 103
\cftchapterprecis	16
\cftchapterprecis	1260
\cftchapterprecishere	16
\cftchapterprecishere	1261, 1263
\cftchapterprecistoc	16
\cftchapterprecistoc	1262, 1266
\cftdot	8
\cftdot	207
\cftdotfill	207, 297, 358, 417, 425, 478, 530, 584, 638, 734, 836, 891, 948, 1042
\cftdotsep	9
\cftdotsep	217, 418, 479, 531, 585, 639, 735, 837, 892, 949, 1041
\cftfigafterpnum	724

\cftfigaftersnum	713, 724	\cftpartfillnum	273, 287
\cftfigaftersnumb	714, 724	\cftpartfont	270, 287
\cftfigdotsep	724	\cftpartindent	256, 259, 287
\cftfigfillnum	721, 724	\cftpartinitnumwidth	10
\cftfigfont	718, 724	\cftpartinitnumwidth	263, 293
\cftfigindent	705, 708, 724	\cftpartleader	287
\cftfigleader	724	\cftpartnumwidth	262, 287
\cftfigname	1140, 1170 , 1196, 1232	\cftpartpagefont	287
\cftfignumwidth	711, 724	\cftpartpnumfinal	287
\cftfigpagefont	724	\cftpartpresnum	264, 270, 287
\cftfigpnumfinal	724	\cftpnumalign	9
\cftfigpresnum	712, 724	\cftpnumalign	9 , 206, 312, 370, 439, 491, 543, 597, 651, 747, 849, 905, 962, 1054, 1256
\cftlocalchange	16	\cftpnumfinaldefault	11
\cftlocalchange	1273	\cftpnumfinaldefault	94 , 301, 362, 430, 482, 534, 588, 642, 738, 840, 895, 952, 1045
\cftlofpsthook	1313	\cftsecafterpnum	406
\cftlofpsthook	1310	\cftsecaftersnum	396, 406
\cftlofttitlefont	693	\cftsecaftersnumb	397, 406
\cftlofttitlename	55, 62, 96, 660, 663	\cftsecdotsep	406
\cftlotposthook	1314	\cftsecfillnum	404, 406
\cftlotprehook	1311	\cftsecfont	401, 406
\cftlotttitlefont	795	\cftsecindent	388, 391, 406
\cftlotttitlename	57, 64, 97, 761, 764	\cftsecleader	406
\cftmarklof	52	\cftsecname	1130, 1170 , 1186, 1222
\cftmarklot	52	\cftsecnumwidth	394, 406
\cftmarktoc	52	\cftsecpagefont	406
\cftmarkZ	7	\cftsecpnumfinal	406
\cftmarkZ	1070	\cftsecpresnum	395, 406
\cftnodots	9	\cftsection	8
\cftnodots	210, 217 , 298, 359, 426	\cftsection	101, 103
\cftpagenumbersoff	12	\cftsetindents	11
\cftpagenumbersoff	1181	\cftsetindents	1125
\cftpagenumberson	12	\cftsetpnumwidth	9
\cftpagenumberson	1217	\cftsetpnumwidth	204, 1274
\cftparaafterpnum	569	\cftsetrmarg	9
\cftparaaftersnum	559, 569	\cftsetrmarg	204, 1274
\cftparaaftersnumb	560, 569	\cftsubfigafterpnum	881
\cftparadotsep	569	\cftsubfigaftersnum	867, 881
\cftparafillnum	567, 569	\cftsubfigaftersnumb	868, 881
\cftparafont	564, 569	\cftsubfigdotsep	881
\cftparaindent	551, 554, 569	\cftsubfigfillnum	875, 881
\cftparaleader	569	\cftsubfigfont	872, 881
\cftparaname	1136, 1170 , 1192, 1228	\cftsubfigindent	859, 862, 881
\cftparanumwidth	557, 569	\cftsubfigleader	881
\cftparapagefont	569	\cftsubfigname	1142, 1170 , 1198, 1234
\cftparapnumfinal	569	\cftsubfignumwidth	865, 881
\cftparapresnum	558, 569	\cftsubfigpagefont	881
\cftparfillskip	12	\cftsubfigpnumfinal	881
\cftparfillskip	219, 1168	\cftsubfigpresnum	866, 881
\cftparskip	9	\cftsubparaafterpnum	623
\cftparskip	91 , 163, 167, 176, 672, 774, 1094	\cftsubparaaftersnum	613, 623
\cftpartafterpnum	287		
\cftpartaftersnum	265, 287		
\cftpartaftersnumb	266, 287		
\cftpartdotsep	287		

\cftsubparaaftersnumb	614, 623	\cftsubtagpnumfinal	965
\cftsubparadotsep	623	\cfttabafterpnum	826
\cftsubparafillnum	621, 623	\cfttabaftersnum	815, 826
\cftsubparafont	618, 623	\cfttabaftersnumb	816, 826
\cftsubparaindent	605, 608, 623	\cfttabdotsep	826
\cftsubparaleader	623	\cfttabfillnum	823, 826
\cftsubparaname	1138, 1170 , 1194, 1230	\cfttabfont	820, 826
\cftsubparanumwidth	611, 623	\cfttabindent	807, 810, 826
\cftsubparapagefont	623	\cfttableader	826
\cftsubparapnumfinal	623	\cfttabname	1144, 1170 , 1200, 1236
\cftsubparapresnum	612, 623	\cfttabnumwidth	813, 826
\cftsubsecafterpnum	463, 494	\cfttabpagefont	826
\cftsubsecaftersnum	455, 463	\cfttabpnumfinal	826
\cftsubsecaftersnumb	456, 463	\cfttabpresnum	814, 826
\cftsubsecdotsep	463	\cfttitlecommand	8
\cftsubsecfillnum	461, 483	\cfttitlecommand	98, 186, 683, 785, 1101
\cftsubsecfont	459, 463	\cfttocposthook	3, 1312
\cftsubsecindent	447, 450, 463	\cfttocprehook	3, 1309, 1317
\cftsubsecleader	463, 486	\cfttoctitlefont	196
\cftsubsecname	1132, 1170 , 1188, 1224	\cfttoctitlename	53, 60, 84, 87, 95
\cftsubsecnumwidth	453, 463	\cftXafterpnum	10
\cftsubsecpagefont	463, 491	\cftXafterpnum	1037
\cftsubsecpnumfinal	463, 494	\cftXaftersnum	10
\cftsubsecpresnum	454, 463	\cftXaftersnum	1037
\cftsubsubsecafterpnum	515	\cftXaftersnumb	10
\cftsubsubsecaftersnum	507, 515	\cftXaftersnumb	1037
\cftsubsubsecaftersnumb	508, 515	\cftXdotsep	10
\cftsubsubsecdotsep	515	\cftXdotsep	1037
\cftsubsubsecfillnum	513, 515	\cftXfillnum	10
\cftsubsubsecfont	511, 515	\cftXfillnum	1047
\cftsubsubsecindent	499, 502, 515	\cftXfont	10
\cftsubsubsecleader	515	\cftXfont	1037
\cftsubsubsecname	1134, 1170 , 1190, 1226	\cftXindent	9
\cftsubsubsecnumwidth	505, 515	\cftXindent	1019
\cftsubsubsecpagefont	515	\cftXleader	10
\cftsubsubsecpnumfinal	515	\cftXleader	1037
\cftsubsubsecpresnum	506, 515	\cftXnumwidth	9
\cftsubtabafterpnum	938	\cftXnumwidth	1019
\cftsubtabaftersnum	925, 938	\cftXpagefont	10
\cftsubtabaftersnumb	926, 938	\cftXpagefont	1037
\cftsubtabdotsep	891, 938	\cftXpnumfinal	10
\cftsubtabfillnum	933, 938	\cftXpnumfinal	1037
\cftsubtabfont	930, 938	\cftXpresnum	10
\cftsubtabindent	917, 920, 938	\cftXpresnum	1037
\cftsubtableader	938	\cftZposthook	17
\cftsubtabname	1146, 1170 , 1202, 1238	\cftZposthook	1122
\cftsubtabnumwidth	923, 938	\cftZprehook	17
\cftsubtabpagefont	938	\cftZprehook	17, 1122
\cftsubtabpnumfinal	938	\cftZtitlefont	8
\cftsubtabpresnum	924, 938	\cftZtitlefont	1102
\cftsubtagafterpnum	965	\cftZtitlename	8
\cftsubtagfillnum	954	\cftZtitlename	95, 1069
\cftsubtagleader	957	\chapter	107, 1082
\cftsubtagpagefont	962	\contentsline	3

<code>\contentsline</code>	1276, 1278	<code>\MakeUppercase</code>	53, 55, 57, 1075, 1076
<code>\contentsname</code>	95	<code>\MessageBreak</code>	35, 36, 37
<code>\csname</code>	5, 987, 988, 1004, 1005, 1006, 1017, 1019, 1020, 1102, 1103, 1285, 1286, 1303, 1307	<code>\mkern</code>	214
D			
<code>\DeclareDocumentCommand</code>	105, 119, 134, 147	<code>\newif</code>	11, 12, 17, 29, 41, 44, 232, 233, 1290
<code>\DeclareRobustCommand</code>	1181, 1217	<code>\newlistentry</code>	14
<code>\dimeval</code>	112	<code>\newlistentry</code>	978, 1062, 1064
<code>\DocumentMetadata</code>	2, 3, 8, 21	<code>\newlistof</code>	13
E			
<code>\edef</code>	987	<code>\newlistof</code>	1060
<code>\endcsname</code>	5, 987, 988, 1004, 1005, 1006, 1017, 1019, 1020, 1102, 1103, 1285, 1286, 1303, 1307	<code>\newwrite</code>	1285
<code>\endinput</code>	24	<code>\noexpand</code>	988
<code>\ExpandArgs</code>	1046	<code>\normalfont</code>	197, 200, 416, 417, 419, 474, 478, 480, 526, 530, 532, 580, 584, 586, 634, 638, 640, 694, 697, 730, 734, 736, 796, 799, 832, 836, 838, 887, 891, 893, 944, 948, 950, 1037, 1042, 1043, 1110, 1118
H			
<code>\Huge</code>	197, 694, 796, 1110, 1118	<code>\numberline</code>	4
I			
<code>\IfDocumentMetadataTF</code>	103, 1291	<code>\numberline</code>	220
<code>\immediate</code>	1286	O	
<code>\interlinepenalty</code>	139, 152, 260, 332, 392, 451, 503, 555, 609, 709, 811, 863, 921, 1001	<code>\onecolumn</code>	69
<code>\inteval</code>	1046	<code>\openout</code>	1286
L			
<code>\l@X</code>	14	P	
<code>\Large</code>	200, 697, 799	<code>\PackageError</code>	992
<code>\large</code>	292, 297, 299, 319, 320	<code>\PackageInfo</code>	25, 27
<code>\leaders</code>	214	<code>\PackageWarning</code>	22, 34, 983, 1316
<code>\leftskip</code>	256, 267, 328, 338, 388, 398, 447, 457, 499, 509, 551, 561, 605, 615, 705, 715, 807, 817, 859, 869, 917, 927, 997, 1007, 1268, 1269	<code>\par</code>	3
<code>\listfigurename</code>	96	<code>\parindent</code>	176, 259, 331, 391, 450, 502, 554, 608, 672, 708, 774, 810, 862, 920, 1000, 1094
<code>\listoffigures</code>	7	<code>\parskip</code>	167, 176, 672, 774, 1094
<code>\listoffigures</code>	666	<code>\patchcmd</code>	1301, 1305
<code>\listoftables</code>	7	<code>\phantomsection</code>	78, 83, 86, 659, 662, 760, 763
<code>\listoftables</code>	768	<code>\providecommand</code>	66, 74, 78, 207, 208, 217, 896, 953
<code>\listofX</code>	14	R	
<code>\listofX</code>	1078	<code>\RequirePackage</code>	2
<code>\listtablename</code>	97	<code>\ResumeTagging</code>	308, 366, 435, 487, 539, 593, 647, 743, 845, 901, 958, 1050
<code>lofdepth</code>	752	S	
<code>lotdepth</code>	752	<code>\sectfont</code>	198, 201, 319, 320, 376, 695, 698, 797, 800, 1108, 1116
M			
<code>\makeatletter</code>	1282	<code>\section</code>	121, 1084
<code>\makebox</code> .	9, 24, 312, 370, 439, 491, 543, 597, 651, 747, 849, 905, 962, 1054, 1256	<code>\starttoc</code>	54
N			
<code>\newif</code>	11, 12, 17, 29, 41, 44, 232, 233, 1290	<code>\string</code>	1316, 1317
<code>\newlistentry</code>	14	<code>\SuspendTagging</code> ...	306, 364, 433, 485, 537, 591, 645, 741, 843, 899, 956, 1048
<code>\newlistentry</code>	978, 1062, 1064	T	
<code>\newlistof</code>	13	<code>\tableofcontents</code>	7
<code>\newlistof</code>	1060	<code>\tableofcontents</code>	161
<code>\newwrite</code>	1285		
<code>\noexpand</code>	988		
<code>\normalfont</code>	197, 200, 416, 417, 419, 474, 478, 480, 526, 530, 532, 580, 584, 586, 634, 638, 640, 694, 697, 730, 734, 736, 796, 799, 832, 836, 838, 887, 891, 893, 944, 948, 950, 1037, 1042, 1043, 1110, 1118		

TeX and L^AT_EX 2_ε commands:

<code>\@afterheading</code>	114, 127, 145, 158	<code>\@cftsetsubtab</code>	937, 970, 974
<code>\@afterindenttrue</code>	259, 331, 391, 450, 502, 554, 608, 708, 810, 862, 920, 1000	<code>\@cftsubfigoptfalse</code>	44
<code>\@cft@fail</code>	1304, 1308	<code>\@cftsubfigopttrue</code>	45
<code>\@cft@patch@starttoctrue</code>	1293, 1297	<code>\@cfttitlesectrue</code>	18
<code>\@cftasnum</code>	225, 229, 265, 336, 396, 455, 507, 559, 613, 713, 815, 867, 925, 1005	<code>\@cfttocbibindfalse</code>	31
<code>\@cftasnumb</code>	227, 229, 266, 337, 397, 456, 508, 560, 614, 714, 816, 868, 926, 1006	<code>\@cfttocbibindtrue</code>	31
<code>\@cftbsnum</code>	225, 229, 264, 335, 395, 454, 506, 558, 612, 712, 814, 866, 924, 1004	<code>\@cfttocfinish</code>	66, 183, 679, 781, 1088, 1098
<code>\@cftdobiblof</code>	656, 675	<code>\@cfttocstart</code>	66, 173, 669, 771, 1080, 1091
<code>\@cftdobiblot</code>	757, 777	<code>\@contentsline@destination</code>	269, 272, 311, 314, 340, 343, 369, 372, 400, 403, 438, 441, 458, 460, 490, 493, 510, 512, 542, 545, 563, 566, 596, 599, 617, 620, 650, 653, 717, 720, 746, 749, 819, 822, 848, 851, 871, 874, 904, 907, 929, 932, 961, 964, 1009, 1012, 1053, 1056
<code>\@cftdobibtoc</code>	80, 179	<code>\@currentHref</code>	1276, 1278
<code>\@cftdopartfalse</code>	237	<code>\@eha</code>	992
<code>\@cftdoparttrue</code>	240, 244	<code>\@firstoftwo</code>	6
<code>\@cfttemp</code>	1126, 1127, 1130, 1132, 1134, 1136, 1138, 1140, 1142, 1144, 1146, 1182, 1183, 1186, 1188, 1190, 1192, 1194, 1196, 1198, 1200, 1202, 1218, 1219, 1222, 1224, 1226, 1228, 1230, 1232, 1234, 1236, 1238	<code>\@highpenalty</code>	250, 326
<code>\@cfthaschapterfalse</code>	20	<code>\@ifclassloaded</code>	14, 15, 16
<code>\@cfthaschaptertrue</code>	26	<code>\@ifpackagelater</code>	33
<code>\@cfthaspartfalse</code>	234	<code>\@ifpackageloaded</code>	18, 31
<code>\@cfthasparttrue</code>	234	<code>\@ifundefined</code>	979, 983
<code>\@cftifundefined</code>	3, 19, 21, 234	<code>\@namedef</code>	994, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1047, 1066, 1069, 1071, 1074, 1079, 1090, 1101, 1108, 1110, 1116, 1118, 1121, 1122, 1123, 1167, 1254
<code>\@cftinitnumwidth</code>	220, 263	<code>\@nameuse</code>	995, 996, 997, 1000, 1003, 1009, 1010, 1012, 1013, 1018, 1022, 1023, 1025, 1026, 1028, 1029, 1031, 1032, 1034, 1035, 1042, 1049, 1053, 1054, 1056, 1057, 1086, 1095, 1105, 1106, 1113, 1114, 1162, 1163, 1168, 1255, 1256, 1257
<code>\@cftkomafalse</code>	13	<code>\@nobreakfalse</code>	277, 281, 1288
<code>\@cftkomatrue</code>	14, 15, 16	<code>\@nobreaktrue</code>	276, 280
<code>\@cftl@subfig</code>	854, 971, 975	<code>\@pnumwidth</code>	204, 312, 370, 439, 491, 543, 597, 651, 747, 849, 905, 962, 1054, 1256
<code>\@cftl@subtab</code>	912, 971, 975	<code>\@restonecolfalse</code>	71
<code>\@cftmakeZtitle</code>	1101	<code>\@restonecoltrue</code>	69
<code>\@cftmakeloftitle</code>	673, 683	<code>\@secondoftwo</code>	8
<code>\@cftmakelottitle</code>	775, 785	<code>\@secpenalty</code>	136, 149, 252, 385
<code>\@cftmaketoctitle</code>	177, 186	<code>\@starttoc</code>	3, 53, 54, 164, 165, 181, 677, 779, 1087, 1096, 1280
<code>\@cftnctocfalse</code>	41	<code>\@starttoc@latex@orig</code>	1280, 1296
<code>\@cftnctoctrue</code>	42	<code>\@tempa</code>	209, 211
<code>\@cftpagestyle</code>	48, 114, 127, 138, 151	<code>\@tempb</code>	210, 211
<code>\@cftpnumoff</code>	1166, 1184, 1186, 1188, 1190, 1192, 1194, 1196, 1198, 1200, 1202, 1204	<code>\@tocextra</code>	87, 663, 764
<code>\@cftpnumon</code>	1220, 1222, 1224, 1226, 1228, 1230, 1232, 1234, 1236, 1238, 1240, 1253		
<code>\@cftquit</code>	10, 24, 28		
<code>\@cftsetindents</code>	1128, 1130, 1132, 1134, 1136, 1138, 1140, 1142, 1144, 1146, 1148, 1161		
<code>\@cftsetsubfig</code>	880, 970, 974		

<code>\@tocrmarg</code>	205, 257, 329, 389, 448, 500, 552, 606, 706, 808, 860, 918, 998, 1270
<code>\c@lofdepth</code>	703, 856, 914
<code>\c@lotdepth</code>	805
<code>\c@tocdepth</code>	238, 242, 325, 381, 445, 497, 549, 603
<code>\c@X</code>	979
<code>\ext@Z</code>	1066
<code>\hb@xt@</code>	223
<code>\if@bibchapter</code>	82, 658, 759
<code>\if@cft@patch@starttoc</code> ...	1290, 1300
<code>\if@cft@dopart</code>	232
<code>\if@cfthaschapter</code>	10, 67, 75, 98, 189, 196, 239, 243, 249, 275, 323, 347, 382, 412, 467, 519, 573, 627, 686, 693, 788, 795, 1081, 1104
<code>\if@cfthaspart</code>	233, 235, 287
<code>\if@cftkoma</code>	12, 58, 198, 201, 270, 317, 375, 695, 698, 797, 800, 1070, 1107, 1115
<code>\if@cftnctoc</code> ...	41, 162, 667, 769, 1078
<code>\if@cftsubfigopt</code> ...	44, 752, 969, 973
<code>\if@cfttitlesec</code>	17, 270
<code>\if@cfttocbibind</code> ...	29, 178, 674, 776
<code>\if@compatibility</code>	279
<code>\if@dotoclof</code>	657
<code>\if@dotoclot</code>	758
<code>\if@dotocloc</code>	81
<code>\if@files</code>	1284
<code>\if@restonecol</code>	76
<code>\if@twocolumn</code>	68
<code>\l@</code>	1015
<code>\l@chapter</code>	323
<code>\l@figure</code>	702
<code>\l@paragraph</code>	548
<code>\l@part</code>	232
<code>\l@section</code>	380
<code>\l@subfigure</code>	855
<code>\l@subparagraph</code>	602
<code>\l@subsection</code>	444
<code>\l@subsubsection</code>	496
<code>\l@subtable</code>	913
<code>\l@table</code>	804
<code>\l@X</code>	994
<code>\OLD@starttoc</code>	164, 168
<code>\protected@file@percent</code> ..	1276, 1279
<code>\size@chapter</code>	198, 695, 797, 1108
<code>\size@section</code>	201, 698, 800, 1116
<code>\toclevel@subfig</code>	881
<code>\toclevel@subfigure</code>	856, 871, 874, 896, 904, 907
<code>\toclevel@subtable</code>	914, 929, 932, 938
<code>\toclevel@X</code>	1046
<code>\textit</code>	1265, 1271
<code>\theX</code>	14
<code>\theX</code>	979
<code>\thispagestyle</code>	49
<code>\tocloftpagestyle</code>	7
<code>\tocloftpagestyle</code>	48
<code>\topskip</code>	112
<code>\twocolumn</code>	76
U	
<code>\UseHookWithArguments</code>	224, 226, 268, 271, 310, 313, 339, 342, 368, 371, 399, 402, 437, 440, 458, 460, 489, 492, 510, 512, 541, 544, 562, 565, 595, 598, 616, 619, 649, 652, 716, 719, 745, 748, 818, 821, 847, 850, 870, 873, 903, 906, 928, 931, 960, 963, 1008, 1011, 1052, 1055
<code>\UseName</code>	112, 113, 115, 117, 125, 126, 128, 130, 137, 140, 141, 142, 144, 150, 153, 154, 155, 157, 1072, 1075, 1076, 1082, 1084
<code>\UseStructureName</code>	1267
<code>\UseTaggingSocket</code>	306, 308, 364, 366, 433, 435, 485, 487, 537, 539, 591, 593, 645, 647, 741, 743, 843, 845, 899, 901, 956, 958, 1048, 1050, 1167, 1267
Z	
<code>\Zdepth</code>	8
<code>\Zdepth</code>	1066
<code>\ZZdepth</code>	14