

The `tocloft` package^{*}

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

2026/09/12

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	43
3.2	New list of...	46
3.3	Switching page numbering	51
3.4	Experimental utilities	53
	Index	57

List of Figures

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

^{*}This file (`tocloft.dtx`) has version number v3.0c, last revised 2026/09/12. 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 $\text{\LaTeX}$ 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 $\text{\LaTeX}$. 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 $\text{\LaTeX}$'s methods

This is a general description of how $\text{\LaTeX}$ 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` $\text{\LaTeX}$ generates a `.toc` file if the document contains a `\tableofcontents` command. The sectioning commands² put entries into the `.toc` file by calling the $\text{\LaTeX}$ `\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.

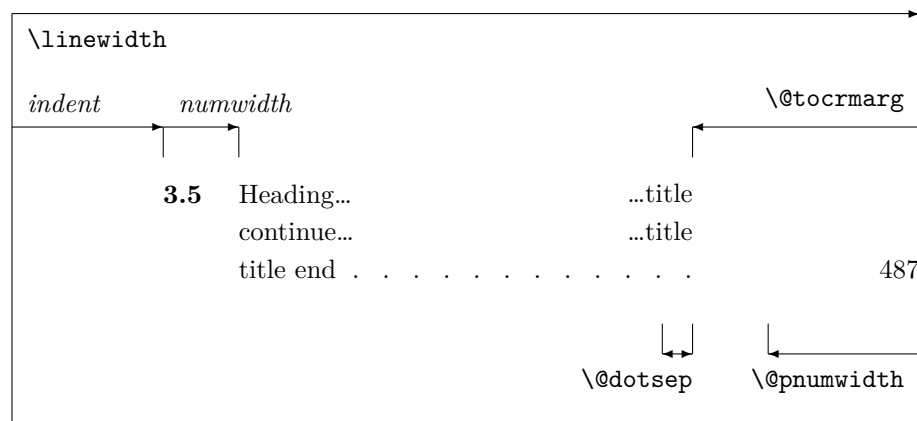


Figure 1: Layout of a ToC (LoF, LoT) entry

`\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 `\cfttoccititlename`, `\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 idiosyncrasies, `\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}{\hfill}

```

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

```

\renewcommand{\cftXleader}{\hfill}
\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]{\hfill}
\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}{ans}{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}{\%
34       \PackageWarning{tocloft}{%
35         You are using a version of the tocbibind package\MessageBreak
36         that is not compatible with tocloft.\MessageBreak
37         The results may be surprising.\MessageBreak
38         Consider installing the current version of tocbibind.}}
39     \fi
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
\cft@listname
103 \newcommand\cft@listname{}
104 \IfDocumentMetadataTF
105 {
106   \@ifundefined{IfTemplateExistsTF}
107   {
108     \DeclareInstance{headformat}{cftchapter}{display}
109     {
110       , heading-indent = Opt
111       , prefix-number-sep = \WordSpaceAmount{1}
112       , title-format = #1\UseName{cftafter\cft@listname title}
113     }
114     \DeclareDocumentCommand \cftchapter {0}{m}
115     {%
116       \renewcommand\cft@listname{#2}%
117       \chapter
118       [
119         toc={},
120         placement=normal,
121         unnumbered,
122         after-penalty-vspace=\dimeval{\UseName{cftbefore#2titleskip}-\topskip},
123         after-vspace=\UseName{cftafter#2titleskip},
124         final-code=\@cftpagestyle\@afterheading,
125         heading-decls=\UseName{cft#2titlefont},
126         headformat-instance=cftchapter,
127         #1
128       ] {\UseName{cft#2titlename}}%
129     }
130     \DeclareInstance{headformat}{cftsection}{hang}
131     {
132       , heading-indent = Opt
133       , title-format = #1\UseName{cftafter\cft@listname title}
134     }
135     \DeclareDocumentCommand \cftsection {0}{m}
136     {%
137       \renewcommand\cft@listname{#2}%
138       \section[
139         toc={},
140         placement=normal,
141         unnumbered,
142         before-vspace=\dimeval{\UseName{cftbefore#2titleskip}-\topskip},
143         after-vspace=\UseName{cftafter#2titleskip},
144         final-code=\@cftpagestyle\@afterheading,
145         heading-decls=\tagpdfparaOff\UseName{cft#2titlefont}\tagpdfparaOn,
146         headformat-instance=cftsection,
147         #1
148       ] {\UseName{cft#2titlename}}%
149     }
150   }

```

New code for the next release:

```

151 {

```

```

152 \DeclareInstance{headformat}{cftchapter}{display}
153 {
154   , heading-indent      = Opt
155   , separator-a         = \nobreakspace
156 }
157 \DeclareDocumentCommand \cftchapter {0}{m}
158 {%
159   \renewcommand\cft@listname{#2}%
160   \chapter
161   [
162     toc={},
163     placement=normal,
164     unnumbered,
165     after-penalty-vspace=\dimeval{\UseName{cftbefore#2titleskip}-\topskip},
166     after-vspace=\UseName{cftafter#2titleskip},
167     final-code=\@cftpagestyle\@afterheading,
168     heading-decls=\UseName{cft#2titlefont},
169     headformat-instance=cftchapter,
170     title-format      = ##1\UseName{cftafter\cft@listname title},
171     #1
172   ] {\UseName{cft#2titlename}}%
173 }
174 \DeclareInstance{headformat}{cftsection}{hang}
175 {
176   , heading-indent = Opt
177 }
178 \DeclareDocumentCommand \cftsection {0}{m}
179 {%
180   \renewcommand\cft@listname{#2}%
181   \section[
182     toc={},
183     placement=normal,
184     unnumbered,
185     before-vspace=\dimeval{\UseName{cftbefore#2titleskip}-\topskip},
186     after-vspace=\UseName{cftafter#2titleskip},
187     final-code=\@cftpagestyle\@afterheading,
188     heading-decls=\UseName{cft#2titlefont},
189     headformat-instance=cftsection,
190     title-format      = ##1\UseName{cftafter\cft@listname title},
191     #1
192   ] {\UseName{cft#2titlename}}%
193 }
194 }
195 }
196 {
197 \DeclareDocumentCommand \cftchapter {0}{m}
198 {%
199   \addpenalty\@secpenalty
200   \vspace*{\UseName{cftbefore#2titleskip}}%
201   \@cftpagestyle
202   {\interlinepenalty\@M
203    {\UseName{cft#2titlefont}\UseName{cft#2titlename}}%
204    {\UseName{cftafter#2title}}%

```

```

205     \UseName{cftmark#2}%
206     \par\nobreak
207     \vskip \UseName{cftafter#2titleskip}%
208     \@afterheading}%
209 }
210 \DeclareDocumentCommand \cftsection {0}{m}
211 {%
212     \addpenalty\@secpenalty
213     \vspace{\UseName{cftbefore#2titleskip}}%
214     \@cftpagestyle
215     {\interlinepenalty\@M
216     {\UseName{cft#2titlefont}}\UseName{cft#2titlename}}}%
217     {\UseName{cftafter#2title}}}%
218     \UseName{cftmark#2}%
219     \par\nobreak
220     \vskip \UseName{cftafter#2titleskip}%
221     \@afterheading}%
222 }
223 }

```

(End of definition for `\cftchapter`, `\cftsection`, and `\cft@listname`. 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.

```

224 \AtBeginDocument{%
225 \if@cftnctoc
226 % ensure \cftparskip is still set properly
227 \let\OLD@starttoc\@starttoc
228 \renewcommand{\@starttoc}[1]{%
229     \begingroup
230     \parskip=\cftparskip
231     \OLD@starttoc{#1}%
232     \endgroup
233 }
234 \else
235 \renewcommand{\tableofcontents}{%
236     \@cfttocstart

```

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

```

237     \par
238     \begingroup
239     \parindent\z@ \parskip\cftparskip
240     \@cftmaketoctitle

```

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

```

241     \if@cfttocbibind
242     \cftdobibtoc
243 \fi

```


Finally, read the .toc file and finish up.

```

244     \@starttoc{toc}%
245     \endgroup
246     \@cfttocfinish}
247 \fi
248 }
```

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

\@cftmaketoc This command typesets the title for the ToC.

```

249 \newcommand{\@cftmaketoc}{\cfttitlecommand{toc}}%
```

(End of definition for \@cftmaketoc.)

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

```

\cftaftertoctitleskip
250 \newlength{\cftbeforetoctitleskip}
251 \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*.

```

252 \ifcftchapter
253   \setlength{\cftbeforetoctitleskip}{50pt}
254   \setlength{\cftaftertoctitleskip}{40pt}
255 \else
256   \setlength{\cftbeforetoctitleskip}{3.5ex \@plus 1ex \@minus .2ex}
257   \setlength{\cftaftertoctitleskip}{2.3ex \@plus .2ex}
258 \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).

```

259 \ifcftchapter
260   \newcommand{\cfttoctitlefont}{\normalfont\Huge\bfseries}
261   \ifcftkoma\renewcommand{\cfttoctitlefont}{\size@chapter\sectfont}\fi
262 \else
263   \newcommand{\cfttoctitlefont}{\normalfont\Large\bfseries}
264   \ifcftkoma\renewcommand{\cfttoctitlefont}{\size@section\sectfont}\fi
265 \fi
266 \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
267 \newcommand{\cftsetpnumwidth}[1]{\renewcommand{\@pnumwidth}{#1}}
268 \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.

```

269 \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 `ccaption` package.

```

270 \providecommand{\cftdot}{.}
271 \providecommand{\cftdotfill}[1]{%
272   \def\@tempa{#1}%
273   \def\@tempb{\cftnodots}%
274   \ifx\@tempa\@tempb
275     \hfill
276   \else
277     \leaders\hbox{$\m@th\mkern #1 mu\hbox{\cftdot}\mkern #1 mu$}\hfill
278   \fi
279 }
```

(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`
`\cftnodots` 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.)

```

280 \providecommand{\cftdotsep}{4.5}
281 \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 `\addcontentsline` 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`.

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

```
282 \newcommand{\cftparrfillskip}{\parfillskip=0pt plus1fil}
```

(End of definition for `\cftparrfillskip`. 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`.

```
283 \newcommand\@cftinitnumwidth[1]{}
284 \renewcommand{\numberline}[1]{%
285   \@cftinitnumwidth{#1}%
286   \hb@xt@\@tempdima{%
287     \UseHookWithArguments{contentsline/number/before}{1}{#1}%
288     \@cftbsnum #1\@cftasnum\hfil
289     \UseHookWithArguments{contentsline/number/after}{1}{#1}%
290     \@cftasnumb
291   }
```

(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

```
292 \newcommand{\@cftbsnum}{}
293 \newcommand{\@cftasnum}{}
294 \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@subsection` 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.

```
295 \newif\if@cftdopart
296 \newif\if@cftthaspart
297 \@cftifundefined{part}{\@cftthaspartfalse}{\@cftthasparttrue}
298 \if@cftthaspart
299 \renewcommand*{\l@part}[2]{%
300   \@cftdopartfalse
301   \ifnum \c@tocdepth >-2\relax

```

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

```

302 \if@cfthaschapter
303 \@cftdoparttrue
304 \fi
305 \ifnum \c@tocdepth >\m@ne
306 \if@cfthaschapter\else
307 \@cftdoparttrue
308 \fi
309 \fi
310 \fi
311 \if@cftdopart
312 \if@cfthaschapter
313 \addpenalty{-\@highpenalty}%
314 \else
315 \addpenalty\@secpenalty
316 \fi
317 \addvspace{\cftbeforepartskip}%
318 \begingroup
319 {\leftskip \cftpartindent\relax
320 \rightskip \@tocrmarg
321 \parfillskip -\rightskip
322 \parindent \cftpartindent\relax\@afterindenttrue
323 \interlinepenalty\@M
324 \leavevmode
325 \@tempdima \cftpartnumwidth\relax
326 \let\@cftinitnumwidth\cftpartinitnumwidth
327 \let\@cftbsnum \cftpartpresnum
328 \let\@cftasnum \cftpartaftersnum
329 \let\@cftasnumb \cftpartaftersnumb
330 \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.

```

331 \UseHookWithArguments{contentsline/text/before}{4}
332 {-1}{#1}{#2}{\@contentsline@destination}%
333 {\cftpartfont \if@cftkoma\else\if@cfttitlesec\else\cftpartpresnum\fi\fi #1}%
334 \UseHookWithArguments{contentsline/text/after}{4}
335 {-1}{#1}{#2}{\@contentsline@destination}%
336 \cftpartfillnum{#2}}
337 \nobreak
338 \if@cfthaschapter
339 \global\@nobreaktrue
340 \everypar{\global\@nobreakfalse\everypar{}}%
341 \else
342 \if@compatibility
343 \global\@nobreaktrue
344 \everypar{\global\@nobreakfalse\everypar{}}%
345 \fi
346 \fi
347 \endgroup
348 \fi}
349 \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.

```

350 \if@cfthaspart
351   \newlength{\cftbeforepartskip}
352   \setlength{\cftbeforepartskip}{2.25em \@plus\p@}
353   \newlength{\cftpartnumwidth}
354   \setlength{\cftpartnumwidth}{3em}
355   \newcommand{\cftpartfont}{\large\bfseries}
356   \newcommand{\cftpartinitnumwidth}[1]{\settowidth\@tempdima{#1}}
357   \newcommand{\cftpartpresnum}{}
358   \newcommand{\cftpartaftersnum}{}
359   \newcommand{\cftpartaftersnumb}{\hspace{1em}}
360   \newcommand{\cftpartleader}{\large\bfseries\cftdotfill{\cftpartdotsep}}
361   \newcommand{\cftpartdotsep}{\cftnodots}
362   \newcommand{\cftpartpagefont}{\large\bfseries}
363   \newcommand{\cftpartafterpnum}{}
364   \newcommand{\cftpartpnumfinal}{\cftpnumfinaldefault}
365   \newlength{\cftpartindent}
366   \setlength{\cftpartindent}{0em}
367   \newcommand{\cftpartfillnum}[1]{%
368     {%
369       \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
370       \cftpartleader
371       \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
372     }%
373     {\UseHookWithArguments{contentsline/page/before}{4}
374      {-1}{#1}{\@contentsline@destination}%
375      \makebox[\@pnumwidth][\cftpnumalign]{\cftpartpagefont #1}%
376      \UseHookWithArguments{contentsline/page/after}{4}
377      {-1}{#1}{\@contentsline@destination}%
378      \cftpartafterpnum\cftpartpnumfinal\par}%
379   }

```

koma classes use some different settings.

```

380 \if@cfTkoma
381   \setlength{\cftpartnumwidth}{2em}
382   \renewcommand{\cftpartfont}{\sectfont\large}
383   \renewcommand{\cftpartpagefont}{\sectfont\large}
384 \fi
385 \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.

```

386 \if@cfthaschapter
387 \renewcommand*{\l@chapter}[2]{%
388   \ifnum \c@tocdepth > \m@ne
389     \addpenalty{-\@highpenalty}%
390     \vskip \cftbeforechapskip
391     {\leftskip \cftchapindent\relax
392      \rightskip \@tocrmarg
393      \parfillskip -\rightskip

```

```

394 \parindent \cftchapindent\relax\@afterindenttrue
395 \interlinepenalty\@M
396 \leavevmode
397 \@tempdima \cftchapnumwidth\relax
398 \let\@cftbsnum \cftchappresnum
399 \let\@cftasnum \cftchapaftersnum
400 \let\@cftasnumb \cftchapaftersnumb
401 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
402 \UseHookWithArguments{contentsline/text/before}{4}
403 {0}{#1}{#2}{\@contentsline@destination}%
404 {\cftchapfont #1}\nobreak
405 \UseHookWithArguments{contentsline/text/after}{4}
406 {0}{#1}{#2}{\@contentsline@destination}%
407 \cftchapfillnum{#2}}%
408 \fi}%
409 \fi

```

(End of definition for \l@chapter.)

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

```

\cftchapindent
\cftchapnumwidth
\cftchapfont
\cftchappresnum
\cftchapaftersnum
\cftchapaftersnumb
\cftchapleader
\cftchapdotsep
\cftchappagefont
\cftchapafterpnum
\cftchappnumfinal
\cftchapfillnum
410 \if@cfthaschapter
411 \newlength{\cftbeforechapskip}
412 \setlength{\cftbeforechapskip}{1.0em \@plus\p@}
413 \newlength{\cftchapindent}
414 \setlength{\cftchapindent}{0em}
415 \newlength{\cftchapnumwidth}
416 \setlength{\cftchapnumwidth}{1.5em}
417 \newcommand{\cftchapfont}{\bfseries}
418 \newcommand{\cftchappresnum}{}
419 \newcommand{\cftchapaftersnum}{}
420 \newcommand{\cftchapaftersnumb}{}
421 \newcommand{\cftchapleader}{\bfseries\cftdotfill{\cftchapdotsep}}
422 \newcommand{\cftchapdotsep}{\cftnodots}
423 \newcommand{\cftchappagefont}{\bfseries}
424 \newcommand{\cftchapafterpnum}{}
425 \newcommand{\cftchappnumfinal}{\cftpnumfinaldefault}
426 \newcommand{\cftchapfillnum}[1]{%
427 {\UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
428 \cftchapleader
429 \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
430 }\nobreak
431 \UseHookWithArguments{contentsline/page/before}{4}
432 {0}{#1}{\@contentsline@destination}%
433 \makebox[\@pnumwidth][\cftpnumalign]{\cftchappagefont #1}%
434 \UseHookWithArguments{contentsline/page/after}{4}
435 {0}{#1}{\@contentsline@destination}%
436 \cftchapafterpnum\cftchappnumfinal\par
437 }

```

koma classes have different chapter settings.

```

438 \if@cfkoma
439 \renewcommand{\cftchapfont}{\sectfont}
440 \fi

```

441 `\fi`

442

(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 section heading. It is a parameterised copy of the default `\l@section` (see `classes.dtx` for the original definition).

```

443 \renewcommand*{\l@section}[2]{%
444   \ifnum \c@tocdepth >\z@
445     \if@cfthaschapter
446       \vskip \cftbeforesecskip
447     \else
448       \addpenalty\@secpenalty
449       \addvspace{\cftbeforesecskip}
450     \fi
451     {\leftskip \cftsecindent\relax
452      \rightskip \@tocrmarg
453      \parfillskip -\rightskip
454      \parindent \cftsecindent\relax\@afterindenttrue
455      \interlinepenalty\@M
456      \leavevmode
457      \@tempdima \cftsecnumwidth\relax
458      \let\@cftbsnum \cftsecpresnum
459      \let\@cftasnum \cftsecaftersnum
460      \let\@cftasnumb \cftsecaftersnumb
461      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
462      \UseHookWithArguments{contentsline/text/before}{4}
463      {1}{#1}{#2}{\@contentsline@destination}%
464      {\cftsecfont #1}\nobreak
465      \UseHookWithArguments{contentsline/text/after}{4}
466      {1}{#1}{#2}{\@contentsline@destination}%
467      \cftsecfillnum{#2}}%
468   \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
\cftsecfont
\cftsecpresnum
\cftsecaftersnum
\cftsecaftersnumb
\cftsecleader
\cftsecdotsep
\cftsecpagefont
\cftsecfterpnum
\cftsecpnumfinal
\cftsecfillnum
469 \newlength{\cftbeforesecskip}
470 \newlength{\cftsecindent}
471 \newlength{\cftsecnumwidth}
472 \newcommand{\cftsecpresnum}{}
473 \newcommand{\cftsecaftersnum}{}
474 \newcommand{\cftsecaftersnumb}{}
475 \if@cfthaschapter
476   \setlength{\cftbeforesecskip}{\z@ \@plus.2\p@}
477   \setlength{\cftsecindent}{1.5em}
478   \setlength{\cftsecnumwidth}{2.3em}
479   \newcommand{\cftsecfont}{\normalfont}
480   \newcommand{\cftsecleader}{\normalfont\cftdotfill{\cftsecdotsep}}
481   \newcommand{\cftsecdotsep}{\cftdotsep}
482   \newcommand{\cftsecpagefont}{\normalfont}
483 \else

```

```

484 \setlength{\cftbeforesecskip}{1.0em \@plus\p@}
485 \setlength{\cftsecindent}{0em}
486 \setlength{\cftsecnumwidth}{1.5em}
487 \newcommand{\cftsecfont}{\bfseries}
488 \newcommand{\cftsecleader}{\bfseries\cftdotfill{\cftsecdotsep}}
489 \newcommand{\cftsecdotsep}{\cftnodots}
490 \newcommand{\cftsecpagefont}{\bfseries}
491 \fi
492 \newcommand{\cftsecafterpnum}{}
493 \newcommand{\cftsecpnumfinal}{\cftpnumfinaldefault}
494 \newcommand{\cftsecfillnum}[1]{%
495   {%
496     \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
497     \cftsecleader
498     \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
499   }\nobreak
500   \UseHookWithArguments{contentsline/page/before}{4}
501   {1}{\#1}{\@contentsline@destination}%
502   \makebox[\@pnumwidth][\cftpnumalign]{\cftsecpagefont #1}%
503   \UseHookWithArguments{contentsline/page/after}{4}
504   {1}{\#1}{\@contentsline@destination}%
505   \cftsecafterpnum\cftsecpnumfinal\par
506 }

```

(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).

```

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

```

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

```

508 \ifnum \c@tocdepth >\@ne

```

Add some vertical space.

```

509 \vskip \cftbeforesubsecskip

```

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

```

510 {\leftskip \cftsubsecindent\relax

```

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

```

511 \rightskip \@tocrmarg

```

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

```

512 \parfillskip -\rightskip

```

Set the paragraph indent to the entry's indentation.

```

513 \parindent \cftsubsecindent\relax\@afterindenttrue

```

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

```

514 \interlinepenalty\@M

```


Make sure that we have left vertical mode.

```
515      \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.

```
516      \@tempdima \cftsubsecnumwidth\relax
517      \let\@cftbsnum \cftsubsecpresnum
518      \let\@cftasnum \cftsubsecaftersnum
519      \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.

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

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

```
521      \UseHookWithArguments{contentsline/text/before}{4}{2}{#1}{#2}{\@contentsline@destination}
522      {\cftsubsecfont #1}\nobreak
523      \UseHookWithArguments{contentsline/text/after}{4}{2}{#1}{#2}{\@contentsline@destination}
```

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

```
524      \cftsubsecfillnum{#2}}%
525      \fi}
```

(End of definition for \l@section.)

```
\cftbeforesubsecskip
\cftsubsecindent
\cftsubsecnumwidth
\cftsubsecfont
\cftsubsecpresnum
\cftsubsecaftersnum
\cftsubsecaftersnumb
\cftsubsecleader
\cftsubsecdotsep
\cftsubsecpagefont
\cftsubsecafterpnum
\cftsubsecpnumfinal
```

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

```
526 \newlength{\cftbeforesubsecskip}
527 \setlength{\cftbeforesubsecskip}{\z@ \@plus.2\p@}
528 \newlength{\cftsubsecindent}
529 \newlength{\cftsubsecnumwidth}
530 \if@cfthaschapter
531 \setlength{\cftsubsecindent}{3.8em}
532 \setlength{\cftsubsecnumwidth}{3.2em}
533 \else
534 \setlength{\cftsubsecindent}{1.5em}
535 \setlength{\cftsubsecnumwidth}{2.3em}
536 \fi
537 \newcommand{\cftsubsecfont}{\normalfont}
538 \newcommand{\cftsubsecpresnum}{}
539 \newcommand{\cftsubsecaftersnum}{}
540 \newcommand{\cftsubsecaftersnumb}{}
541 \newcommand{\cftsubsecleader}{\normalfont\cftdotfill{\cftsubsecdotsep}}
542 \newcommand{\cftsubsecdotsep}{\cftdotsep}
543 \newcommand{\cftsubsecpagefont}{\normalfont}
544 \newcommand{\cftsubsecafterpnum}{}
545 \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.

```

546 \newcommand{\cftsubsecfillnum}[1]{%
547   {%
548     \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
549     \cftsubsecleader
550     \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
551   }\nobreak
552   \UseHookWithArguments{contentsline/page/before}{4}
553   {2}{\@pnumwidth}{\@pnumwidth}{\cftsubsecpagefont #1}%
554   \makebox[\@pnumwidth]{\cftsubsecpagefont #1}%
555   \UseHookWithArguments{contentsline/page/after}{4}
556   {2}{\@pnumwidth}{\cftsubsecpagefont #1}%
557   \cftsubsecafterpnum\cftsubsecpnumfinal\par
558 }

```

(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).

```

559 \renewcommand*\l@subsubsection[2]{%
560   \ifnum \c@tocdepth > \tw@
561     \vskip \cftbeforesubsubsecskip
562     {\leftskip \cftsubsubsecindent\relax
563       \rightskip \@tocrmarg
564       \parfillskip -\rightskip
565       \parindent \cftsubsubsecindent\relax\@afterindenttrue
566       \interlinepenalty\@M
567       \leavevmode
568       \@tempdima \cftsubsubsecnumwidth\relax
569       \let\@cftbsnum \cftsubsubsecpresnum
570       \let\@cftasnum \cftsubsubsecaftersnum
571       \let\@cftasnumb \cftsubsubsecaftersnumb
572       \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
573       \UseHookWithArguments{contentsline/text/before}{4}{3}{#1}{#2}{\@contentsline@destination}
574       {\cftsubsubsecfont #1}\nobreak
575       \UseHookWithArguments{contentsline/text/after}{4}{3}{#1}{#2}{\@contentsline@destination}
576       \cftsubsubsecfillnum{#2}}%
577   \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

```

```

578 \newlength{\cftbeforesubsubsecskip}
579 \setlength{\cftbeforesubsubsecskip}{\z@ \@plus.2\p@}
580 \newlength{\cftsubsubsecindent}
581 \newlength{\cftsubsubsecnumwidth}
582 \ifcftaschapter
583   \setlength{\cftsubsubsecindent}{7.0em}
584   \setlength{\cftsubsubsecnumwidth}{4.1em}
585 \else

```

```

586 \setlength{\cftsubsubsecindent}{3.8em}
587 \setlength{\cftsubsubsecnumwidth}{3.2em}
588 \fi
589 \newcommand{\cftsubsubsecfont}{\normalfont}
590 \newcommand{\cftsubsubsecpresnum}{}
591 \newcommand{\cftsubsubsecaftersnum}{}
592 \newcommand{\cftsubsubsecaftersnumb}{}
593 \newcommand{\cftsubsubsecleader}{\normalfont\cftdotfill{\cftsubsubsecdotsep}}
594 \newcommand{\cftsubsubsecdotsep}{\cftdotsep}
595 \newcommand{\cftsubsubsecpagefont}{\normalfont}
596 \newcommand{\cftsubsubsecafterpnum}{}
597 \newcommand{\cftsubsubsecpnumfinal}{\cftpnumfinaldefault}
598 \newcommand{\cftsubsubsecfillnum}[1]{%
599   {%
600     \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
601     \cftsubsubsecleader
602     \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
603   }\nobreak
604   \UseHookWithArguments{contentsline/page/before}{4}
605     {3}{\cftsubsubsecfillnum}{\cftsubsubsecfillnum}%
606   \makebox[\cftsubsubsecnumwidth]{\cftsubsubsecfillnum\cftsubsubsecfillnum}%
607   \UseHookWithArguments{contentsline/page/after}{4}
608     {3}{\cftsubsubsecfillnum}{\cftsubsubsecfillnum}%
609   \cftsubsubsecafterpnum\cftsubsubsecpnumfinal\par
610 }

```

(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).

```

611 \renewcommand*{\l@paragraph}[2]{%
612   \ifnum \c@tocdepth >3\relax
613     \vskip \cftbeforeparaskip
614     {\leftskip \cftparaindent\relax
615       \rightskip \@tocrmarg
616       \parfillskip -\rightskip
617       \parindent \cftparaindent\relax\@afterindenttrue
618       \interlinepenalty\@M
619       \leavevmode
620       \@tempdima \cftparanumwidth\relax
621       \let\cftbsnum \cftparapresnum
622       \let\cftasnum \cftparaaftersnum
623       \let\cftasnumb \cftparaaftersnumb
624       \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
625       \UseHookWithArguments{contentsline/text/before}{4}
626         {4}{\cftbsnum}{\cftasnum}{\cftasnumb}{\cftparaindent}%
627       {\cftparafont \cftbsnum}\nobreak
628       \UseHookWithArguments{contentsline/text/after}{4}
629         {4}{\cftbsnum}{\cftasnum}{\cftasnumb}{\cftparaindent}%
630       \cftparaftersnum\cftparaftersnumb\par
631     }

```

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

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

`\cftparaindent`

`\cftparanumwidth` 632 `\newlength{\cftbeforeparaskip}`

`\cftparafont` 633 `\setlength{\cftbeforeparaskip}{\z@ \@plus.2\p@}`

`\cftparapresnum` 634 `\newlength{\cftparaindent}`

`\cftparaaftersnum` 635 `\newlength{\cftparanumwidth}`

`\cftparaaftersnumb` 636 `\ifcftthaschapter`

`\cftparaleader` 637 `\setlength{\cftparaindent}{10em}`

`\cftparadotsep` 638 `\setlength{\cftparanumwidth}{5em}`

`\cftparapagefont` 639 `\else`

`\cftparaafterpnum` 640 `\setlength{\cftparaindent}{7.0em}`

`\cftparapnumfinal` 641 `\setlength{\cftparanumwidth}{4.1em}`

`\cftparafillnum` 642 `\fi`

643 `\newcommand{\cftparafont}{\normalfont}`

644 `\newcommand{\cftparapresnum}{}`

645 `\newcommand{\cftparaaftersnum}{}`

646 `\newcommand{\cftparaaftersnumb}{}`

647 `\newcommand{\cftparaleader}{\normalfont\cftdotfill{\cftparadotsep}}`

648 `\newcommand{\cftparadotsep}{\cftdotsep}`

649 `\newcommand{\cftparapagefont}{\normalfont}`

650 `\newcommand{\cftparaafterpnum}{}`

651 `\newcommand{\cftparapnumfinal}{\cftpnumfinaldefault}`

652 `\newcommand{\cftparafillnum}[1]{%`

653 `{%`

654 `\UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%`

655 `\cftparaleader`

656 `\ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%`

657 `}\nobreak`

658 `\UseHookWithArguments{contentsline/page/before}{4}`

659 `{4}{\#1}{\@contentsline@destination}%`

660 `\makebox[\@pnumwidth][\cftpnumalign]{\cftparapagefont #1}%`

661 `\UseHookWithArguments{contentsline/page/after}{4}`

662 `{4}{\#1}{\@contentsline@destination}%`

663 `\cftparaafterpnum\cftparapnumfinal\par`

664 `}`

(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).

665 `\renewcommand*{\l@subparagraph}[2]{%`

666 `\ifnum \c@tocdepth >4\relax`

667 `\vskip \cftbeforesubparaskip`

668 `{\leftskip \cftsubparaindent\relax`

669 `\rightskip \@tocrmarg`

670 `\parfillskip -\rightskip`

671 `\parindent \cftsubparaindent\relax\@afterindenttrue`

672 `\interlinepenalty\@M`

673 `\leavevmode`

674 `\@tempdima \cftsubparanumwidth\relax`

675 `\let\cftbsnum \cftsubparapresnum`

676 `\let\cftasnum \cftsubparaaftersnum`

677 `\let\cftasnumb \cftsubparaaftersnumb`

```

678 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
679 \UseHookWithArguments{contentsline/text/before}{4}
680 {5}{#1}{#2}{\@contentsline@destination}%
681 {\cftsubparafont #1}\nobreak
682 \UseHookWithArguments{contentsline/text/after}{4}
683 {5}{#1}{#2}{\@contentsline@destination}%
684 \cftsubparafillnum{#2}}%
685 \fi}

```

(End of definition for \l@subparagraph.)

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

```

\cftsubparaindent
\cftsubparanumwidth
\cftsubparafont
\cftsubparapresnum
\cftsubparaaftersnum
\cftsubparaaftersnumb
\cftsubparaleader
\cftsubparadotsep
\cftsubparapagefont
\cftsubparaafterpnum
\cftsubparapnumfinal
\cftsubparafillnum
686 \newlength{\cftbeforesubparaskip}
687 \setlength{\cftbeforesubparaskip}{\z@ \@plus.2\p@}
688 \newlength{\cftsubparaindent}
689 \newlength{\cftsubparanumwidth}
690 \if@cftthaschapter
691 \setlength{\cftsubparaindent}{12em}
692 \setlength{\cftsubparanumwidth}{6em}
693 \else
694 \setlength{\cftsubparaindent}{10em}
695 \setlength{\cftsubparanumwidth}{5em}
696 \fi
697 \newcommand{\cftsubparafont}{\normalfont}
698 \newcommand{\cftsubparapresnum}{}
699 \newcommand{\cftsubparaaftersnum}{}
700 \newcommand{\cftsubparaaftersnumb}{}
701 \newcommand{\cftsubparaleader}{\normalfont\cftdotfill{\cftsubparadotsep}}
702 \newcommand{\cftsubparadotsep}{\cftdotsep}
703 \newcommand{\cftsubparapagefont}{\normalfont}
704 \newcommand{\cftsubparaafterpnum}{}
705 \newcommand{\cftsubparapnumfinal}{\cftpnumfinaldefault}
706 \newcommand{\cftsubparafillnum}[1]{%
707   {%
708     \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
709     \cftsubparaleader
710     \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
711   }\nobreak
712   \UseHookWithArguments{contentsline/page/before}{4}
713   {5}{#1}{\@contentsline@destination}%
714   \makebox[\@pnumwidth][\cftpnumalign]{\cftsubparapagefont #1}%
715   \UseHookWithArguments{contentsline/page/after}{4}
716   {5}{#1}{\@contentsline@destination}%
717   \cftsubparaafterpnum\cftsubparapnumfinal\par
718 }

```

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

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

```

719 \newcommand{\cftdobiblof}{%
720 \if@dotoclof
721 \if@bibchapter
722 \phantomsection

```

```

723     \addcontentsline{toc}{chapter}{\cftloftitlename}
724   \else
725     \phantomsection
726     \addcontentsline{toc}{\@tocextra}{\cftloftitlename}
727   \fi
728 \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).

```

729 \AtBeginDocument{%
730 \if@cftnctoc\else
731 \renewcommand{\listoffigures}{%
732   \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.

```

733   \par
734   \begingroup
735     \parindent\z@ \parskip\cftparskip
736     \cftmakeloftitle
737     \if@cfttocbibind
738       \cftdobiblof
739     \fi
740     \starttoc{lof}%
741   \endgroup

```

Finally, restore any multicolumn typesetting.

```

742   \cfttocfinish}%
743 \fi
744 }
745

```

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

\cftmakeloftitle This command typesets the title for the LoF.

```

746 \newcommand{\cftmakeloftitle}{\cfttitlecommand{lof}}%

```

(End of definition for \cftmakeloftitle.)

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

```

747 \newlength{\cftbeforeloftitleskip}
748 \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*.

```

749 \if@cfthaschapter
750   \setlength{\cftbeforeloftitleskip}{50pt}
751   \setlength{\cftafterloftitleskip}{40pt}
752 \else
753   \setlength{\cftbeforeloftitleskip}{3.5ex \@plus 1ex \@minus .2ex}
754   \setlength{\cftafterloftitleskip}{2.3ex \@plus .2ex}
755 \fi

```

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

`\cftlofttitlefont` The LoF title is typeset in the style given by `\cftlofttitlefont`. The macro
`\cftafterlofttitle` `\cftafterlofttitle` 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 `\cftlofttitlefont` will make the title flushright).

```

756 \if@cfthaschapter
757   \newcommand{\cftlofttitlefont}{\normalfont\Huge\bfseries}
758   \if@cfTkoma\renewcommand{\cftlofttitlefont}{\size@chapter\sectfont}\fi
759 \else
760   \newcommand{\cftlofttitlefont}{\normalfont\Large\bfseries}
761   \if@cfTkoma\renewcommand{\cftlofttitlefont}{\size@section\sectfont}\fi
762 \fi
763 \newcommand{\cftafterlofttitle}{}
764

```

(End of definition for `\cftlofttitlefont` and `\cftafterlofttitle`. 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).

```

765 \renewcommand*{\l@figure}[2]{%
766   \ifnum \c@lofdepth >\z@
767     \vskip \cftbeforefigskip
768     {\leftskip \cftfigindent\relax
769      \rightskip \@tocrmarg
770      \parfillskip -\rightskip
771      \parindent \cftfigindent\relax\@afterindenttrue
772      \interlinepenalty\@M
773      \leavevmode
774      \@tempdima \cftfignumwidth\relax
775      \let\@cftbsnum \cftfigpresnum
776      \let\@cftasnum \cftfigaftersnum
777      \let\@cftasnumb \cftfigaftersnumb
778      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
779      \UseHookWithArguments{contentsline/text/before}{4}
780       {1}{#1}{#2}{\@contentsline@destination}%
781      {\cftfigfont #1}\nobreak
782      \UseHookWithArguments{contentsline/text/after}{4}
783       {1}{#1}{#2}{\@contentsline@destination}%
784      \cftfigfillnum{#2}}%
785   \fi
786 }

```

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

`\cftbeforefigskip` These are the user commands to control the typesetting of Figure caption entries. They
`\cftfigindent` are initialised to give the standard appearance.

```

787 \newlength{\cftbeforefigskip}
788 \setlength{\cftbeforefigskip}{\z@ \@plus.2\p@}
789 \newlength{\cftfigindent}
790 \setlength{\cftfigindent}{1.5em}

```

`\cftfignumwidth`
`\cftfigfont`
`\cftfigpresnum`
`\cftfigaftersnum`
`\cftfigaftersnumb`
`\cftfigleader`
`\cftfigdotsep`
`\cftfigpagefont`
`\cftfigafterpnum`
`\cftfigpnumfinal`
`\cftfigfillnum`

```

791 \newlength{\cftfignumwidth}
792 \setlength{\cftfignumwidth}{2.3em}
793 \newcommand{\cftfigfont}{\normalfont}
794 \newcommand{\cftfigpresnum}{}
795 \newcommand{\cftfigaftersnum}{}
796 \newcommand{\cftfigaftersnumb}{}
797 \newcommand{\cftfigleader}{\normalfont\cftdotfill{\cftfigdotsep}}
798 \newcommand{\cftfigdotsep}{\cftdotsep}
799 \newcommand{\cftfigpagefont}{\normalfont}
800 \newcommand{\cftfigafterpnum}{}
801 \newcommand{\cftfigpnumfinal}{\cftpnumfinaldefault}
802 \newcommand{\cftfigfillnum}[1]{%
803   {%
804     \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
805     \cftfigleader
806     \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
807   }\nobreak
808   \UseHookWithArguments{contentsline/page/before}{4}
809   {1}{\#1}{\@contentsline@destination}%
810   \makebox[\@pnumwidth][\cftpnumalign]{\cftfigpagefont #1}%
811   \UseHookWithArguments{contentsline/page/after}{4}
812   {1}{\#1}{\@contentsline@destination}%
813   \cftfigafterpnum\cftfigpnumfinal\par
814 }

```

(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
lotdepth here if that package is not used.

```

815 \if@cftsubfigopt\else
816 \newcounter{lofdepth}\setcounter{lofdepth}{1}
817 \newcounter{lotdepth}\setcounter{lotdepth}{1}
818 \fi
819

```

(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`.

```

820 \newcommand{\@cftdobiblot}{%
821 \if@dotoclot
822 \if@bibchapter
823 \phantomsection
824 \addcontentsline{toc}{chapter}{\cftlottitlename}
825 \else
826 \phantomsection
827 \addcontentsline{toc}{\@tocextra}{\cftlottitlename}
828 \fi
829 \fi}
830

```

(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).

```

831 \AtBeginDocument{%
832 \if@cftnctoc\else
833 \renewcommand{\listoftables}{%
834 \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.

```

835 \par
836 \begingroup
837 \parindent\z@ \parskip\cftparskip
838 \@cftmakelottitle
839 \if@cfttocbibind
840 \cftdobibliot
841 \fi
842 \@starttoc{lot}%
843 \endgroup

```

Finally, restore any multicolumn typesetting.

```

844 \@cfttocfinish}%
845 \fi
846 }
847

```

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

`\@cftmakelottitle` This command typesets the title for the LoT.

```

848 \newcommand{\@cftmakelottitle}{\cfttitlecommand{lot}}%

```

(End of definition for \@cftmakelottitle.)

`\cftbeforelottitleskip` These two lengths control the vertical spacing before and after the LoT title.

```

\cftafterlottitleskip
849 \newlength{\cftbeforelottitleskip}
850 \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*`.

```

851 \if@cfthaschapter
852 \setlength{\cftbeforelottitleskip}{50pt}
853 \setlength{\cftafterlottitleskip}{40pt}
854 \else
855 \setlength{\cftbeforelottitleskip}{3.5ex \@plus 1ex \@minus .2ex}
856 \setlength{\cftafterlottitleskip}{2.3ex \@plus .2ex}
857 \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` 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).

```

858 \if@cfthaschapter

```

```

859 \newcommand{\cftlottitlefont}{\normalfont\Huge\bfseries}
860 \if@ctkoma\renewcommand{\cftlottitlefont}{\size@chapter\sectfont}\fi
861 \else
862 \newcommand{\cftlottitlefont}{\normalfont\Large\bfseries}
863 \if@ctkoma\renewcommand{\cftlottitlefont}{\size@section\sectfont}\fi
864 \fi
865 \newcommand{\cftafterlottitle}{}
866

```

(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).

```

867 \renewcommand*{\l@table}[2]{%
868 \ifnum\c@lotdepth >\z@
869 \vskip \cftbeforetabskip
870 {\leftskip \cfttabindent\relax
871 \rightskip \@tocrmarg
872 \parfillskip -\rightskip
873 \parindent \cfttabindent\relax\@afterindenttrue
874 \interlinepenalty\@M
875 \leavevmode
876 \@tempdima \cfttabnumwidth\relax
877 \let\@cftbsnum \cfttabpresnum
878 \let\@cftasnum \cfttabaftersnum
879 \let\@cftasnumb \cfttabaftersnumb
880 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
881 \UseHookWithArguments{contentsline/text/before}{4}
882 {1}{#1}{#2}{\@contentsline@destination}%
883 {\cfttabfont #1}\nobreak
884 \UseHookWithArguments{contentsline/text/after}{4}
885 {1}{#1}{#2}{\@contentsline@destination}%
886 \cfttabfillnum{#2}}%
887 \fi
888 }

```

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

`\cftbeforetabskip` `\cfttabindent` These are the user commands to control the typesetting of Table caption entries. They are initialised to give the standard appearance.

```

\cfttabnumwidth
\cfttabfont
\cfttabpresnum
\cfttabaftersnum
\cfttabaftersnumb
\cfttableader
\cfttabdotsep
\cfttabpagefont
\cfttabafterpnum
\cfttabpnumfinal
\cfttabfillnum
889 \newlength{\cftbeforetabskip}
890 \setlength{\cftbeforetabskip}{\z@ \@plus.2\p@}
891 \newlength{\cfttabindent}
892 \setlength{\cfttabindent}{1.5em}
893 \newlength{\cfttabnumwidth}
894 \setlength{\cfttabnumwidth}{2.3em}
895 \newcommand{\cfttabfont}{\normalfont}
896 \newcommand{\cfttabpresnum}{}
897 \newcommand{\cfttabaftersnum}{}
898 \newcommand{\cfttabaftersnumb}{}
899 \newcommand{\cfttableader}{\normalfont\cftdotfill{\cfttabdotsep}}
900 \newcommand{\cfttabdotsep}{\cftdotsep}
901 \newcommand{\cfttabpagefont}{\normalfont}
902 \newcommand{\cfttabafterpnum}{}

```

```

903 \newcommand{\cfttabpnumfinal}{\cftpnumfinaldefault}
904 \newcommand{\cfttabfillnum}[1]{%
905   {%
906     \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
907     \cfttableader
908     \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
909   }\nobreak
910   \UseHookWithArguments{contentsline/page/before}{4}
911   {1}{\#1}{\@contentsline@destination}%
912   \makebox[\@pnumwidth]{\cftpnumalign}{\cfttabpagefont #1}%
913   \UseHookWithArguments{contentsline/page/after}{4}
914   {1}{\#1}{\@contentsline@destination}%
915   \cfttabafterpnum\cfttabpnumfinal\par
916 }

```

(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.

`\l@subfigure` `\l@subfigure{<title>}{<page>}` typesets the LoF entry for a subfigure caption heading. It is essentially the same as the parameterised code for `\l@figure` except that account has to be taken of `lofdepth`.

```

917 \newcommand{\@cftl@subfig}{%
918 \renewcommand*{\l@subfigure}[2]{%
919   \ifnum \toclevel@subfigure > \c@lofdepth
920   \else
921     \vskip \cftbeforesubfigskip
922     {\leftskip \cftsubfigindent\relax
923     \rightskip \@tocrmarg
924     \parfillskip -\rightskip
925     \parindent \cftsubfigindent\relax\@afterindenttrue
926     \interlinepenalty\@M
927     \leavevmode
928     \@tempdima \cftsubfignumwidth\relax
929     \let\@cftbsnum \cftsubfigpresnum
930     \let\@cftasnum \cftsubfigaftersnum
931     \let\@cftasnumb \cftsubfigaftersnumb
932     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
933     \UseHookWithArguments{contentsline/text/before}{4}
934     {\toclevel@subfigure}{\#1}{\#2}{\@contentsline@destination}%
935     {\cftsubfigfont \#1}\nobreak
936     \UseHookWithArguments{contentsline/text/after}{4}
937     {\toclevel@subfigure}{\#1}{\#2}{\@contentsline@destination}%
938     \cftsubfigfillnum{\#2}}%
939   \fi
940 }%
941 }
942

```

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

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

```

943 \newcommand{\@cftsetsubfig}{%
\cftbeforesubfigskip 944 \newlength{\cftbeforesubfigskip}
\cftsubfigindent 945 \setlength{\cftbeforesubfigskip}{\z@ \@plus.2\p@}
\cftsubfignumwidth 946 \newlength{\cftsubfigindent}
\cftsubfigfont 947 \setlength{\cftsubfigindent}{3.8em}
\cftsubfigpresnum 948 \newlength{\cftsubfignumwidth}
\cftsubfigaftersnum 949 \setlength{\cftsubfignumwidth}{2.5em}
\cftsubfigaftersnumb 950 \newcommand{\cftsubfigfont}{\normalfont}
\cftsubfigleader 951 \newcommand{\cftsubfigpresnum}{}
\cftsubfigdotsep 952 \newcommand{\cftsubfigaftersnum}{}
\cftsubfigpagefont 953 \newcommand{\cftsubfigaftersnumb}{}
\cftsubfigafterpnum 954 \newcommand{\cftsubfigleader}{\normalfont\cftdotfill{\cftsubtabdotsep}}
\cftsubfigpnumfinal 955 \newcommand{\cftsubfigdotsep}{\cftdotsep}
\cftsubfigfillnum 956 \newcommand{\cftsubfigpagefont}{\normalfont}
\toclevel@subfig 957 \newcommand{\cftsubfigafterpnum}{}
\cftsubfigfillnum 958 \newcommand{\cftsubfigpnumfinal}{\cftpnumfinaldefault}
959 \providecommand{\toclevel@subfigure}{1}
960 \newcommand{\cftsubfigfillnum}[1]{%
961 {%
962 \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
963 \cftsubfigleader
964 \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
965 }\nobreak
966 \UseHookWithArguments{contentsline/page/before}{4}
967 {\toclevel@subfigure}{\##1}{\@contentsline@destination}%
968 \makebox[\@pnumwidth][\cftpnumalign]{\cftsubfigpagefont \##1}%
969 \UseHookWithArguments{contentsline/page/after}{4}
970 {\toclevel@subfigure}{\##1}{\@contentsline@destination}%
971 \cftsubfigafterpnum\cftsubfigpnumfinal\par
972 }

```

This is the end of \@cftsetsubfig.

```

973 }
974

```

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

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

```

975 \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.
976 \renewcommand*{\l@subtable}[2]{%
977 \ifnum \toclevel@subtable > \c@lofdepth
978 \else
979 \vskip \cftbeforesubtabskip
980 {\leftskip \cftsubtabindent\relax
981 \rightskip \@tocrmarg

```

```

982 \parfillskip -\rightskip
983 \parindent \cftsubtabindent\relax\@afterindenttrue
984 \interlinepenalty\@M
985 \leavevmode
986 \@tempdima \cftsubtabnumwidth\relax
987 \let\@cftbsnum \cftsubtabpresnum
988 \let\@cftasnum \cftsubtabaftersnum
989 \let\@cftasnumb \cftsubtabaftersnumb
990 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
991 \UseHookWithArguments{contentsline/text/before}{4}
992 {\toclevel@subtable}{##1}{##2}{\@contentsline@destination}%
993 {\cftsubtabfont ##1}\nobreak
994 \UseHookWithArguments{contentsline/text/after}{4}
995 {\toclevel@subtable}{##1}{##2}{\@contentsline@destination}%
996 \cftsubtabfillnum{##2}}%
997 \fi
998 }%
999 }

```

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

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

```

1000 \newcommand{\cftsetsubtab}{%

```

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

```

\cftbeforesubtabskip 1001 \newlength{\cftbeforesubtabskip}
\cftsubtabindent 1002 \setlength{\cftbeforesubtabskip}{\z@ \@plus.2\p@}
\cftsubtabnumwidth 1003 \newlength{\cftsubtabindent}
\cftsubtabfont 1004 \setlength{\cftsubtabindent}{3.8em}
\cftsubtabpresnum 1005 \setlength{\cftsubtabnumwidth}{2.5em}
\cftsubtabaftersnum 1006 \newcommand{\cftsubtabfont}{\normalfont}
\cftsubtabaftersnumb 1007 \newcommand{\cftsubtabpresnum}{\cftsubtabfont}
\cftsubtableleader 1008 \newcommand{\cftsubtabaftersnum}{\cftsubtabfont}
\cftsubtabdotsep 1009 \newcommand{\cftsubtabaftersnumb}{\cftsubtabfont}
\cftsubtabpagefont 1010 \newcommand{\cftsubtableleader}{\normalfont\cftdotfill{\cftsubtabdotsep}}
\cftsubtabafterpnum 1011 \newcommand{\cftsubtabdotsep}{\cftdotsep}
\cftsubtabpnumfinal 1012 \newcommand{\cftsubtabpagefont}{\normalfont}
\toclevel@subtable 1013 \newcommand{\cftsubtabafterpnum}{\cftsubtabfont}
\cftsubtabfillnum 1014 \newcommand{\cftsubtabpnumfinal}{\cftpnumfinaldefault}
1015 \providecommand{\toclevel@subtable}{1}
1016 \newcommand{\cftsubtagfillnum}[1]{%
1017 {%
1018 \UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
1019 \cftsubtagleader
1020 \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
1021 }\nobreak
1022 \UseHookWithArguments{contentsline/page/before}{4}
1023 {\toclevel@subtable}{##1}{\@contentsline@destination}%
1024 \makebox[\@pnumwidth][\cftpnumalign]{\cftsubtagpagefont ##1}%
1025 \UseHookWithArguments{contentsline/page/after}{4}
1026 {\toclevel@subtable}{##1}{\@contentsline@destination}%
1027 \cftsubtagafterpnum\cftsubtagpnumfinal\par
1028 }
1029 }

```

This is the end of `\@cftsetsubtab`.

```
1030 }
1031
```

(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.

```
1032 \ifcftsubfigopt
1033   \cftsetsubfig\@cftsetsubtab
1034   \AtBeginDocument{\@cftl@subfig\@cftl@subtab}
1035 \fi
1036 %% \AtBeginDocument{\ifcftsubfigopt
1037 %%   \cftsetsubfig\@cftsetsubtab
1038 %%   \cftl@subfig\@cftl@subtab
1039 %% \fi}
1040
```

3.2 New list of..

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

```
1041 \newcommand{\newlistentry}[4][\@empty]{%
```

Check if $\langle within \rangle$ and $\langle counter \rangle$ have been defined. It is an error if $\langle within \rangle$ has not been defined, and an error if $\langle counter \rangle$ has been defined. Set the default counter values.

```
\c@X
\theX
1042 \@ifundefined{c@#2}{%      check & set the counter
1043   \ifx \@empty#1\relax
1044     \newcounter{#2}
1045   \else
1046     \@ifundefined{c@#1}{\PackageWarning{tocloft}%
1047       {#1 has no counter for use as a `within'}}
1048     \newcounter{#2}%
1049     {\newcounter{#2}[#1]%
1050     \expandafter\edef\csname the#2\endcsname{%
1051       \expandafter\noexpand\csname the#1\endcsname.\noexpand\arabic{#2}}}
1052   \fi
1053   \setcounter{#2}{0}
1054 }
1055 {\PackageError{tocloft}{#2 has been previously defined}{\@eha}}
1056
```

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

`\l@X{ $\langle title \rangle$ }{ $\langle page \rangle$ }` typesets the entry.

```
\l@X
1057 \@namedef{l@#2}##1##2{%
```

Only typeset if the `\Zdepth` is greater than $\langle level-1 \rangle$.

```
1058   \ifnum \@nameuse{c@#3depth} > #4\relax
1059     \vskip \@nameuse{cftbefore#2skip}
1060     {\leftskip \@nameuse{cft#2indent}\relax
1061     \rightskip \@tocrmarg
1062     \parfillskip -\rightskip
```

```

1063 \parindent \@nameuse{cft#2indent}\relax\@afterindenttrue
1064 \interlinepenalty\@M
1065 \leavevmode
1066 \@tempdima \@nameuse{cft#2numwidth}\relax
1067 \expandafter\let\expandafter\@cftbsnum\csname cft#2presnum\endcsname
1068 \expandafter\let\expandafter\@cftasnum\csname cft#2aftersnum\endcsname
1069 \expandafter\let\expandafter\@cftasnumb\csname cft#2aftersnumb\endcsname
1070 \advance\leftskip\@tempdima \null\nobreak\hskip -\leftskip
1071 \UseHookWithArguments{contentsline/text/before}{4}
1072 {\@nameuse{toclevel@#2}}{##1}{##2}{\@contentsline@destination}%
1073 {\@nameuse{cft#2font}}{##1}\nobreak
1074 \UseHookWithArguments{contentsline/text/after}{4}
1075 {\@nameuse{toclevel@#2}}{##1}{##2}{\@contentsline@destination}%
1076 \@nameuse{cft#2fillnum}{##2}}%
1077 \fi
1078 } % end of \l@#2
1079

```

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.

```

1080 \expandafter\newlength\csname \cftbefore#2skip\endcsname
1081 \setlength{\@nameuse{cftbefore#2skip}}{\z@ \@plus .2\p@}

```

```

\cftXindent 1082 \expandafter\newlength\csname cft#2indent\endcsname
\cftXnumwidth 1083 \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.

```

1084 \ifcase #4\relax % 0
1085 \setlength{\@nameuse{cft#2indent}}{0em}
1086 \setlength{\@nameuse{cft#2numwidth}}{1.5em}
1087 \or % 1
1088 \setlength{\@nameuse{cft#2indent}}{1.5em}
1089 \setlength{\@nameuse{cft#2numwidth}}{2.3em}
1090 \or % 2
1091 \setlength{\@nameuse{cft#2indent}}{3.8em}
1092 \setlength{\@nameuse{cft#2numwidth}}{3.2em}
1093 \or % 3
1094 \setlength{\@nameuse{cft#2indent}}{7.0em}
1095 \setlength{\@nameuse{cft#2numwidth}}{4.1em}
1096 \else % anything else
1097 \setlength{\@nameuse{cft#2indent}}{10.0em}
1098 \setlength{\@nameuse{cft#2numwidth}}{5.0em}
1099 \fi

```

And the remaining commands.

```

\cftXfont 1100 \@namedef{cft#2font}{\normalfont}
\cftXpresnum 1101 \@namedef{cft#2presnum}{ }
\cftXaftersnum 1102 \@namedef{cft#2aftersnum}{ }
\cftXaftersnumb 1103 \@namedef{cft#2aftersnumb}{ }
\cftXdotsep 1104 \@namedef{cft#2dotsep}{\cftdotsep}
\cftXleader 1105 \@namedef{cft#2leader}{\normalfont\cftdotfill{\@nameuse{cft#2dotsep}}}
\cftXpagefont 1106 \@namedef{cft#2pagefont}{\normalfont}
\cftXafterpnum
\cftXpnumfinal

```

```

1107 \namedef{cft#2afterpnum}{%
1108 \namedef{cft#2pnumfinal}{\cftpnumfinaldefault}

```

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

```

\toclevel@X 1109 \ExpandArgs{ce}\def{toclevel@#2}{\interval{#4+1}}

```

Typeset the leader and page number.

```

\cftXfillnum 1110 \namedef{cft#2fillnum}##1{%
1111   {\UseTaggingSocket{toc/leaders/before}\SuspendTagging{toc/leaders}%
1112    \@nameuse{cft#2leader}%
1113    \ResumeTagging{toc/leaders}\UseTaggingSocket{toc/leaders/after}%
1114   }\nobreak
1115   \UseHookWithArguments{contentsline/page/before}{4}
1116   {\@nameuse{toclevel@#2}}{##1}{\@contentsline@destination}
1117   \makebox[\@pnumwidth][\cftpnumalign]{\@nameuse{cft#2pagefont}##1}%
1118   \UseHookWithArguments{contentsline/page/after}{4}
1119   {\@nameuse{toclevel@#2}}{##1}{\@contentsline@destination}%
1120   \@nameuse{cft#2afterpnum}\@nameuse{cft#2pnumfinal}\par
1121 }

```

This ends the definition of `\newlistentry`.

```

1122 }

```

(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.

```

1123 \newcommand{\newlistof}[4][\@empty]{%

```

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

```

1124 \ifx \@empty#1\relax
1125   \newlistentry{#2}{#3}{0}
1126 \else
1127   \newlistentry[#1]{#2}{#3}{0}
1128 \fi

```

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

\ext@Z The file extension and listing depth.

```

\Zdepth 1129 \namedef{ext@#2}{#3}
1130 \newcounter{#3depth}
1131 \setcounter{#3depth}{1}

```

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

\cftZtitlename

```

1132 \namedef{cft#3titlename}{#4}

```

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

\cftmarkZ The heading marks for the listing.

```

1133 \if@cftkoma
1134   \namedef{cftmark#3}{%
1135     \mkboth{\UseName{cft#3titlename}}{\UseName{cft#3titlename}}}
1136   \else

```



```

1137 \namedef{cftmark#3}{%
1138 \mkboth{\MakeUppercase{\UseName{cft#3titlename}}}
1139 \MakeUppercase{\UseName{cft#3titlename}}}}
1140 \fi

```

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

\listofX Typeset the listing title and entries.

```

1141 \if@cftnctoc

```

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

```

1142 \namedef{listof#2}{%
1143 \cfttocstart
1144 \if@cfthaschapter
1145 \chapter*{\UseName{cft#3titlename}}
1146 \else
1147 \section*{\UseName{cft#3titlename}}
1148 \fi
1149 \nameuse{cftmark#3}
1150 \@starttoc{#3}%
1151 \cfttocfinish}
1152 \else

```

Otherwise use the fully parameterised definition.

```

1153 \namedef{listof#2}{%
1154 \cfttocstart
1155 \par
1156 \begingroup
1157 \parindent\z@ \parskip\cftparskip
1158 \nameuse{@cftmake#3title}
1159 \@starttoc{#3}%
1160 \endgroup
1161 \cfttocfinish}
1162 \fi
1163

```

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

\cftmakeZtitle Typeset the title.

```

1164 \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

```

1165 \expandafter\newlength\csname cftbefore#3titleskip\endcsname
1166 \expandafter\newlength\csname cftafter#3titleskip\endcsname
1167 \if@cfthaschapter
1168 \setlength{\@nameuse{cftbefore#3titleskip}}{50pt}
1169 \setlength{\@nameuse{cftafter#3titleskip}}{40pt}
1170 \if@cftkoma
1171 \namedef{cft#3titlefont}{\size@chapter\sectfont}
1172 \else
1173 \namedef{cft#3titlefont}{\normalfont\Huge\bfseries}

```

```

1174     \fi
1175   \else
1176     \setlength{\@nameuse{cftbefore#3titleskip}}{3.5ex \@plus 1ex \@minus .2ex}
1177     \setlength{\@nameuse{cftafter#3titleskip}}{2.3ex \@plus .2ex}
1178     \if@cftkoma
1179       \@namedef{cft#3titlefont}{\size@section\sectfont}
1180     \else
1181       \@namedef{cft#3titlefont}{\normalfont\Huge\bfseries}
1182     \fi
1183   \fi

```

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

`\cftafterZtitle` Something to go after the title.

```

1184     \@namedef{cftafter#3title}{}

```

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

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

```

1185     \@namedef{cft#3prehook}{}
1186     \@namedef{cft#3posthook}{}

```

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

```

1187 }

```

(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.

```

1188 \newcommand{\cftsetindents}[3]{%
1189   \def\@cftemp{#1}
1190   \ifx\@cftemp\cftchapname
1191     \@cftsetindents{chap}{#2}{#3}
1192   \else
1193     \ifx\@cftemp\cftsecname \cftsetindents{sec}{#2}{#3}
1194   \else
1195     \ifx\@cftemp\cftsubsecname \cftsetindents{subsec}{#2}{#3}
1196   \else
1197     \ifx\@cftemp\cftsubsubsecname \cftsetindents{subsubsec}{#2}{#3}
1198   \else
1199     \ifx\@cftemp\cftparaname \cftsetindents{para}{#2}{#3}
1200   \else
1201     \ifx\@cftemp\cftsubparaname \cftsetindents{subpara}{#2}{#3}
1202   \else
1203     \ifx\@cftemp\cftfigname \cftsetindents{fig}{#2}{#3}
1204   \else
1205     \ifx\@cftemp\cftsubfigname \cftsetindents{subfig}{#2}{#3}
1206   \else
1207     \ifx\@cftemp\cfttabname \cftsetindents{tab}{#2}{#3}
1208   \else
1209     \ifx\@cftemp\cftsubtabname \cftsetindents{subtab}{#2}{#3}
1210   \else
1211     \@cftsetindents{#1}{#2}{#3}

```

```

1212             \fi
1213         \fi
1214     \fi
1215 \fi
1216 \fi
1217 \fi
1218 \fi
1219 \fi
1220 \fi
1221 \fi
1222 }
1223

```

(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.

```

1224 \newcommand{\@cftsetindents}[3]{%
1225     \setlength{\@nameuse{cft#1indent}}{#2}
1226     \setlength{\@nameuse{cft#1numwidth}}{#3}
1227 }
1228

```

(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.

```

1229 \newcommand{\@cftpnumoff}[1]{%
1230     \@namedef{cft#1fillnum}##1{\UseTaggingSocket{struct/end}%
1231         \cftpnumfillskip\@nameuse{cft#1afterpnum}\@nameuse{cft#1pnumfinal}\par}}
1232

```

(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
\cftsubsubsecname
\cftparaname
\cftsubparaname
\cftfigname
\cftsubfigname
\cfttabname
\cftsubtabname
1233 \newcommand*{\cftchapname}{chapter}
1234 \newcommand*{\cftsecname}{section}
1235 \newcommand*{\cftsubsecname}{subsection}
1236 \newcommand*{\cftsubsubsecname}{subsubsection}
1237 \newcommand*{\cftparaname}{paragraph}
1238 \newcommand*{\cftsubparaname}{subparagraph}
1239 \newcommand*{\cftfigname}{figure}
1240 \newcommand*{\cftsubfigname}{subfigure}
1241 \newcommand*{\cfttabname}{table}
1242 \newcommand*{\cftsubtabname}{subtable}
1243

```

(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.

```

1244 \DeclareRobustCommand{\cftpagenumbersoff}[1]{%
1245   \def\@cfttemp{#1}
1246   \ifx\@cfttemp\cftchapname
1247     \@cftpnumoff{chap}
1248   \else
1249     \ifx\@cfttemp\cftsecname \@cftpnumoff{sec}
1250   \else
1251     \ifx\@cfttemp\cftsubsecname \@cftpnumoff{subsec}
1252   \else
1253     \ifx\@cfttemp\cftsubsubsecname \@cftpnumoff{subsubsec}
1254   \else
1255     \ifx\@cfttemp\cftparaname \@cftpnumoff{para}
1256   \else
1257     \ifx\@cfttemp\cftsubparaname \@cftpnumoff{subpara}
1258   \else
1259     \ifx\@cfttemp\cftfigname \@cftpnumoff{fig}
1260   \else
1261     \ifx\@cfttemp\cftsubfigname \@cftpnumoff{subfig}
1262   \else
1263     \ifx\@cfttemp\cfttabname \@cftpnumoff{tab}
1264   \else
1265     \ifx\@cfttemp\cftsubtabname \@cftpnumoff{subtab}
1266   \else
1267     \@cftpnumoff{#1}
1268   \fi
1269   \fi
1270   \fi
1271   \fi
1272   \fi
1273   \fi
1274   \fi
1275   \fi
1276   \fi
1277   \fi
1278 }
1279

```

(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`.

```

1280 \DeclareRobustCommand{\cftpagenumberon}[1]{%
1281   \def\@cfttemp{#1}
1282   \ifx\@cfttemp\cftchapname
1283     \@cftpnumon{chap}
1284   \else
1285     \ifx\@cfttemp\cftsecname \@cftpnumon{sec}
1286   \else
1287     \ifx\@cfttemp\cftsubsecname \@cftpnumon{subsec}

```

```

1288 \else
1289 \ifx\@cfttemp\cftsubsubsecname \@cftpnumon{subsubsec}
1290 \else
1291 \ifx\@cfttemp\cftparaname \@cftpnumon{para}
1292 \else
1293 \ifx\@cfttemp\cftsubparaname \@cftpnumon{subpara}
1294 \else
1295 \ifx\@cfttemp\cftfigname \@cftpnumon{fig}
1296 \else
1297 \ifx\@cfttemp\cftsubfigname \@cftpnumon{subfig}
1298 \else
1299 \ifx\@cfttemp\cfttabname \@cftpnumon{tab}
1300 \else
1301 \ifx\@cfttemp\cftsubtabname \@cftpnumon{subtab}
1302 \else
1303 \@cftpnumon{#1}
1304 \fi
1305 \fi
1306 \fi
1307 \fi
1308 \fi
1309 \fi
1310 \fi
1311 \fi
1312 \fi
1313 \fi
1314 }
1315

```

(End of definition for \cftpagenumberon. 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.

```

1316 \newcommand{\@cftpnumon}[1]{%
1317 \namedef{cft#1fillnum}##1{%
1318 {\@nameuse{cft#1leader}}\nobreak
1319 \makebox[\@pnumwidth][\cftpnumalign]{\@nameuse{cft#1pagefont}##1}%
1320 \@nameuse{cft#1afterpnum}\@nameuse{cft#1pnumfinal}\par
1321 }%
1322 }

```

(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 is expected to be called immediately after a \chapter command.

```

1323 \newcommand{\cftchapterprecis}[1]{%
1324 \cftchapterprecishere{#1}
1325 \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.

```
1326 \newcommand{\cftchapterprecishere}[1]{%
1327   \par\vspace*{-2\baselineskip}
```

Typeset its argument using italic font in a quote environment.

```
1328   \begin{quote}\textit{#1}\end{quote}
1329   \par\nobreak\@afterheading}
```

(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.

```
1330 \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.

```
1331   {\UseTaggingSocket{struct/begin}{tag=\protect\UseStructureName{toc/item}}}%
1332   \UseTaggingSocket{struct-mc}{tag=\protect\UseStructureName{toc/item/text}}}%
1333   \leftskip \cftchapindent\relax
1334   \advance\leftskip \cftchapnumwidth\relax
```

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

```
1335   \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.

```
1336   \textit{#1}\protect\par}%
1337   \UseTaggingSocket{struct/end}}}%
1338
```

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

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

```
1339 \newcommand{\cftlocalchange}[3]{%
1340   \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.

```
1341 \newcommand{\cftaddtitleline}[4]{\addtocontents{#1}{%
1342   \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.

```
1343 \newcommand{\cftaddnumtitleline}[5]{\addtocontents{#1}{%
1344   \protect\contentsline{#2}{\protect\numberline{#3}{#4}{#5}}{\@currentHref}%
1345   \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`, `scrreprt`
- `scrwtf`
- `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.

```

1346 \def\starttoc@latex@orig#1{%
1347 \begingroup
1348 \makeatletter
1349 \@input{\jobname.#1}%
1350 \if@files
1351 \expandafter\newwrite\csname tf@#1\endcsname
1352 \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
1353 \fi
1354 \@nobreakfalse
1355 \endgroup}
1356 \newif\if@cft@patch@starttoc
1357 \IfDocumentMetadataTF
1358 {
1359 \cft@patch@starttoctrue
1360 }
1361 {
1362 \ifx\starttoc\starttoc@latex@orig
1363 \cft@patch@starttoctrue
1364 \fi
1365 }

1366 \if@cft@patch@starttoc
1367 \patchcmd\starttoc
1368 {\begingroup}
1369 {\csname cft#1prehook\endcsname\begingroup}
1370 {}
1371 {%
1372 \PackageWarning{tocloft}
1373 {Patching of \string\starttoc\space failed.\MessageBreak Prehook not inserted.}
1374 }
1375 \patchcmd\starttoc
1376 {\endgroup}
1377 {\endgroup\csname cft#1posthook\endcsname}
1378 {}
1379 {%
1380 \PackageWarning{tocloft}
1381 {Patching of \string\starttoc\space failed.\MessageBreak Posthook not inserted.}
1382 }
1383 \newcommand\cfttocprehook{}
1384 \newcommand\cftlofprehook{}
1385 \newcommand\cftlotprehook{}
1386 \newcommand\cfttocposthook{}
1387 \newcommand\cftlofposthook{}
1388 \newcommand\cftlotposthook{}
1389 \else
1390 \PackageWarning{tocloft}{\string\starttoc\space has already been redefined;
1391 \string\cfttocprehook\space et al. not installed.}
1392 \fi

```


(End of definition for \@starttoc.)

The end of this package.

1393 `</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		
<code>\@cftasnum</code>	17	<code>\addtocontents</code> 5
<code>\@cftasnumb</code>	17	<code>\addtocontents</code> 1330, 1340, 1341, 1343
<code>\@cftbsnum</code>	17	<code>\AtBeginDocument</code>
<code>\@dotsep</code>	4	 18, 30, 224, 729, 831, 1034, 1036
<code>\@dottedtocline</code>	5	
<code>\@pnumwidth</code>	4	
<code>\@tocrmarg</code>	4	
		B
		<code>\baselineskip</code> 1327
		C
A		<code>\cftaddnumtitleline</code> 16
<code>\addcontentsline</code>	3	<code>\cftaddnumtitleline</code> 1343
<code>\addcontentsline</code> 84, 87, 723, 726, 824, 827		<code>\cftaddtitleline</code> 16

\cftaddtitleline	1341	\cftdotsep	9
\cftafterlofttitle	756	\cftdotsep	280, 481, 542,
\cftafterlofttitleskip	747		594, 648, 702, 798, 900, 955, 1012, 1104
\cftafterlotttitle	858	\cftfigafterpnum	787
\cftafterlottitleskip	849	\cftfigaftersnum	776, 787
\cftaftertoctitle	259	\cftfigaftersnumb	777, 787
\cftaftertoctitleskip	250	\cftfigdotsep	787
\cftafterZtitle	8	\cftfigfillnum	784, 787
\cftafterZtitle	3, 1184	\cftfigfont	781, 787
\cftafterZtitleskip	7	\cftfigindent	768, 771, 787
\cftafterZtitleskip	1165	\cftfigleader	787
\cftbeforechapskip	390, 410	\cftfigname	1203, 1233, 1259, 1295
\cftbeforefigskip	767, 787	\cftfignumwidth	774, 787
\cftbeforelofttitleskip	747	\cftfigpagefont	787
\cftbeforelottitleskip	849	\cftfigpnumfinal	787
\cftbeforeparaskip	613, 632	\cftfigpresnum	775, 787
\cftbeforepartskip	317, 350	\cftlocalchange	16
\cftbeforesecskip	446, 449, 469	\cftlocalchange	1339
\cftbeforesubfigskip	921, 944	\cftlofposthook	1387
\cftbeforesubparaskip	667, 686	\cftlofprehook	1384
\cftbeforesubsecskip	509, 526	\cftlofttitlefont	756
\cftbeforesubsubsecskip	561, 578	\cftlofttitlename	55, 62, 96, 723, 726
\cftbeforesubtabskip	979, 1001	\cftlotposthook	1388
\cftbeforetabskip	869, 889	\cftlotprehook	1385
\cftbeforetoctitleskip	250	\cftlottitlefont	858
\cftbeforeXskip	9	\cftlottitlename	57, 64, 97, 824, 827
\cftbeforeXskip	1080	\cftmarklof	52
\cftbeforeZtitleskip	7	\cftmarklot	52
\cftbeforeZtitleskip	1165	\cftmarktoc	52
\cftchapafterpnum	410	\cftmarkZ	7
\cftchapaftersnum	399, 410	\cftmarkZ	1133
\cftchapaftersnumb	400, 410	\cftnodots	9
\cftchapdotsep	410	\cftnodots	273, 280, 361, 422, 489
\cftchapfillnum	407, 410	\cftpagenumbersoff	12
\cftchapfont	404, 410	\cftpagenumbersoff	1244
\cftchapindent	391, 394, 410, 1333	\cftpagenumberon	12
\cftchapleader	410	\cftpagenumberon	1280
\cftchapname	1190, 1233, 1246, 1282	\cftparaafterpnum	632
\cftchapnumwidth	397, 410, 1334	\cftparaaftersnum	622, 632
\cftchappagefont	410	\cftparaaftersnumb	623, 632
\cftchappnumfinal	410	\cftparadotsep	632
\cftchappresnum	398, 410	\cftparafillnum	630, 632
\cftchapter	8	\cftparafont	627, 632
\cftchapter	99, 103	\cftparaindent	614, 617, 632
\cftchapterprecis	16	\cftparaleader	632
\cftchapterprecis	1323	\cftparaname	1199, 1233, 1255, 1291
\cftchapterprecishere	16	\cftparanumwidth	620, 632
\cftchapterprecishere	1324, 1326	\cftparapagefont	632
\cftchapterprecistoc	16	\cftparapnumfinal	632
\cftchapterprecistoc	1325, 1330	\cftparapresnum	621, 632
\cftdot	8	\cftparfillskip	12
\cftdot	270	\cftparfillskip	282, 1231
\cftdotfill	270, 360, 421, 480, 488, 541,	\cftparskip	9
	593, 647, 701, 797, 899, 954, 1011, 1105	\cftparskip	91, 226, 230, 239, 735, 837, 1157

\cftpartafterpnum	350	\cftsubfigpnumfinal	944
\cftpartaftersnum	328, 350	\cftsubfigpresnum	929, 944
\cftpartaftersnumb	329, 350	\cftsubparaafterpnum	686
\cftpardotsep	350	\cftsubparaaftersnum	676, 686
\cftpardfillnum	336, 350	\cftsubparaaftersnumb	677, 686
\cftpardfont	333, 350	\cftsubparadotsep	686
\cftpardindent	319, 322, 350	\cftsubparafillnum	684, 686
\cftpardinitnumwidth	10	\cftsubparafont	681, 686
\cftpardinitnumwidth	326, 356	\cftsubparaindent	668, 671, 686
\cftpardleader	350	\cftsubparaleader	686
\cftpardnumwidth	325, 350	\cftsubparaname	1201, 1233, 1257, 1293
\cftpardpagefont	350	\cftsubparanumwidth	674, 686
\cftpardpnumfinal	350	\cftsubparapagefont	686
\cftpardpresnum	327, 333, 350	\cftsubparapnumfinal	686
\cftpnumalign	9	\cftsubparapresnum	675, 686
\cftpnumalign	9, 269, 375, 433, 502, 554, 606, 660, 714, 810, 912, 968, 1025, 1117, 1319	\cftsubsecafterpnum	526, 557
\cftpnumfinaldefault	11	\cftsubsecaftersnum	518, 526
\cftpnumfinaldefault	94, 364, 425, 493, 545, 597, 651, 705, 801, 903, 958, 1015, 1108	\cftsubsecaftersnumb	519, 526
\cftsecafterpnum	469	\cftsubsecdotsep	526
\cftsecaftersnum	459, 469	\cftsubsecfillnum	524, 546
\cftsecaftersnumb	460, 469	\cftsubsecfont	522, 526
\cftsecdotsep	469	\cftsubsecindent	510, 513, 526
\cftsecfillnum	467, 469	\cftsubsecleader	526, 549
\cftsecfont	464, 469	\cftsubsecname	1195, 1233, 1251, 1287
\cftsecindent	451, 454, 469	\cftsubsecnumwidth	516, 526
\cftsecleader	469	\cftsubsecpagefont	526, 554
\cftsecname	1193, 1233, 1249, 1285	\cftsubsecpnumfinal	526, 557
\cftsecnumwidth	457, 469	\cftsubsecpresnum	517, 526
\cftsecpagefont	469	\cftsubsubsecafterpnum	578
\cftsecpnumfinal	469	\cftsubsubsecaftersnum	570, 578
\cftsecpresnum	458, 469	\cftsubsubsecaftersnumb	571, 578
\cftsection	8	\cftsubsubsecdotsep	578
\cftsection	101, 103	\cftsubsubsecfillnum	576, 578
\cftsetindents	11	\cftsubsubsecfont	574, 578
\cftsetindents	1188	\cftsubsubsecindent	562, 565, 578
\cftsetpnumwidth	9	\cftsubsubsecleader	578
\cftsetpnumwidth	267, 1340	\cftsubsubsecname	1197, 1233, 1253, 1289
\cftsetrmarg	9	\cftsubsubsecnumwidth	568, 578
\cftsetrmarg	267, 1340	\cftsubsubsecpagefont	578
\cftsubfigafterpnum	944	\cftsubsubsecpnumfinal	578
\cftsubfigaftersnum	930, 944	\cftsubsubsecpresnum	569, 578
\cftsubfigaftersnumb	931, 944	\cftsubtabafterpnum	1001
\cftsubfigdotsep	944	\cftsubtabaftersnum	988, 1001
\cftsubfigfillnum	938, 944	\cftsubtabaftersnumb	989, 1001
\cftsubfigfont	935, 944	\cftsubtabdotsep	954, 1001
\cftsubfigindent	922, 925, 944	\cftsubtabfillnum	996, 1001
\cftsubfigleader	944	\cftsubtabfont	993, 1001
\cftsubfigname	1205, 1233, 1261, 1297	\cftsubtabindent	980, 983, 1001
\cftsubfignumwidth	928, 944	\cftsubtableader	1001
\cftsubfigpagefont	944	\cftsubtabname	1209, 1233, 1265, 1301
		\cftsubtabnumwidth	986, 1001
		\cftsubtabpagefont	1001
		\cftsubtabpnumfinal	1001
		\cftsubtabpresnum	987, 1001

\cftsubtagafterpnum	1028	\cftZtitlename	8
\cftsubtagfillnum	1017	\cftZtitlename	95, 1132
\cftsubtagleader	1020	\chapter	117, 160, 1145
\cftsubtagpagefont	1025	\contentsline	3
\cftsubtagpnumfinal	1028	\contentsline	1342, 1344
\cfttabafterpnum	889	\contentsname	95
\cfttabaftersnum	878, 889	\csname	5, 1050, 1051,
\cfttabaftersnumb	879, 889		1067, 1068, 1069, 1080, 1082, 1083,
\cfttabdotsep	889		1165, 1166, 1351, 1352, 1369, 1377
\cfttabfillnum	886, 889	D	
\cfttabfont	883, 889	\DeclareDocumentCommand	
\cfttabindent	870, 873, 889		114, 135, 157, 178, 197, 210
\cfttableader	889	\DeclareInstance	108, 130, 152, 174
\cfttabname	1207, 1233, 1263, 1299	\DeclareRobustCommand	1244, 1280
\cfttabnumwidth	876, 889	\dimeval	122, 142, 165, 185
\cfttabpagefont	889	\DocumentMetadata	2, 3, 8, 21
\cfttabpnumfinal	889	E	
\cfttabpresnum	877, 889	\edef	1050
\cfttitlecommand	8	\endcsname	5, 1050, 1051,
\cfttitlecommand	98, 249, 746, 848, 1164		1067, 1068, 1069, 1080, 1082, 1083,
\cfttocposthook	3, 1386		1165, 1166, 1351, 1352, 1369, 1377
\cfttocprehook	3, 1383, 1391	\endinput	24
\cfttoctitlefont	259	\ExpandArgs	1109
\cfttoctitlename	53, 60, 84, 87, 95	H	
\cftXafterpnum	10	\Huge	260, 757, 859, 1173, 1181
\cftXafterpnum	1100	I	
\cftXaftersnum	10	\IfDocumentMetadataTF	104, 1357
\cftXaftersnum	1100	\immediate	1352
\cftXaftersnumb	10	\interlinepenalty	
\cftXaftersnumb	1100		202, 215, 323, 395, 455, 514,
\cftXdotsep	10		566, 618, 672, 772, 874, 926, 984, 1064
\cftXdotsep	1100	\interval	1109
\cftXfillnum	10	L	
\cftXfillnum	1110	\l@X	14
\cftXfont	10	\Large	263, 760, 862
\cftXfont	1100	\large	355, 360, 362, 382, 383
\cftXindent	9	\leaders	277
\cftXindent	1082	\leftskip	319, 330, 391, 401, 451,
\cftXleader	10		461, 510, 520, 562, 572, 614, 624,
\cftXleader	1100		668, 678, 768, 778, 870, 880, 922,
\cftXnumwidth	9		932, 980, 990, 1060, 1070, 1333, 1334
\cftXnumwidth	1082	\listfigurename	96
\cftXpagefont	10	\listoffigures	7
\cftXpagefont	1100	\listoffigures	729
\cftXpnumfinal	10	\listoftables	7
\cftXpnumfinal	1100	\listoftables	831
\cftXpresnum	10	\listofX	14
\cftXpresnum	1100	\listofX	1141
\cftZposthook	17	\listtablename	97
\cftZposthook	1185	\lofdepth	815
\cftZprehook	17	\lotdepth	815
\cftZprehook	17, 1185		
\cftZtitlefont	8		
\cftZtitlefont	1165		

M	
<code>\makeatletter</code>	1348
<code>\makebox</code>	9, 25, 375, 433, 502, 554, 606, 660, 714, 810, 912, 968, 1025, 1117, 1319
<code>\MakeUppercase</code>	53, 55, 57, 1138, 1139
<code>\MessageBreak</code>	35, 36, 37, 1373, 1381
<code>\mkern</code>	277
N	
<code>\newif</code> 11, 12, 17, 29, 41, 44, 295, 296, 1356	
<code>\newlistentry</code>	14
<code>\newlistentry</code>	1041, 1125, 1127
<code>\newlistof</code>	13
<code>\newlistof</code>	1123
<code>\newwrite</code>	1351
<code>\nobreakspace</code>	155
<code>\noexpand</code>	1051
<code>\normalfont</code>	260, 263, 479, 480, 482, 537, 541, 543, 589, 593, 595, 643, 647, 649, 697, 701, 703, 757, 760, 793, 797, 799, 859, 862, 895, 899, 901, 950, 954, 956, 1007, 1011, 1013, 1100, 1105, 1106, 1173, 1181
<code>\numberline</code>	4
<code>\numberline</code>	283
O	
<code>\onecolumn</code>	69
<code>\openout</code>	1352
P	
<code>\PackageError</code>	1055
<code>\PackageInfo</code>	25, 27
<code>\PackageWarning</code>	22, 34, 1046, 1372, 1380, 1390
<code>\par</code>	3
<code>\parindent</code>	239, 322, 394, 454, 513, 565, 617, 671, 735, 771, 837, 873, 925, 983, 1063, 1157
<code>\parskip</code>	230, 239, 735, 837, 1157
<code>\patchcmd</code>	1367, 1375
<code>\phantomsection</code> 78, 83, 86, 722, 725, 823, 826	
<code>\providecommand</code>	66, 74, 78, 270, 271, 280, 959, 1016
R	
<code>\RequirePackage</code>	2
<code>\ResumeTagging</code>	371, 429, 498, 550, 602, 656, 710, 806, 908, 964, 1021, 1113
S	
<code>\sectfont</code>	261, 264, 382, 383, 439, 758, 761, 860, 863, 1171, 1179
<code>\section</code>	138, 181, 1147
<code>\starttoc</code>	56
<code>\string</code>	1373, 1381, 1390, 1391
<code>\SuspendTagging</code> ...	369, 427, 496, 548, 600, 654, 708, 804, 906, 962, 1019, 1111
T	
<code>\tableofcontents</code>	7
<code>\tableofcontents</code>	224
<code>\tagpdfparaOff</code>	145
<code>\tagpdfparaOn</code>	145
T _E X and L ^A T _E X 2 _ε commands:	
<code>\@afterheading</code>	124, 144, 167, 187, 208, 221, 1329
<code>\@afterindenttrue</code> 322, 394, 454, 513, 565, 617, 671, 771, 873, 925, 983, 1063	
<code>\@cft@patch@starttoctrue</code> .	1359, 1363
<code>\@cftasnum</code>	288, 292, 328, 399, 459, 518, 570, 622, 676, 776, 878, 930, 988, 1068
<code>\@cftasnumb</code>	290, 292, 329, 400, 460, 519, 571, 623, 677, 777, 879, 931, 989, 1069
<code>\@cftbsnum</code>	288, 292, 327, 398, 458, 517, 569, 621, 675, 775, 877, 929, 987, 1067
<code>\@cftdobiblof</code>	719, 738
<code>\@cftdobiblot</code>	820, 840
<code>\@cftdobibtoc</code>	80, 242
<code>\@cftdopartfalse</code>	300
<code>\@cftdoparttrue</code>	303, 307
<code>\@cfttemp</code>	1189, 1190, 1193, 1195, 1197, 1199, 1201, 1203, 1205, 1207, 1209, 1245, 1246, 1249, 1251, 1253, 1255, 1257, 1259, 1261, 1263, 1265, 1281, 1282, 1285, 1287, 1289, 1291, 1293, 1295, 1297, 1299, 1301
<code>\@cfthaschapterfalse</code>	20
<code>\@cfthaschaptertrue</code>	26
<code>\@cfthaspartfalse</code>	297
<code>\@cfthasparttrue</code>	297
<code>\@cftifundefined</code>	3, 19, 21, 297
<code>\@cftinitnumwidth</code>	283, 326
<code>\@cftkomafalse</code>	13
<code>\@cftkomatrue</code>	14, 15, 16
<code>\@cftl@subfig</code>	917, 1034, 1038
<code>\@cftl@subtab</code>	975, 1034, 1038
<code>\@cftmakeZtitle</code>	1164
<code>\@cftmakeloftitle</code>	736, 746
<code>\@cftmakelottitle</code>	838, 848
<code>\@cftmaketocitle</code>	240, 249
<code>\@cftnctofalse</code>	41
<code>\@cftnctoctrue</code>	42
<code>\@cftpagestyle</code>	48, 124, 144, 167, 187, 201, 214

<code>\@cftpnumoff</code>	71
<u>1229</u> , 1247, 1249, 1251, 1253, 1255,	
1257, 1259, 1261, 1263, 1265, 1267	
<code>\@cftpnumon</code>	69
1283, 1285, 1287, 1289, 1291, 1293,	
1295, 1297, 1299, 1301, 1303, <u>1316</u>	
<code>\@cftquit</code>	8
<u>10</u> , 24, 28	
<code>\@cftsetindents</code>	199, 212, 315, 448
1191, 1193, 1195, 1197, 1199, 1201,	
1203, 1205, 1207, 1209, 1211, <u>1224</u>	
<code>\@cftsetsubfig</code>	3, 55, 56, 227,
<u>943</u> , 1033, 1037	228, 244, 740, 842, 1150, 1159, <u>1346</u>
<code>\@cftsetsubtab</code>	<u>1346</u> , 1362
<u>1000</u> , 1033, 1037	
<code>\@cftsubfigoptfalse</code>	<u>272</u> , 274
44	
<code>\@cftsubfigopttrue</code>	<u>273</u> , 274
45	
<code>\@cfttitlesectrue</code>	
18	
<code>\@cfttocbibindfalse</code>	87, 726, 827
31	
<code>\@cfttocbibindtrue</code>	
31	
<code>\@cfttocfinish</code>	
..... <u>66</u> , 246, 742, 844, 1151, 1161	
<code>\@cfttocstart</code>	
..... <u>66</u> , 236, 732, 834, 1143, 1154	
<code>\@contentsline@destination</code>	
..... 332, 335, 374, 377, 403, 406,	
432, 435, 463, 466, 501, 504, 521,	
523, 553, 556, 573, 575, 605, 608,	
626, 629, 659, 662, 680, 683, 713,	
716, 780, 783, 809, 812, 882, 885,	
911, 914, 934, 937, 967, 970, 992,	
995, 1024, 1027, 1072, 1075, 1116, 1119	
<code>\@currentHref</code>	
1342, 1344	
<code>\@eha</code>	
1055	
<code>\@firstoftwo</code>	
6	
<code>\@highpenalty</code>	
313, 389	
<code>\@ifclassloaded</code>	
14, 15, 16	
<code>\@ifpackagelater</code>	
33	
<code>\@ifpackageloaded</code>	
18, 31	
<code>\@ifundefined</code>	
106, 1042, 1046	
<code>\@namedef</code>	
1057, 1100, 1101,	
1102, 1103, 1104, 1105, 1106, 1107,	
1108, 1110, 1129, 1132, 1134, 1137,	
1142, 1153, 1164, 1171, 1173, 1179,	
1181, 1184, 1185, 1186, 1230, 1317	
<code>\@nameuse</code>	
1058, 1059,	
1060, 1063, 1066, 1072, 1073, 1075,	
1076, 1081, 1085, 1086, 1088, 1089,	
1091, 1092, 1094, 1095, 1097, 1098,	
1105, 1112, 1116, 1117, 1119, 1120,	
1149, 1158, 1168, 1169, 1176, 1177,	
1225, 1226, 1231, 1318, 1319, 1320	
<code>\@nobreakfalse</code>	
340, 344, 1354	
<code>\@nobreaktrue</code>	
339, 343	
<code>\@pnumwidth</code>	
. 267, 375, 433, 502, 554, 606, 660,	
714, 810, 912, 968, 1025, 1117, 1319	
<code>\@restonecolfalse</code>	
<code>\@restonecoltrue</code>	
<code>\@secondoftwo</code>	
<code>\@secpenalty</code>	
<code>\@starttoc</code>	
3, 55, 56, 227,	
228, 244, 740, 842, 1150, 1159, <u>1346</u>	
<code>\@starttoc@latex@orig</code>	
1346, 1362	
<code>\@tempa</code>	
272, 274	
<code>\@tempb</code>	
273, 274	
<code>\@tocextra</code>	
87, 726, 827	
<code>\@tocrmarg</code>	
..... 268, 320, 392, 452, 511, 563,	
615, 669, 769, 871, 923, 981, 1061, 1335	
<code>\c@lofdepth</code>	
766, 919, 977	
<code>\c@lotdepth</code>	
868	
<code>\c@tocdepth</code>	
301, 305, 388, 444, 508, 560, 612, 666	
<code>\c@X</code>	
<u>1042</u>	
<code>\cft@listname</code>	
<u>103</u>	
<code>\ext@Z</code>	
<u>1129</u>	
<code>\hb@xt@</code>	
286	
<code>\if@bibchapter</code>	
82, 721, 822	
<code>\if@cft@patch@starttoc</code> ...	
1356, 1366	
<code>\if@cftdopart</code>	
<u>295</u>	
<code>\if@cfthaschapter</code>	
<u>10</u> ,	
67, 75, 98, 252, 259, 302, 306, 312,	
338, 386, 410, 445, 475, 530, 582,	
636, 690, 749, 756, 851, 858, 1144, 1167	
<code>\if@cfthaspart</code>	
296, 298, 350	
<code>\if@cftkoma</code>	
.. <u>12</u> , 58, 261, 264, 333, 380, 438,	
758, 761, 860, 863, 1133, 1170, 1178	
<code>\if@cftnctoc</code> ...	
<u>41</u> , 225, 730, 832, 1141	
<code>\if@cftsubfigopt</code> .	
<u>44</u> , 815, 1032, 1036	
<code>\if@cfttitlesec</code>	
<u>17</u> , 333	
<code>\if@cfttocbibind</code> ...	
<u>29</u> , 241, 737, 839	
<code>\if@compatibility</code>	
342	
<code>\if@dotoclof</code>	
720	
<code>\if@dotoclot</code>	
821	
<code>\if@dotocloc</code>	
81	
<code>\if@filesw</code>	
1350	
<code>\if@restonecol</code>	
76	
<code>\if@twocolumn</code>	
68	
<code>\l@</code>	
1078	
<code>\l@chapter</code>	
<u>386</u>	
<code>\l@figure</code>	
<u>765</u>	
<code>\l@paragraph</code>	
<u>611</u>	
<code>\l@part</code>	
<u>295</u>	
<code>\l@section</code>	
<u>443</u>	
<code>\l@subfigure</code>	
<u>918</u>	
<code>\l@subparagraph</code>	
<u>665</u>	
<code>\l@subsection</code>	
<u>507</u>	
<code>\l@subsubsection</code>	
<u>559</u>	
<code>\l@subtable</code>	
<u>976</u>	

<code>\l@table</code>	867	523, 552, 555, 573, 575, 604, 607,
<code>\l@X</code>	1057	625, 628, 658, 661, 679, 682, 712,
<code>\OLD@starttoc</code>	227, 231	715, 779, 782, 808, 811, 881, 884,
<code>\protected@file@percent</code> ..	1342, 1345	910, 913, 933, 936, 966, 969, 991,
<code>\size@chapter</code>	261, 758, 860, 1171	994, 1023, 1026, 1071, 1074, 1115, 1118
<code>\size@section</code>	264, 761, 863, 1179	<code>\UseName</code> 112, 122, 123, 125, 128, 133, 142,
<code>\toclevel@subfig</code>	944	143, 145, 148, 165, 166, 168, 170,
<code>\toclevel@subfigure</code>		172, 185, 186, 188, 190, 192, 200,
.....	919, 934, 937, 959, 967, 970	203, 204, 205, 207, 213, 216, 217,
<code>\toclevel@subtable</code>	977, 992, 995, 1001	218, 220, 1135, 1138, 1139, 1145, 1147
<code>\toclevel@X</code>	1109	<code>\UseStructureName</code>
<code>\textit</code>	1328, 1336	1331, 1332
<code>\theX</code>	14	<code>\UseTaggingSocket</code>
<code>\theX</code>	1042	369,
<code>\thispagestyle</code>	49	371, 427, 429, 496, 498, 548, 550,
<code>\tocloftpagestyle</code>	7	600, 602, 654, 656, 708, 710, 804,
<code>\tocloftpagestyle</code>	48	806, 906, 908, 962, 964, 1019, 1021,
<code>\topskip</code>	122, 142, 165, 185	1111, 1113, 1230, 1331, 1332, 1337
<code>\twocolumn</code>	76	
		W
		<code>\WordSpaceAmount</code>
		111
	U	
<code>\UseHookWithArguments</code>	287,	
	289, 331, 334, 373, 376, 402, 405,	
	431, 434, 462, 465, 500, 503, 521,	
		Z
		<code>\Zdepth</code>
		8
		<code>\Zdepth</code>
		1129
		<code>\ZZdepth</code>
		14