

The `tocloft` package*

Author: Peter Wilson, Herries Press

Maintainer: Will Robertson

will dot robertson at latex-project dot org

2017/08/31

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	<code>L</code> <code>A</code> <code>T</code> <code>E</code> <code>X</code> ’s methods	2
2	The <code>tocloft</code> package	6
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	18
3	The package code	19
3.1	Support for the <code>subfigure</code> package	39
3.2	New list of...	42
3.3	Switching page numbering	47
3.4	Experimental utilities	49

List of Figures

1	Layout of a ToC (LoF, LoT) entry	3
---	--	---

*This file (`tocloft.dtx`) has version number v2.3i, last revised 2017/08/31.

List of Tables

1	Indents and Numwidths	4
---	-----------------------	---

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 \LaTeX DOCSTRIP utility which enables the automatic extraction of the \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

1.1 \LaTeX 's methods

This is a general description of how \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` \LaTeX generates a `.toc` file if the document contains a `\tableofcontents` command. The sectioning commands² put entries into the `.toc` file by calling the \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 `\contentsline{<kind>}{<title>}{<page>}` where `<page>` is the page number. For each `<kind>`, \LaTeX provides a command `\l@kind{<title>}{<page>}` which performs the actual typesetting of the `\contentsline` entry.

`\@pnumwidth` The general layout of a typeset entry is illustrated in Figure 1. There are three

`\@tocmarg`
`\@dotsep`
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

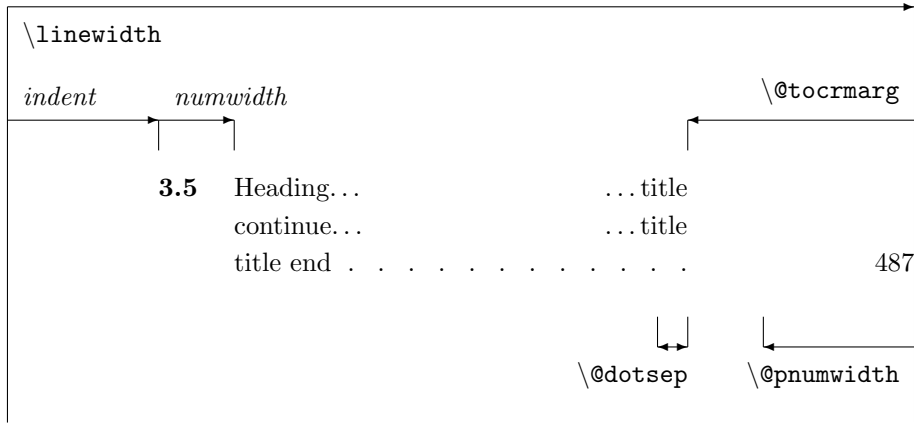


Figure 1: Layout of a ToC (LoF, LoT) entry

internal L^AT_EX 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 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).

`\numberline` 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 $(\linewidth - indent - numwidth - \@tocrmarg)$.

(athurber@emba.uvm.edu) for their suggestions.
²For figures and tables it is the `\caption` command that populates the `.lof` and `.lot` files.
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{<selevel>}{<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 \protects 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\negspace{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 `\negspace` 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
\listoffigures
\listoftables
```

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.
- titles** The `titles` option causes the titles of the ToC, LoF, and LoT lists to be typeset using the default L^AT_EX 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 L^AT_EX custom, the title texts are the values of the `\contentsname`, `\listfigurename` and `\listtablename` commands.

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

`\cftafterZtitleskip`

`\cftZtitlefont`

The code used for typesetting the ToC title looks like

`\cftafterZtitle`

```
{\cfttoctitlefont \contentsname}{\cftaftertoctitle}\par
```

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`).

- `\renewcommand{\cftafterZtitle}{\thispagestyle{empty}}` will make the page with the title empty (i.e., the page number will not be printed).

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.

`\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\par
```

```
{\cftXfont TITLE}{\cftXleader}{\cftXpagefont PAGE}\cftXafterpnum\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,
`\cftXaftersnumb`

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

In the standard classes the ToC entry for a `\part` is just typeset as the number and title, followed by the page number, with the `\cftpartpresnum` macro being called before typesetting the number and title. Due to L^AT_EX idiosyncracies, `\cftpartpresnum` may become doubled in the output if a third-party package behaves differently to that of the default internal L^AT_EX commands. The `tocloft` package contains specific code to prevent this in the case of the KomaScript 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 `\cftpartaftersnum` and `\cftpartaftersnumb` macros have no effect, but they may do something if a non-standard class is used.

<code>\cftXleader</code>	<code>\cftXleader</code> defines the leader between the title and the page number; it can be changed by <code>\renewcommand</code> . The spacing between any dots in the leader is controlled by <code>\cftXdotsep</code> (<code>\@dotsep</code> in Figure 1). It can be changed by <code>\renewcommand</code> and its value must be either a number (e.g., 6.6 or <code>\cftdotsep</code>) or <code>\cftnodots</code> (to disable the dots). The spacing is in terms of <i>math units</i> where there are 18mu to 1em.
<code>\cftXdotsep</code>	
<code>\cftXpagefont</code>	This defines the font to be used for typesetting the page number. It can be changed by <code>\renewcommand</code> .
<code>\cftXafterpnum</code>	This macro is called after the page number has been typeset. Its default is to do nothing. It can be changed by <code>\renewcommand</code> .
<code>\cftsetindents</code>	The command <code>\cftsetindents{<entry>}{<indent>}{<numwidth>}</code> sets the <i><entry></i> 's <i>indent</i> to the length <i><indent></i> and its <i>numwidth</i> to the length <i><numwidth></i> . The <i><entry></i> argument is the name of one of the standard entries (e.g., <code>subsection</code>) or the name of entry that has been defined with the <code>tocloft</code> package. For example <code>\cftsetindents{figure}{0em}{1.5em}</code> 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:

```
\newlength{\mylen} % a "scratch" length
\settoheight{\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}
```

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

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

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}{}
```

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

```
\renewcommand{\cftXleader}{ }
\renewcommand{\cftXafterpnum}{\cftparfillskip}
\renewcommand{\cftpnumalign}{l}
```

By default the `\parfillskip` 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 `\cftparfillskip` 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]{}
\listoftables
\listoffigures
\endgroup
```

`\cftpagenumbersoff`
`\cftpagenumberon`

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.

The command `\cftpagenumberon{⟨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

...

```

\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}{\cftpagenumerson{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}{Z}{...}` has been issued; so *X* is the name of the new counter and corresponds to the *X* in section 2.3, and *Z* 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, `\cftmarkZ`, `\cftbeforeZtitleskip`, `\cftafterZtitleskip`, `\cftZtitlefont`, and `\cftafterZtitle`, 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.

`\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}`.

`\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}{Z}{N}`, where *X* and *Z* 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`, and `\cftXafterpnum`, 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 `\Zdepth` is N or less, where Z is the listing's file extension.

`\theX` The command `\theX` prints the value of the X counter. It is initially defined so that it prints arabic numerals. If the optional *<within>* argument is used, `\theX` is defined as

```
\renewcommand{\theX}{\thewithin.\arabic{X}} otherwise as
\renewcommand{\theX}{\arabic{X}}.
```

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

```
\newlistentry[answer]{subanswer}{1}
\cftsetindents{subanswer}{1.5em}{3.0em}
\renewcommand{\thesubanswer}{\theanswer.\alph{subanswer}}
\newcommand{\subanswer}[1]{%
  \refstepcounter{subanswer}
  \par\textbf{\thesubanswer} #1}
\addcontentsline{ans}{subanswer}{\protect\numberline{\thesubanswer}#1}}
\setcounter{ansdepth}{2}
```

And then:

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

will be typeset as:

Answer 2. Harder

The ...

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

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

To turn off page numbering for the subanswers, do
`\cftpagenumbersoff{subanswer}`

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

```
\newcommand{\subsubpara}{\@startsection{subpara}%
  {6}%
  {\parindent}%
  {3.25ex \@plus1ex \@minus .2ex}%
  {-1em}%
  level
  indent from left margin
  skip above heading
  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.}
...

```

`\cftchapterprecishere`
`\cftchapterprecistoc`

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

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

an entry into $\langle file \rangle$ 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{\langle file \rangle}{\langle kind \rangle}{\langle title \rangle}{\langle page \rangle}` will write a `\contentsline` entry into $\langle file \rangle$ for a $\langle kind \rangle$ entry with title $\langle title \rangle$ and page number $\langle page \rangle$. That is, an entry is made of the form:

```
\contentsline{\langle kind \rangle}{\langle title \rangle}{\langle page \rangle}
```

Any fragile commands used in the arguments must be protected.

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

```
\contentsline{\langle kind \rangle}{\numberline{\langle num \rangle} title}{\langle page \rangle}
```

Any fragile commands used in the arguments must be protected.

As an example of the use of these commands, noting that the default L^AT_EX 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}
\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
`\cftZposthook` problem 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 `\cftZprehook`, 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 are two known potential problems:

- The 1998/11/15 version of `tocbibind` may give surprising results if the `\toctocname`, `\toclofname` or `\toclofname` commands have been used. You should consider getting the current version of `tocbibind`.
- 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`
`\cftasnum`
`\cftasnumb`

Some packages, like the `float` package by Anselm Lingnau, enable the creation of other kinds of *List of . . .*. The `tocloft` package is only minimally able to change the formatting 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 equivalent 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 `minitoc` 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:

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

```
\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)
```

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.

```
2 \newcommand\@cftifundefined[1]{%
3 \begingroup\expandafter\expandafter\expandafter\endgroup
4 \expandafter\ifx\csname #1\endcsname\relax
5 \expandafter\@firstoftwo
6 \else
7 \expandafter\@secondoftwo
8 \fi}
```

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

`\ifcfthaschapter`

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

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

```
11 \newif\ifcftkoma
12 \@cftkomafalse
13 \@ifclassloaded{scrartcl}{\@cftkomatrue}{}
14 \@ifclassloaded{scrreprt}{\@cftkomatrue}{}
15 \@ifclassloaded{scrbook}{\@cftkomatrue}{}

```

`\ifcfttitlesec`

```
16 \newif\ifcfttitlesec
17 \AtBeginDocument{\ifpackageloaded{titlesec}{\@cfttitlesectrue}{}}
```

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

```

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

```

Perhaps quit now.

```
27 \@cftquit
```

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

`\ifcfttocbibind` 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`.

```

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

```

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

```

40 \newif\ifcftnctoc\cftnctocfalse
41 \DeclareOption{titles}{\@cftnctoctrue}
42 %% \ProcessOptions\relax

```

`\ifcftsubfigopt` A boolean used to implement the `subfigure` option.

```

43 \newif\ifcftsubfigopt\cftsubfigoptfalse
44 \DeclareOption{subfigure}{\@cftsubfigopttrue}

```

Process the options.

```

45
46 \ProcessOptions\relax
47

```

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

```

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

```

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

```

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

```

`\@cfttocstart` `\@cfttocfinish` 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.

```

66 \providecommand{\@cfttocstart}{%
67   \if@cfthaschapter
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}

```

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

```

78 \providecommand{\phantomsection}{}
79

```

`\@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}{\contentsname}
85   \else
86     \phantomsection
87     \addcontentsline{toc}{\@tocextra}{\contentsname}
88   \fi
89 \fi}
90

```

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

```

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

```

`\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.

```

94 \AtBeginDocument{%
95 \if@cftnctoc\else
96   \renewcommand{\tableofcontents}{%
97     \cfttocstart

```

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

```

98   \par
99   \begingroup
100     \parindent\z@ \parskip\cftparskip
101     \@cftmaketoctitle

```

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

```

102     \if@cfttocbibind
103       \cftdobibtoc
104     \fi

```

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

```

105     \@starttoc{toc}%
106   \endgroup
107   \@cfttocfinish}
108 \fi
109 }

```

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

```

110 \newcommand{\@cftmaketoctitle}{%
111   \addpenalty\@secpenalty
112   \if@cfthaschapter
113     \vspace*{\cftbeforetoctitleskip}%
114   \else

```

```

115     \vspace{\cftbeforetoctitleskip}%
116     \fi
117     \@cftpagestyle
118     {\interlinepenalty\@M
119     {\cfttoctitlefont\contentsname}{\cftaftertoctitle}%
120     \cftmarktoc
121     \par\nobreak
122     \vskip \cftaftertoctitleskip
123     \@afterheading}}

```

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

```

\cftaftertoctitleskip 124 \newlength{\cftbeforetoctitleskip}
125 \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*`.

```

126 \if@cfthaschapter
127   \setlength{\cftbeforetoctitleskip}{50pt}
128   \setlength{\cftaftertoctitleskip}{40pt}
129 \else
130   \setlength{\cftbeforetoctitleskip}{3.5ex \@plus 1ex \@minus .2ex}
131   \setlength{\cftaftertoctitleskip}{2.3ex \@plus .2ex}
132 \fi

```

`\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).

```

133 \if@cfthaschapter
134   \newcommand{\cfttoctitlefont}{\normalfont\Huge\bfseries}
135   \if@cfkoma\renewcommand{\cfttoctitlefont}{\size@chapter\sectfont}\fi
136 \else
137   \newcommand{\cfttoctitlefont}{\normalfont\Large\bfseries}
138   \if@cfkoma\renewcommand{\cfttoctitlefont}{\size@section\sectfont}\fi
139 \fi
140 \newcommand{\cftaftertoctitle}{}

```

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

```

\cftsetrmarg 141 \newcommand{\cftsetpnumwidth}[1]{\renewcommand{\@pnumwidth}{#1}}
142 \newcommand{\cftsetrmarg}[1]{\renewcommand{\@tocrmarg}{#1}}

```

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

```

143 \newcommand{\cftpnumalign}{r}

```

`\cftdot` In the default ToC, a dotted line can be used to provide a leader between a title and
`\cftdotfill` the 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.

```

144 \providecommand{\cftdot}{.}
145 \providecommand{\cftdotfill}[1]{%
146   \def\@tempa{#1}%
147   \def\@tempb{\cftnodots}%
148   \ifx\@tempa\@tempb
149     \hfill
150   \else
151     \leaders\hbox{$\m@th\mkern #1 mu\hbox{\cftdot}\mkern #1 mu$}\hfill
152   \fi
153 }

```

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

```

154 \providecommand{\cftdotsep}{4.5}
155 \newcommand{\cftnodots}{5000}

```

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>}` 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. The entries are inserted into the file by calling the `\addcontentsline{<file>}{<kind>}{<title>}` command, where `<file>` is the file extension (e.g., `toc`, `lot`) and the other arguments are the same as for the `\contentsline` command. (Arbitrary stuff may also be put into the file via the `\addtocontents{<file>}{<text>}` command). The typesetting of the `\contentsline` entries is performed by commands of the form `\l@kind`. The sectioning and captioning commands call `\addcontentsline` to insert their titles into the `.toc` etc., files.

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

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

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

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

```
156 \newcommand{\cftparfillskip}{\parfillskip=0pt plus1fil}
```

`\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).

```
157 \renewcommand{\numberline}[1]{%
158   \hb@xt@\@tempdima{\@cftbsnum #1\@cftasnum\hfil}\@cftasnumb}
```

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

```
159 \newcommand{\@cftbsnum}{}
160 \newcommand{\@cftasnum}{}
161 \newcommand{\@cftasnumb}{}

```

`\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.

`\ifcftdopart`

```
162 \newif\ifcftdopart
163 \newif\ifcfthaspart
164 \@cftifundefined{part}{\@cfthaspartfalse}{\@cfthasparttrue}
165 \ifcfthaspart
166 \renewcommand*{\l@part}[2]{%
167   \@cftdopartfalse
168   \ifnum \c@tocdepth >-2\relax
169     \ifcfthaschapter
170       \@cftdoparttrue
171     \fi
172     \ifnum \c@tocdepth >\m@ne
173       \ifcfthaschapter\else
174         \@cftdoparttrue
175       \fi
176     \fi
177   \fi
178   \ifcftdopart
179     \ifcfthaschapter
180       \addpenalty{-\@highpenalty}%
181     \else
182       \addpenalty\@secpenalty
183     \fi
184     \addvspace{\cftbeforepartskip}%

```

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

```

185 \begingroup
186   {\leftskip \cftpartindent\relax
187    \rightskip \@tocmarg
188    \parfillskip -\rightskip
189    \parindent \cftpartindent\relax\@afterindenttrue
190    \interlinepenalty\@M
191    \leavevmode
192    \@tempdima \cftpartnumwidth\relax
193    \let\@cftbsnum \cftpartpresnum
194    \let\@cftasnum \cftpartaftersnum
195    \let\@cftasnumb \cftpartaftersnumb
196    \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.

```

197   {\cftpartfont \if@cfkoma\else\if@cfittitlesec\else\cftpartpresnum\fi\fi #1}%
198   \cftpartfillnum{#2}}
199   \nobreak
200   \if@cfthaschapter
201     \global\@nobreaktrue
202     \everypar{\global\@nobreakfalse\everypar{}}%
203   \else
204     \if@compatibility
205       \global\@nobreaktrue
206       \everypar{\global\@nobreakfalse\everypar{}}%
207     \fi
208   \fi
209 \endgroup
210 \fi}
211 \fi

```

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

```

\cftpartnumwidth
\cftpartfont 212 \if@cfthaspart
\cftpartpresnum 213 \newlength{\cftbeforepartskip}
\cftpartaftersnum 214 \setlength{\cftbeforepartskip}{2.25em \@plus\p@}
\cftpartaftersnumb 215 \newlength{\cftpartnumwidth}
\cftpartleader 216 \setlength{\cftpartnumwidth}{0em}
\cftpartdotsep 217 \newcommand{\cftpartfont}{\large\bfseries}
\cftpartpagefont 218 \newcommand{\cftpartpresnum}{}
\cftpartafterpnum 219 \newcommand{\cftpartaftersnum}{}
\cftpartindent 220 \newcommand{\cftpartaftersnumb}{}
\cftpartfillnum 221 \newcommand{\cftpartleader}{\large\bfseries\cftdotfill{\cftpartdotsep}}
222 \newcommand{\cftpartdotsep}{\cftnodots}
223 \newcommand{\cftpartpagefont}{\large\bfseries}
224 \newcommand{\cftpartafterpnum}{}
225 \newlength{\cftpartindent}
226 \setlength{\cftpartindent}{0em}
227 \newcommand{\cftpartfillnum}[1]{%

```

```

228   {\cftpartleader}%
229   {\makebox[\@pnumwidth][\cftpnumalign]{\cftpartpagefont #1}\cftpartafterpnum\par}%
230 }

```

koma classes use some different settings.

```

231 \if@cfkoma
232   \setlength{\cftpartnumwidth}{2em}
233   \renewcommand{\cftpartfont}{\sectfont\large}
234   \renewcommand{\cftpartpagefont}{\sectfont\large}
235 \fi
236 \fi

```

`\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.

```

237 \if@cfthaschapter
238 \renewcommand*{\l@chapter}[2]{%
239   \ifnum \c@tocdepth >\m@ne
240     \addpenalty{-\@highpenalty}%
241     \vskip \cftbeforechapskip
242     {\leftskip \cftchapindent\relax
243      \rightskip \@tocrmarg
244      \parfillskip -\rightskip
245      \parindent \cftchapindent\relax\@afterindenttrue
246      \interlinepenalty\@M
247      \leavevmode
248      \@tempdima \cftchapnumwidth\relax
249      \let\@cftbsnum \cftchappresnum
250      \let\@cftasnum \cftchapaftersnum
251      \let\@cftasnumb \cftchapaftersnumb
252      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
253      {\cftchapfont #1}\nobreak
254      \cftchapfillnum{#2}}%
255   \fi}%
256 \fi

```

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

```

\cftchapnumwidth 257 \if@cfthaschapter
\cftchapfont      258   \newlength{\cftbeforechapskip}
\cftchappresnum   259   \setlength{\cftbeforechapskip}{1.0em \@plus\p@}
\cftchapaftersnum 260   \newlength{\cftchapindent}
\cftchapaftersnumb 261   \setlength{\cftchapindent}{0em}
\cftchapleader    262   \newlength{\cftchapnumwidth}
\cftchapdotsep    263   \setlength{\cftchapnumwidth}{1.5em}
\cftchappagefont  264   \newcommand{\cftchapfont}{\bfseries}
\cftchapafterpnum 265   \newcommand{\cftchappresnum}{}
\cftchapafterpnum 266   \newcommand{\cftchapaftersnum}{}
\cftchapfillnum   267   \newcommand{\cftchapaftersnumb}{}
                  268   \newcommand{\cftchapleader}{\bfseries\cftdotfill{\cftchapdotsep}}

```

```

269 \newcommand{\cftchapdotsep}{\cftnodots}
270 \newcommand{\cftchappagefont}{\bfseries}
271 \newcommand{\cftchapafterpnum}{}
272 \newcommand{\cftchapfillnum}[1]{%
273   {\cftchapleader}\nobreak
274   \makebox[@pnumwidth][\cftpnumalign]{\cftchappagefont #1}\cftchapafterpnum\par
275 }

```

koma classes have different chapter settings.

```

276 \if@cfkoma
277   \renewcommand{\cftchapfont}{\sectfont}
278 \fi
279 \fi
280

```

`\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).

```

281 \renewcommand*{\l@section}[2]{%
282   \ifnum \c@tocdepth >\z@
283     \ifcfthaschapter
284       \vskip \cftbeforesecskip
285     \else
286       \addpenalty\@secpenalty
287       \addvspace{\cftbeforesecskip}
288     \fi
289     {\leftskip \cftsecindent\relax
290      \rightskip \@tocrmarg
291      \parfillskip -\rightskip
292      \parindent \cftsecindent\relax\@afterindenttrue
293      \interlinepenalty\@M
294      \leavevmode
295      \@tempdima \cftsecnumwidth\relax
296      \let\@cftbsnum \cftsecpresnum
297      \let\@cftasnum \cftsecaftersnum
298      \let\@cftasnumb \cftsecaftersnumb
299      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
300      {\cftsecfont #1}\nobreak
301      \cftsecfillnum{#2}}%
302   \fi}

```

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

```

\cftsecindent      303 \newlength{\cftbeforesecskip}
\cftsecnumwidth    304 \newlength{\cftsecindent}
\cftsecfont        305 \newlength{\cftsecnumwidth}
\cftsecpresnum     306 \newcommand{\cftsecpresnum}{}
\cftsecaftersnum   307 \newcommand{\cftsecaftersnum}{}
\cftsecaftersnumb  308 \newcommand{\cftsecaftersnumb}{}
\cftsecleader      309 \ifcfthaschapter
\cftsecdotsep
\cftsecpagefont
\cftsecafterpnum
\cftsecfillnum

```

```

310 \setlength{\cftbeforesecskip}{\z@ \@plus.2\p@}
311 \setlength{\cftsecindent}{1.5em}
312 \setlength{\cftsecnumwidth}{2.3em}
313 \newcommand{\cftsecfont}{\normalfont}
314 \newcommand{\cftsecleader}{\normalfont\cftdotfill{\cftsecdotsep}}
315 \newcommand{\cftsecdotsep}{\cftdotsep}
316 \newcommand{\cftsecpagefont}{\normalfont}
317 \else
318 \setlength{\cftbeforesecskip}{1.0em \@plus\p@}
319 \setlength{\cftsecindent}{0em}
320 \setlength{\cftsecnumwidth}{1.5em}
321 \newcommand{\cftsecfont}{\bfseries}
322 \newcommand{\cftsecleader}{\bfseries\cftdotfill{\cftsecdotsep}}
323 \newcommand{\cftsecdotsep}{\cftnodots}
324 \newcommand{\cftsecpagefont}{\bfseries}
325 \fi
326 \newcommand{\cftsecafterpnum}{}
327 \newcommand{\cftsecfillnum}[1]{%
328   {\cftsecleader}\nobreak
329   \makebox[\@pnumwidth][\cftpnumalign]{\cftsecpagefont #1}\cftsecafterpnum\par
330 }

```

`\l@section` `\l@subsection{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).

```
331 \renewcommand*{\l@subsection}[2]{%
```

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

```
332 \ifnum \c@tocdepth >\@ne
```

Add some vertical space.

```
333 \vskip \cftbeforesubsecskip
```

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

```
334 {\leftskip \cftsubsecindent\relax
```

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

```
335 \rightskip \@tocrmarg
```

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

```
336 \parfillskip -\rightskip
```

Set the paragraph indent to the entry's indentation.

```
337 \parindent \cftsubsecindent\relax\@afterindenttrue
```

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

```
338 \interlinepenalty\@M
```

Make sure that we have left vertical mode.

```
339 \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.

```

340     \@tempdima \cftsubsecnumwidth\relax
341     \let\@cftbsnum \cftsubsecpresnum
342     \let\@cftasnum \cftsubsecaftersnum
343     \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.

```

344     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip

```

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

```

345     {\cftsubsecfont #1}\nobreak

```

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

```

346     \cftsubsecfillnum{#2}}%
347     \fi}

```

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

```

\cftsubsecnumwidth 348 \newlength{\cftbeforesubsecskip}
\cftsubsecfont      349 \setlength{\cftbeforesubsecskip}{\z@ \@plus.2\p@}
\cftsubsecpresnum   350 \newlength{\cftsubsecindent}
\cftsubsecaftersnum 351 \newlength{\cftsubsecnumwidth}
\cftsubsecaftersnumb 352 \if@cfthaschapter
\cftsubsecleader     353 \setlength{\cftsubsecindent}{3.8em}
\cftsubsecdotsep     354 \setlength{\cftsubsecnumwidth}{3.2em}
\cftsubsecpagefont   355 \else
\cftsubsecafterpnum  356 \setlength{\cftsubsecindent}{1.5em}
                    357 \setlength{\cftsubsecnumwidth}{2.3em}
                    358 \fi
                    359 \newcommand{\cftsubsecfont}{\normalfont}
                    360 \newcommand{\cftsubsecpresnum}{}
                    361 \newcommand{\cftsubsecaftersnum}{}
                    362 \newcommand{\cftsubsecaftersnumb}{}
                    363 \newcommand{\cftsubsecleader}{\normalfont\cftdotfill{\cftsubsecdotsep}}
                    364 \newcommand{\cftsubsecdotsep}{\cftdotsep}
                    365 \newcommand{\cftsubsecpagefont}{\normalfont}
                    366 \newcommand{\cftsubsecafterpnum}{}

```

`\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.

```

367 \newcommand{\cftsubsecfillnum}[1]{%
368   {\cftsubsecleader}\nobreak
369   \makebox[\@pnumwidth][\cftpnumalign]{\cftsubsecpagefont #1}\cftsubsecafterpnum\par
370 }

```

`\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).

```

371 \renewcommand*{\l@subsubsection}[2]{%
372   \ifnum \c@tocdepth >\tw@
373     \vskip \cftbeforesubsubsecskip
374     {\leftskip \cftsubsubsecindent\relax
375      \rightskip \@tocrmarg
376      \parfillskip -\rightskip
377      \parindent \cftsubsubsecindent\relax\@afterindenttrue
378      \interlinepenalty\@M
379      \leavevmode
380      \@tempdima \cftsubsubsecnumwidth\relax
381      \let\@cftbsnum \cftsubsubsecpresnum
382      \let\@cftasnum \cftsubsubsecaftersnum
383      \let\@cftasnumb \cftsubsubsecaftersnumb
384      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
385      {\cftsubsubsecfont #1}\nobreak
386      \cftsubsubsecfillnum{#2}}%
387   \fi}

```

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

```

\cftsubsubsecnumwidth 388 \newlength{\cftbeforesubsubsecskip}
\cftsubsubsecfont      389 \setlength{\cftbeforesubsubsecskip}{\z@ \@plus.2\p@}
\cftsubsubsecpresnum   390 \newlength{\cftsubsubsecindent}
\cftsubsubsecaftersnum 391 \newlength{\cftsubsubsecnumwidth}
\cftsubsubsecaftersnumb 392 \if@cfhaschapter
\cftsubsubsecleader    393 \setlength{\cftsubsubsecindent}{7.0em}
\cftsubsubsecdotsep    394 \setlength{\cftsubsubsecnumwidth}{4.1em}
\cftsubsubsecpagefont  395 \else
\cftsubsubsecafterpnum 396 \setlength{\cftsubsubsecindent}{3.8em}
\cftsubsubsecfillnum   397 \setlength{\cftsubsubsecnumwidth}{3.2em}
                        398 \fi
                        399 \newcommand{\cftsubsubsecfont}{\normalfont}
                        400 \newcommand{\cftsubsubsecpresnum}{}
                        401 \newcommand{\cftsubsubsecaftersnum}{}
                        402 \newcommand{\cftsubsubsecaftersnumb}{}
                        403 \newcommand{\cftsubsubsecleader}{\normalfont\cftdotfill{\cftsubsubsecdotsep}}
                        404 \newcommand{\cftsubsubsecdotsep}{\cftdotsep}
                        405 \newcommand{\cftsubsubsecpagefont}{\normalfont}
                        406 \newcommand{\cftsubsubsecafterpnum}{}
                        407 \newcommand{\cftsubsubsecfillnum}[1]{%
                        408   {\cftsubsubsecleader}\nobreak
                        409   \makebox[\@pnumwidth][\cftpnumalign]{\cftsubsubsecpagefont #1}\cftsubsubsecafterpnum\par
                        410 }

```

`\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).

```

411 \renewcommand*{\l@paragraph}[2]{%
412   \ifnum \c@tocdepth >3\relax
413     \vskip \cftbeforeparaskip
414     {\leftskip \cftparaindent\relax
415      \rightskip \@tocrmarg
416      \parfillskip -\rightskip
417      \parindent \cftparaindent\relax\@afterindenttrue
418      \interlinepenalty\@M
419      \leavevmode
420      \@tempdima \cftparanumwidth\relax
421      \let\@cftbsnum \cftparapresnum
422      \let\@cftasnum \cftparaaftersnum
423      \let\@cftasnumb \cftparaaftersnumb
424      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
425      {\cftparafont #1}\nobreak
426      \cftparafillnum{#2}}%
427   \fi}

```

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

```

\cftparanumwidth 428 \newlength{\cftbeforeparaskip}
\cftparafont      429 \setlength{\cftbeforeparaskip}{\z@ \@plus.2\p@}
\cftparapresnum  430 \newlength{\cftparaindent}
\cftparaaftersnum 431 \newlength{\cftparanumwidth}
\cftparaaftersnumb 432 \if@cfthaschapter
\cftparaleader    433 \setlength{\cftparaindent}{10em}
\cftparadotsep    434 \setlength{\cftparanumwidth}{5em}
\cftparapagefont  435 \else
\cftparaafterpnum 436 \setlength{\cftparaindent}{7.0em}
\cftparafillnum   437 \setlength{\cftparanumwidth}{4.1em}
                  438 \fi
                  439 \newcommand{\cftparafont}{\normalfont}
                  440 \newcommand{\cftparapresnum}{}
                  441 \newcommand{\cftparaaftersnum}{}
                  442 \newcommand{\cftparaaftersnumb}{}
                  443 \newcommand{\cftparaleader}{\normalfont\cftdotfill{\cftparadotsep}}
                  444 \newcommand{\cftparadotsep}{\cftdotsep}
                  445 \newcommand{\cftparapagefont}{\normalfont}
                  446 \newcommand{\cftparaafterpnum}{}
                  447 \newcommand{\cftparafillnum}[1]{%
                  448   {\cftparaleader}\nobreak
                  449   \makebox[\@pnumwidth][\cftpnumalign]{\cftparapagefont #1}\cftparaafterpnum\par
                  450 }

```

`\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).

```

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

```



```

452 \ifnum \c@tocdepth >4\relax
453   \vskip \cftbeforesubparaskip
454   {\leftskip \cftsubparaindent\relax
455     \rightskip \@tocrmarg
456     \parfillskip -\rightskip
457     \parindent \cftsubparaindent\relax\@afterindenttrue
458     \interlinepenalty\@M
459     \leavevmode
460     \@tempdima \cftsubparanumwidth\relax
461     \let\@cftbsnum \cftsubparapresnum
462     \let\@cftasnum \cftsubparaaftersnum
463     \let\@cftasnumb \cftsubparaaftersnumb
464     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
465     {\cftsubparafont #1}\nobreak
466     \cftsubparafillnum{#2}}%
467 \fi}

```

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

```

\cftsubparanumwidth 468 \newlength{\cftbeforesubparaskip}
\cftsubparafont      469 \setlength{\cftbeforesubparaskip}{\z@ \@plus.2\p@}
\cftsubparapresnum  470 \newlength{\cftsubparaindent}
\cftsubparaaftersnum 471 \newlength{\cftsubparanumwidth}
\cftsubparaaftersnumb 472 \if@cfthaschapter
\cftsubparaleader    473   \setlength{\cftsubparaindent}{12em}
\cftsubparadotsep    474   \setlength{\cftsubparanumwidth}{6em}
\cftsubparapagefont  475 \else
\cftsubparaafterpnum 476   \setlength{\cftsubparaindent}{10em}
\cftsubparafillnum   477   \setlength{\cftsubparanumwidth}{5em}
                     478 \fi
                     479 \newcommand{\cftsubparafont}{\normalfont}
                     480 \newcommand{\cftsubparapresnum}{}
                     481 \newcommand{\cftsubparaaftersnum}{}
                     482 \newcommand{\cftsubparaaftersnumb}{}
                     483 \newcommand{\cftsubparaleader}{\normalfont\cftdotfill{\cftsubparadotsep}}
                     484 \newcommand{\cftsubparadotsep}{\cftdotsep}
                     485 \newcommand{\cftsubparapagefont}{\normalfont}
                     486 \newcommand{\cftsubparaafterpnum}{}
                     487 \newcommand{\cftsubparafillnum}[1]{%
                     488   {\cftsubparaleader}\nobreak
                     489   \makebox[\@pnumwidth][\cftpnumalign]{\cftsubparapagefont #1}\cftsubparaafterpnum\par
                     490 }

```

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

```

491 \newcommand{\@cftdobiblof}{%
492   \if@dotoclof
493     \if@bibchapter
494       \phantomsection

```

```

495     \addcontentsline{toc}{chapter}{\listfigurename}
496     \else
497     \phantomsection
498     \addcontentsline{toc}{\@tocextra}{\listfigurename}
499     \fi
500 \fi}
501

```

`\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).

```

502 \AtBeginDocument{%
503 \if@cftnctoc\else
504 \renewcommand{\listoffigures}{%
505   \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.

```

506   \par
507   \begingroup
508     \parindent\z@ \parskip\cftparskip
509     \@cftmakeoftitle
510     \if@cfttocbibind
511       \cftdobiblof
512     \fi
513     \@starttoc{lof}%
514   \endgroup

```

Finally, restore any multicolumn typesetting.

```

515   \@cfttocfinish}%
516 \fi
517 }
518

```

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

```

519 \newcommand{\@cftmakeoftitle}{%
520   \addpenalty\@secpenalty
521   \if@cftthaschapter
522     \vspace*{\cftbeforelofttitleskip}
523   \else
524     \vspace{\cftbeforelofttitleskip}
525   \fi
526   \@cftpagestyle
527   {\interlinepenalty\M
528    {\cftlofttitlefont\listfigurename}{\cftafterlofttitle}
529   \cftmarklof
530   \par\nobreak

```

```

531 \vskip \cftafterlofttitleskip
532 \@afterheading}}
533

```

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

```

\cftafterlofttitleskip 534 \newlength{\cftbeforelofttitleskip}
535 \newlength{\cftafterlofttitleskip}

```

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*`.

```

536 \ifcfthaschapter
537 \setlength{\cftbeforelofttitleskip}{50pt}
538 \setlength{\cftafterlofttitleskip}{40pt}
539 \else
540 \setlength{\cftbeforelofttitleskip}{3.5ex \@plus 1ex \@minus .2ex}
541 \setlength{\cftafterlofttitleskip}{2.3ex \@plus .2ex}
542 \fi

```

`\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).

```

543 \ifcfthaschapter
544 \newcommand{\cftlofttitlefont}{\normalfont\Huge\bfseries}
545 \ifcftkoma\renewcommand{\cftlofttitlefont}{\size@chapter\sectfont}\fi
546 \else
547 \newcommand{\cftlofttitlefont}{\normalfont\Large\bfseries}
548 \ifcftkoma\renewcommand{\cftlofttitlefont}{\size@section\sectfont}\fi
549 \fi
550 \newcommand{\cftafterlofttitle}{}
551

```

`\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).

```

552 \renewcommand*{\l@figure}[2]{%
553 \ifnum \c@lofdepth >\z@
554 \vskip \cftbeforefigskip
555 {\leftskip \cftfigindent\relax
556 \rightskip \@tocrmarg
557 \parfillskip -\rightskip
558 \parindent \cftfigindent\relax\@afterindenttrue
559 \interlinepenalty\@M
560 \leavevmode
561 \@tempdima \cftfignumwidth\relax
562 \let\@cftbsnum \cftfigpresnum
563 \let\@cftasnum \cftfigaftersnum
564 \let\@cftasnumb \cftfigaftersnumb

```

```

565     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
566     {\cftfigfont #1}\nobreak
567     \cftfigfillnum{#2}}%
568   \fi
569 }

```

`\cftbeforefigskip` These are the user commands to control the typesetting of Figure caption entries.

`\cftfigindent` They are initialised to give the standard appearance.

```

\cftfignumwidth 570 \newlength{\cftbeforefigskip}
\cftfigfont      571 \setlength{\cftbeforefigskip}{\z@ \@plus.2\p@}
\cftfigpresnum   572 \newlength{\cftfigindent}
\cftfigaftersnum 573 \setlength{\cftfigindent}{1.5em}
\cftfigaftersnumb 574 \newlength{\cftfignumwidth}
\cftfigleader     575 \setlength{\cftfignumwidth}{2.3em}
\cftfigdotsep    576 \newcommand{\cftfigfont}{\normalfont}
\cftfigpresnum    577 \newcommand{\cftfigpresnum}{}
\cftfigpagefont  578 \newcommand{\cftfigaftersnum}{}
\cftfigafterpnum 579 \newcommand{\cftfigaftersnumb}{}
\cftfigfillnum   580 \newcommand{\cftfigleader}{\normalfont\cftdotfill{\cftfigdotsep}}
                    581 \newcommand{\cftfigdotsep}{\cftdotsep}
                    582 \newcommand{\cftfigpagefont}{\normalfont}
                    583 \newcommand{\cftfigafterpnum}{}
                    584 \newcommand{\cftfigfillnum}[1]{%
585   {\cftfigleader}\nobreak
586   \makebox[\@pnumwidth] [\cftpnumalign]{\cftfigpagefont #1}\cftfigafterpnum\par
587 }

```

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

```

588 \if@cftsubfigopt\else
589   \newcounter{lofdepth}\setcounter{lofdepth}{1}
590   \newcounter{lotdepth}\setcounter{lotdepth}{1}
591 \fi
592

```

`\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`.

```

593 \newcommand{\cftdobiblot}{%
594   \if@dotoclot
595     \if@bibchapter
596       \phantomsection
597       \addcontentsline{toc}{chapter}{\listtablename}
598     \else
599       \phantomsection
600       \addcontentsline{toc}{\@tocextra}{\listtablename}
601     \fi
602   \fi}
603

```

`\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).

```
604 \AtBeginDocument{%
605 \if@cftnctoc\else
606 \renewcommand{\listoftables}{%
607 \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.

```
608 \par
609 \begingroup
610 \parindent\z@ \parskip\cftparskip
611 \cftmakelottitle
612 \if@cfttocbibind
613 \cftdobiblot
614 \fi
615 \@starttoc{lot}%
616 \endgroup
```

Finally, restore any multicolumn typesetting.

```
617 \cfttocfinish}%
618 \fi
619 }
620
```

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

```
621 \newcommand{\cftmakelottitle}{%
622 \addpenalty\@secpenalty
623 \if@cfthaschapter
624 \vspace*{\cftbeforelottitleskip}
625 \else
626 \vspace{\cftbeforelottitleskip}
627 \fi
628 \@cftpagestyle
629 {\interlinepenalty\M
630 {\cftlottitlefont\listtablename}{\cftafterlottitle}
631 \cftmarklot
632 \par\nobreak
633 \vskip \cftafterlottitleskip
634 \@afterheading}}
635
```

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

```
\cftafterlottitleskip 636 \newlength{\cftbeforelottitleskip}
637 \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*`.

```
638 \ifcfthaschapter
639   \setlength{\cftbeforelottitleskip}{50pt}
640   \setlength{\cftafterlottitleskip}{40pt}
641 \else
642   \setlength{\cftbeforelottitleskip}{3.5ex \@plus 1ex \@minus .2ex}
643   \setlength{\cftafterlottitleskip}{2.3ex \@plus .2ex}
644 \fi
```

`\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).

```
645 \ifcfthaschapter
646   \newcommand{\cftlottitlefont}{\normalfont\Huge\bfseries}
647   \ifcftkoma\renewcommand{\cftlottitlefont}{\size@chapter\sectfont}\fi
648 \else
649   \newcommand{\cftlottitlefont}{\normalfont\Large\bfseries}
650   \ifcftkoma\renewcommand{\cftlottitlefont}{\size@section\sectfont}\fi
651 \fi
652 \newcommand{\cftafterlottitle}{}
653
```

`\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).

```
654 \renewcommand*{\l@table}[2]{%
655   \ifnum\c@lotdepth >\z@
656     \vskip \cftbeforetabskip
657     {\leftskip \cfttabindent\relax
658      \rightskip \@tocrmarg
659      \parfillskip -\rightskip
660      \parindent \cfttabindent\relax\@afterindenttrue
661      \interlinepenalty\@M
662      \leavevmode
663      \@tempdima \cfttabnumwidth\relax
664      \let\@cftbsnum \cfttabpresnum
665      \let\@cftasnum \cfttabaftersnum
666      \let\@cftasnumb \cfttabaftersnumb
667      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
668      {\cfttabfont #1}\nobreak
669      \cfttabfillnum{#2}}%
670   \fi
671 }
```

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

```
\cfttabindent
\cfttabnumwidth
\cfttabfont
\cfttabpresnum
\cfttabaftersnum
\cfttabaftersnumb
\cfttableader
\cfttabdotsep
\cfttabpagefont
\cfttabafterpnum
\cfttabfillnum
```

```

672 \newlength{\cftbeforetabskip}
673 \setlength{\cftbeforetabskip}{\z@ \@plus.2\p@}
674 \newlength{\cfttabindent}
675 \setlength{\cfttabindent}{1.5em}
676 \newlength{\cfttabnumwidth}
677 \setlength{\cfttabnumwidth}{2.3em}
678 \newcommand{\cfttabfont}{\normalfont}
679 \newcommand{\cfttabpresnum}{}
680 \newcommand{\cfttabaftersnum}{}
681 \newcommand{\cfttabaftersnumb}{}
682 \newcommand{\cfttableader}{\normalfont\cftdotfill{\cfttabdotsep}}
683 \newcommand{\cfttabdotsep}{\cftdotsep}
684 \newcommand{\cfttabpagefont}{\normalfont}
685 \newcommand{\cfttabafterpnum}{}
686 \newcommand{\cfttabfillnum}[1]{%
687   {\cfttableader}\nobreak
688   \makebox[\@pnumwidth][\cftpnumalign]{\cfttabpagefont #1}\cfttabafterpnum\par
689 }

```

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.
690 \newcommand{\@cftl@subfig}{%

\l@subfigure \l@subfigure{\langle title \rangle}{\langle page \rangle} 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.
691 \renewcommand*{\l@subfigure}[2]{%
692   \ifnum \c@lofdepth > \toclevel@subfigure
693     \vskip \cftbeforesubfigskip
694     {\leftskip \cftsubfigindent\relax
695       \rightskip \@tocrmarg
696       \parfillskip -\rightskip
697       \parindent \cftsubfigindent\relax\@afterindenttrue
698       \interlinepenalty\@M
699       \leavevmode
700       \@tempdima \cftsubfignumwidth\relax
701       \let\@cftbsnum \cftsubfigpresnum
702       \let\@cftasnum \cftsubfigaftersnum
703       \let\@cftasnumb \cftsubfigaftersnumb
704       \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
705       {\cftsubfigfont ##1}\nobreak
706       \cftsubfigfillnum{##2}}%
707   \fi
708 }%

```

709 }
710

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

711 `\newcommand{\cftsetsubfig}{%`

```
\cftbeforesubfigskip
  \cftsubfigindent 712 \newlength{\cftbeforesubfigskip}
  \cftsubfignumwidth 713 \setlength{\cftbeforesubfigskip}{\z@ \@plus.2\p@}
  \cftsubfigfont 714 \newlength{\cftsubfigindent}
  \cftsubfigpresnum 715 \setlength{\cftsubfigindent}{3.8em}
  \cftsubfigaftersnum 716 \newlength{\cftsubfignumwidth}
\cftsubfigaftersnumb 717 \setlength{\cftsubfignumwidth}{2.5em}
  \cftsubfigleader 718 \newcommand{\cftsubfigfont}{\normalfont}
  \cftsubfigdotsep 719 \newcommand{\cftsubfigpresnum}{}
  \cftsubfigpagefont 720 \newcommand{\cftsubfigaftersnum}{}
  \cftsubfigafterpnum 721 \newcommand{\cftsubfigaftersnumb}{}
  \toclevel@subfig 722 \newcommand{\cftsubfigleader}{\normalfont\cftdotfill{\cftsubtabdotsep}}
  \cftsubfigfillnum 723 \newcommand{\cftsubfigdotsep}{\cftdotsep}
  724 \newcommand{\cftsubfigpagefont}{\normalfont}
  725 \newcommand{\cftsubfigafterpnum}{}
  726 \providecommand{\toclevel@subfigure}{1}
  727 \newcommand{\cftsubfigfillnum}[1]{%
  728   {\cftsubfigleader}\nobreak
  729   \makebox[\@pnumwidth] [\cftpnumalign]{\cftsubfigpagefont ##1}\cftsubfigafterpnum\par
  730 }
```

This is the end of `\cftsetsubfig`.

731 }
732

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

733 `\newcommand{\cftl@subtab}{%`

`\l@subtable` `\l@subtable{<title>}{<page>}` typesets the LoT entry for a subtable caption heading. It is essentially the same as the parameterised code for `\l@table` except account has to be taken of `lotdepth`.

```
734 \renewcommand*{\l@subtable}[2]{%
735   \ifnum \c@lotdepth > \toclevel@subtable
736     \vskip \cftbeforesubtabskip
737     {\leftskip \cftsubtabindent\relax
738       \rightskip \@tocrmarg
739       \parfillskip -\rightskip
740       \parindent \cftsubtabindent\relax\@afterindenttrue
741       \interlinepenalty\@M
742       \leavevmode
743       \@tempdima \cftsubtabnumwidth\relax
744       \let\cftbsnum \cftsubtabpresnum
745       \let\cftasnum \cftsubtabaftersnum
```



```

746 \let\cftasnumb \cftsubtabaftersnumb
747 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
748 {\cftsubtabfont ##1}\nobreak
749 \cftsubtabfillnum{##2}}%
750 \fi
751 }%
752 }

```

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

```
753 \newcommand{\cftsetsubtab}{%
```

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

```

\cftsubtabindent
\cftsubtabnumwidth 754 \newlength{\cftbeforesubtabskip}
\cftsubtabfont 755 \setlength{\cftbeforesubtabskip}{\z@ \@plus.2\p@}
\cftsubtabpresnum 756 \newlength{\cftsubtabindent}
\cftsubtabaftersnum 757 \setlength{\cftsubtabindent}{3.8em}
\cftsubtabaftersnumb 758 \newlength{\cftsubtabnumwidth}
\cftsubtableader 759 \setlength{\cftsubtabnumwidth}{2.5em}
\cftsubtabdotsep 760 \newcommand{\cftsubtabfont}{\normalfont}
\cftsubtabpagefont 761 \newcommand{\cftsubtabpresnum}{}
\cftsubtabafterpnum 762 \newcommand{\cftsubtabaftersnum}{}
\toclevel@subtable 763 \newcommand{\cftsubtabaftersnumb}{}
\cftsubtabfillnum 764 \newcommand{\cftsubtableader}{\normalfont\cftdotfill{\cftsubtabdotsep}}
765 \newcommand{\cftsubtabdotsep}{\cftdotsep}
766 \newcommand{\cftsubtabpagefont}{\normalfont}
767 \newcommand{\cftsubtabafterpnum}{}
768 \providecommand{\toclevel@subtable}{1}
769 \newcommand{\cftsubtabfillnum}[1]{%
770 {\cftsubtableader}\nobreak
771 \makebox[\@pnumwidth][\cftpnumalign]{\cftsubtabpagefont ##1}\cftsubtabafterpnum\par
772 }

```

This is the end of `\cftsetsubtab`.

```

773 }
774

```

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.

```

775
776 \if@cftsubfigopt
777 \cftsetsubfig\cftsetsubtab
778 \AtBeginDocument{\cftl@subfig\cftl@subtab}
779 \fi
780 %% \AtBeginDocument{\if@cftsubfigopt
781 %% \cftsetsubfig\cftsetsubtab
782 %% \cftl@subfig\cftl@subtab
783 %% \fi}
784

```

3.2 New list of...

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

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

`\c@X` Check if *<within>* and *<counter>* have been defined. It is an error if *<within>* has not
`\theX` been defined, and an error if *<counter>* has been defined. Set the default counter values.

```
786 \ifundefined{c@#2}{%    check & set the counter
787   \ifx \@empty#1\relax
788     \newcounter{#2}
789   \else
790     \ifundefined{c@#1}{\PackageWarning{tocloft}%
791       {#1 has no counter for use as a 'within'}
792     \newcounter{#2}}%
793     {\newcounter{#2}[#1]%
794     \expandafter\edef\csname the#2\endcsname{%
795       \expandafter\noexpand\csname the#1\endcsname.\noexpand\arabic{#2}}}
796   \fi
797   \setcounter{#2}{0}
798 }
799 {\PackageError{tocloft}{#2 has been previously defined}{\@eha}}
800
```

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

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

```
801 \@namedef{l@#2}##1##2{%
```

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

```
802   \ifnum \@nameuse{c@#3depth} > #4\relax
803     \vskip \@nameuse{cftbefore#2skip}
804     {\leftskip \@nameuse{cft#2indent}\relax
805     \rightskip \@tocrmarg
806     \parfillskip -\rightskip
807     \parindent \@nameuse{cft#2indent}\relax\@afterindenttrue
808     \interlinepenalty\@M
809     \leavevmode
810     \@tempdima \@nameuse{cft#2numwidth}\relax
811     \expandafter\let\expandafter\@cftbsnum\csname cft#2presnum\endcsname
812     \expandafter\let\expandafter\@cftasnum\csname cft#2aftersnum\endcsname
813     \expandafter\let\expandafter\@cftasnumb\csname cft#2aftersnumb\endcsname
814     \advance\leftskip\@tempdima \null\nobreak\hskip -\leftskip
815     {\@nameuse{cft#2font}##1}\nobreak
816     \@nameuse{cft#2fillnum}{##2}}%
817   \fi
818 } % end of \l@#2
819
```

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.

```
\cftbeforeXskip
820 \expandafter\newlength\csname cftbefore#2skip\endcsname
821 \setlength{\@nameuse{cftbefore#2skip}}{\z@ \@plus .2\p@}
```

```
\cftXindent
\cftXnumwidth 822 \expandafter\newlength\csname cft#2indent\endcsname
823 \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.

```
824 \ifcase #4\relax % 0
825 \setlength{\@nameuse{cft#2indent}}{0em}
826 \setlength{\@nameuse{cft#2numwidth}}{1.5em}
827 \or % 1
828 \setlength{\@nameuse{cft#2indent}}{1.5em}
829 \setlength{\@nameuse{cft#2numwidth}}{2.3em}
830 \or % 2
831 \setlength{\@nameuse{cft#2indent}}{3.8em}
832 \setlength{\@nameuse{cft#2numwidth}}{3.2em}
833 \or % 3
834 \setlength{\@nameuse{cft#2indent}}{7.0em}
835 \setlength{\@nameuse{cft#2numwidth}}{4.1em}
836 \else % anything else
837 \setlength{\@nameuse{cft#2indent}}{10.0em}
838 \setlength{\@nameuse{cft#2numwidth}}{5.0em}
839 \fi
```

`\cftXfont` And the remaining commands.

```
\cftXpresnum 840 \@namedef{cft#2font}{\normalfont}
\cftXaftersnum 841 \@namedef{cft#2presnum}{ }
\cftXaftersnumb 842 \@namedef{cft#2aftersnumb}{ }
\cftXdotsep 843 \@namedef{cft#2aftersnumb}{ }
\cftXleader 844 \@namedef{cft#2dotsep}{\cftdotsep}
\cftXpagefont 845 \@namedef{cft#2leader}{\normalfont\cftdotfill{\@nameuse{cft#2dotsep}}}
\cftXafterpnum 846 \@namedef{cft#2pagefont}{\normalfont}
847 \@namedef{cft#2afterpnum}{ }
```

`\toclevel@X` The hyperref package needs a command `\toclevel@X`, holding the $\langle level-1 \rangle$ value.

```
848 \@namedef{toclevel@#2}{#4}
```

`\cftXfillnum` Typeset the leader and page number.

```
849 \@namedef{cft#2fillnum}##1{%
850 {\@nameuse{cft#2leader}}\nobreak
851 \makebox[\@pnumwidth][\cftpnumalign]{\@nameuse{cft#2pagefont}##1}\@nameuse{cft#2afterpnum}
852 }
```

This ends the definition of `\newlistentry`.

```
853 }
```

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

```
854 \newcommand{\newlistof}[4] [\@empty] {%
```

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

```
855 \ifx \@empty#1\relax
856 \newlistentry{#2}{#3}{0}
857 \else
858 \newlistentry[#1]{#2}{#3}{0}
859 \fi
```

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

```
\Zdepth 860 \@namedef{ext@#2}{#3}
861 \newcounter{#3depth}
862 \setcounter{#3depth}{1}
```

`\cftmarkZ` The heading marks for the listing.

```
863 \if@cftkoma
864 \@namedef{cftmark#3}{%
865 \mkboth{#4}{#4}}
866 \else
867 \@namedef{cftmark#3}{%
868 \mkboth{\MakeUppercase{#4}}{\MakeUppercase{#4}}}
869 \fi
```

`\listofX` Typeset the listing title and entries.

```
870 \if@cftnctoc
```

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

```
871 \@namedef{listof#2}{%
872 \cfttocstart
873 \if@cfthaschapter
874 \chapter*{#4}
875 \else
876 \section*{#4}
877 \fi
878 \@nameuse{cftmark#3}
879 \starttoc{#3}%
880 \cfttocfinish}
881 \else
```

Otherwise use the fully parameterised definition.

```
882 \@namedef{listof#2}{%
883 \cfttocstart
884 \par
885 \begingroup
```

```

886     \parindent\z@ \parskip\cftparskip
887     \@nameuse{cftmake#3title}
888     \@starttoc{#3}%
889     \endgroup
890     \@cfttocfinish}
891 \fi
892

```

`\@cftmakeZtitle` Typeset the title.

```

893 \@namedef{cftmake#3title}{%
894   \addpenalty\@secpenalty
895   \if@cfthaschapter
896     \vspace*{\@nameuse{cftbefore#3titleskip}}%
897   \else
898     \vspace{\@nameuse{cftbefore#3titleskip}}%
899   \fi
900   \@cftpagestyle
901   {\interlinepenalty\@M
902    {\@nameuse{cft#3titlefont}#4}{\@nameuse{cftafter#3title}}%
903    \@nameuse{cftmark#3}%
904    \par\nobreak
905    \vskip \@nameuse{cftafter#3titleskip}%
906    \@afterheading}}
907

```

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

```

\cftZtitlefont
908 \expandafter\newlength\csname cftbefore#3titleskip\endcsname
909 \expandafter\newlength\csname cftafter#3titleskip\endcsname
910 \if@cfthaschapter
911   \setlength{\@nameuse{cftbefore#3titleskip}}{50pt}
912   \setlength{\@nameuse{cftafter#3titleskip}}{40pt}
913   \if@cftkoma
914     \@namedef{cft#3titlefont}{\size@chapter\sectfont}
915   \else
916     \@namedef{cft#3titlefont}{\normalfont\Huge\bfseries}
917   \fi
918 \else
919   \setlength{\@nameuse{cftbefore#3titleskip}}{3.5ex \@plus 1ex \@minus .2ex}
920   \setlength{\@nameuse{cftafter#3titleskip}}{2.3ex \@plus .2ex}
921   \if@cftkoma
922     \@namedef{cft#3titlefont}{\size@section\sectfont}
923   \else
924     \@namedef{cft#3titlefont}{\normalfont\Huge\bfseries}
925   \fi
926 \fi

```

`\cftafterZtitle` Something to go after the title.

```

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

```

```

\cftZprehook Hooks before and after the list of entries.
\cftZposthook 928 \namedef{cft#3prehook}{}
               929 \namedef{cft#3posthook}{}

               This is the end of the definition of \newlistof.
               930 }

\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.

931 \newcommand{\cftsetindents}[3]{%
932 \def\@cftemp{#1}
933 \ifx\@cftemp\cftchapname
934 \@cftsetindents{chap}{#2}{#3}
935 \else
936 \ifx\@cftemp\cftsecname \@cftsetindents{sec}{#2}{#3}
937 \else
938 \ifx\@cftemp\cftsubsecname \@cftsetindents{subsec}{#2}{#3}
939 \else
940 \ifx\@cftemp\cftsubsubsecname \@cftsetindents{subsubsec}{#2}{#3}
941 \else
942 \ifx\@cftemp\cftparaname \@cftsetindents{para}{#2}{#3}
943 \else
944 \ifx\@cftemp\cftsubparaname \@cftsetindents{subpara}{#2}{#3}
945 \else
946 \ifx\@cftemp\cftfigname \@cftsetindents{fig}{#2}{#3}
947 \else
948 \ifx\@cftemp\cftsubfigname \@cftsetindents{subfig}{#2}{#3}
949 \else
950 \ifx\@cftemp\cfttabname \@cftsetindents{tab}{#2}{#3}
951 \else
952 \ifx\@cftemp\cftsubtabname \@cftsetindents{subtab}{#2}{#3}
953 \else
954 \@cftsetindents{#1}{#2}{#3}
955 \fi
956 \fi
957 \fi
958 \fi
959 \fi
960 \fi
961 \fi
962 \fi
963 \fi
964 \fi
965 }
966

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

```

```

967 \newcommand{\@cftsetindents}[3]{%
968   \setlength{\@nameuse{cft#1indent}}{#2}
969   \setlength{\@nameuse{cft#1numwidth}}{#3}
970 }
971

```

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.

```

972 \newcommand{\@cftpnumoff}[1]{%
973   \@namedef{cft#1fillnum}##1{%
974     \cftparfillskip\@nameuse{cft#1afterpnum}\par}}
975

```

`\cftchapname` `\cftsecname` `\cftsubsecname` 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.

```

\cftsubsubsecname 976 \newcommand*{\cftchapname}{chapter}
\cftparaname       977 \newcommand*{\cftsecname}{section}
\cftsubparaname    978 \newcommand*{\cftsubsecname}{subsection}
\cftfigname        979 \newcommand*{\cftsubsubsecname}{subsubsection}
\cftsubfigname     980 \newcommand*{\cftparaname}{paragraph}
\cfttabname        981 \newcommand*{\cftsubparaname}{subparagraph}
\cftsubtabname     982 \newcommand*{\cftfigname}{figure}
                   983 \newcommand*{\cftsubfigname}{subfigure}
                   984 \newcommand*{\cfttabname}{table}
                   985 \newcommand*{\cftsubtabname}{subtable}
                   986

```

`\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.

```

987 \DeclareRobustCommand{\cftpagenumbersoff}[1]{%
988   \def\@cftemp{#1}
989   \ifx\@cftemp\cftchapname
990     \@cftpnumoff{chap}
991   \else
992     \ifx\@cftemp\cftsecname \cftpnumoff{sec}
993   \else
994     \ifx\@cftemp\cftsubsecname \cftpnumoff{subsec}
995   \else
996     \ifx\@cftemp\cftsubsubsecname \cftpnumoff{subsubsec}
997   \else

```

```

998     \ifx\@cftemp\cftparaname \@cftpnumoff{para}
999     \else
1000     \ifx\@cftemp\cftsubparaname \@cftpnumoff{subpara}
1001     \else
1002     \ifx\@cftemp\cftfigname \@cftpnumoff{fig}
1003     \else
1004     \ifx\@cftemp\cftsubfigname \@cftpnumoff{subfig}
1005     \else
1006     \ifx\@cftemp\cfttabname \@cftpnumoff{tab}
1007     \else
1008     \ifx\@cftemp\cftsubtabname \@cftpnumoff{subtab}
1009     \else
1010     \@cftpnumoff{#1}
1011     \fi
1012     \fi
1013     \fi
1014     \fi
1015     \fi
1016     \fi
1017     \fi
1018     \fi
1019     \fi
1020 \fi
1021 }
1022

```

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

```

1023 \DeclareRobustCommand{\cftpagenumberon}[1]{%
1024   \def\@cftemp{#1}
1025   \ifx\@cftemp\cftchapname
1026     \@cftpnumon{chap}
1027   \else
1028     \ifx\@cftemp\cftsecname \@cftpnumon{sec}
1029   \else
1030     \ifx\@cftemp\cftsubsecname \@cftpnumon{subsec}
1031   \else
1032     \ifx\@cftemp\cftsubsubsecname \@cftpnumon{subsubsec}
1033   \else
1034     \ifx\@cftemp\cftparaname \@cftpnumon{para}
1035   \else
1036     \ifx\@cftemp\cftsubparaname \@cftpnumon{subpara}
1037   \else
1038     \ifx\@cftemp\cftfigname \@cftpnumon{fig}
1039   \else
1040     \ifx\@cftemp\cftsubfigname \@cftpnumon{subfig}
1041   \else
1042     \ifx\@cftemp\cfttabname \@cftpnumon{tab}
1043   \else
1044     \ifx\@cftemp\cftsubtabname \@cftpnumon{subtab}

```



```

1045             \else
1046             \cftpnumon{#1}
1047             \fi
1048             \fi
1049             \fi
1050             \fi
1051             \fi
1052             \fi
1053             \fi
1054             \fi
1055             \fi
1056             \fi
1057 }
1058

```

`\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.

```

1059 \newcommand{\cftpnumon}[1]{%
1060   \namedef{cft#1fillnum}##1{%
1061     {\@nameuse{cft#1leader}}\nobreak
1062     \makebox[\@pnumwidth][\cftpnumalign]{\@nameuse{cft#1pagefont}##1}\@nameuse{cft#1afterpnum}}
1063   }%
1064 }

```

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.

```

1065 \newcommand{\cftchapterprecis}[1]{%
1066   \cftchapterprecishere{#1}
1067   \cftchapterprecistoc{#1}}

```

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

```

1068 \newcommand{\cftchapterprecishere}[1]{%
1069   \vspace*{-2\baselineskip}

```

Typeset its argument using italic font in a quote environment.

```

1070   \begin{quote}\textit{#1}\end{quote}}

```

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

1071 `\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.

1072 `{\leftskip \cftchapindent\relax`
 1073 `\advance\leftskip \cftchapnumwidth\relax`

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

1074 `\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.

1075 `\textit{#1}\protect\par}}`
 1076

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

1077 `\newcommand{\cftlocalchange}[3]{%`
 1078 `\addtocontents{#1}{\protect\cftsetpnumwidth{#2} \protect\cftsetrmarg{#3}}`

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

1079 `\newcommand{\cftaddtitleline}[4]{\addtocontents{#1}{%`
 1080 `\protect\contentsline{#2}{#3}{#4}}`

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

1081 `\newcommand{\cftaddnumtitleline}[5]{\addtocontents{#1}{%`
 1082 `\protect\contentsline{#2}{\protect\numberline{#3}#4}#5}}`

And, if dear old `hyperref` has been used, we have to fix up these two macros.

1083 `\AtBeginDocument{%`
 1084 `\@ifpackageloaded{hyperref}{%`
 1085 `\renewcommand{\cftaddtitleline}[4]{\addtocontents{#1}{%`
 1086 `\protect\contentsline{#2}{#3}{#4}{\@currentHref}}`
 1087 `\renewcommand{\cftaddnumtitleline}[5]{\addtocontents{#1}{%`
 1088 `\protect\contentsline{#2}{\protect\numberline{#3}#4}#5}{\@currentHref}}`
 1089 `}}`
 1090 `}`
 1091

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

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

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

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

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

```

1092 \def\starttoc@latex@orig#1{%
1093   \begingroup
1094     \makeatletter
1095     \@input{\jobname.#1}%
1096     \if@filesw
1097       \expandafter\newwrite\csname tf@#1\endcsname
1098       \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
1099     \fi
1100   \@nobreakfalse
1101 \endgroup}
1102 \ifx\starttoc\starttoc@latex@orig
1103   \def\starttoc#1{%

```

```

1104 \csname cft#1prehook\endcsname
1105 \begingroup
1106 \makeatletter
1107 \@input{\jobname.#1}%
1108 \if@filesw
1109 \expandafter\newwrite\csname tf@#1\endcsname
1110 \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
1111 \fi
1112 \@nobreakfalse
1113 \endgroup
1114 \csname cft#1posthook\endcsname}
1115 \newcommand\cfttocprehook{}
1116 \newcommand\cftlofprehook{}
1117 \newcommand\cftlotprehook{}
1118 \newcommand\cfttocposthook{}
1119 \newcommand\cftlofposthook{}
1120 \newcommand\cftlotposthook{}
1121 \else
1122 \PackageWarning{tocloft}{\string\@starttoc\space has already been redefined; tocloft bailing}
1123 \fi

```

The end of this package.

```
1124 </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

Numbers written in *italics* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols	1032, 1034,	\cftsetindents ...
\@afterheading	1036, 1038, 934, 936,
. 123, 532, 634, 906	1040, 1042, 1044	938, 940, 942,
\@afterindenttrue .	\cftaschapterfalse 19	944, 946, 948,
. . . . 189, 245,	\cftaschaptertrue 25	950, 952, 954, <u>967</u>
292, 337, 377,	\cftaspartfalse . 164	\cftsetsubfig
417, 457, 558,	\cftasparttrue .. 164 <u>711</u> , 777, 781
660, 697, 740, 807	\cftifundefined . .	\cftsetsubtab
\cftasnum . 18, 158, <u>2</u> , 18, 20, 164 <u>753</u> , 777, 781
<u>159</u> , 194, 250,	\cftkomafalse 12	\cftsubfigoptfalse 43
297, 342, 382,	\cftkomatrue 13, 14, 15	\cftsubfigopttrue . 44
422, 462, 563,	\cftl@subfig	\cfttitlesecttrue . 17
665, 702, 745, 812 <u>690</u> , 778, 782	\cfttocbibindfalse 30
\cftasnumb . 18, 158,	\cftl@subtab	\cfttocbibindtrue . 30
<u>159</u> , 195, 251, <u>733</u> , 778, 782	\cfttocfinish
298, 343, 383,	\cftmakeZtitle . . . <u>893</u> <u>66</u> , 107,
423, 463, 564,	\cftmakeloftitle .	515, 617, 880, 890
666, 703, 746, 813 509, <u>519</u>	\cfttocstart <u>66</u> , 97,
\cftbsnum . 18, 158,	\cftmakelottitle .	505, 607, 872, 883
<u>159</u> , 193, 249, 611, <u>621</u>	\currentHref 1086, 1088
296, 341, 381,	\cftmaketocctitle .	\dotsep 3
421, 461, 562, 101, <u>110</u>	\dottedtocline 4
664, 701, 744, 811	\cftnctocfalse . . . 40	\@eha 799
\cftdobiblof . <u>491</u> , 511	\cftnctoctrue 41	\@firstoftwo 5
\cftdobiblot . <u>593</u> , 613	\cftpagestyle . 48,	\@highpenalty . 180, 240
\cftdobibtoc . . <u>80</u> , 103	117, 526, 628, 900	\@ifclassloaded
\cftdopartfalse . . 167	\cftpnumoff . . <u>972</u> , 13, 14, 15
\cftdoparttrue 170, 174	990, 992, 994,	\@ifpackagelater . . 32
\cfttemp 932, 933, 936,	996, 998, 1000,	\@ifpackageloaded .
938, 940, 942,	1002, 1004, 17, 30, 1084
944, 946, 948,	1006, 1008, 1010	\@ifundefined . 786, 790
950, 952, 988,	\cftpnumon . . . 1026,	\@namedef 801,
989, 992, 994,	1028, 1030,	840, 841, 842,
996, 998, 1000,	1032, 1034,	843, 844, 845,
1002, 1004,	1036, 1038,	846, 847, 848,
1006, 1008,	1040, 1042,	849, 860, 864,
1024, 1025,	1044, 1046, <u>1059</u>	867, 871, 882,
1028, 1030,	\cftquit <u>9</u> , 23, 27	893, 914, 916,

922, 924, 927, 928, 929, 973, 1060	\AtBeginDocument .. 17, 29, 94, 502, 604, 778, 780, 1083	\cftbeforesubparaskip 453, <u>468</u>
\@nameuse		\cftbeforesubsecskip 333, <u>348</u>
. 802, 803, 804, 807, 810, 815, 816, 821, 825, 826, 828, 829, 831, 832, 834, 835, 837, 838, 845, 850, 851, 878, 887, 896, 898, 902, 903, 905, 911, 912, 919, 920, 968, 969, 974, 1061, 1062	B \baselineskip 1069	\cftbeforesubsubsecskip 373, <u>388</u>
\@nobreakfalse 202, 206, 1100, 1112	C \c@lofdepth ... 553, 692 \c@lotdepth ... 655, 735 \c@tocdepth ... 168, 172, 239, 282, 332, 372, 412, 452 \c@X <u>786</u>	\cftbeforesubtabskip 736, <u>754</u>
\@nobreaktrue . 201, 205	\cftaddnumtitleline ... 16, <u>1081</u> , 1087	\cftbeforetabskip 656, <u>672</u>
\@pnumwidth .. 3, 141, 229, 274, 329, 369, 409, 449, 489, 586, 688, 729, 771, 851, 1062	\cftaddtitleline 16, <u>1079</u> , 1085	\cftbeforetoctitleskip 113, 115, <u>124</u>
\@restonecolfalse . 71	\cftafterloftitle 528, <u>543</u>	\cftbeforeXskip . 9, <u>820</u>
\@restonecoltrue .. 69	\cftafterloftitleskip 531, <u>534</u>	\cftbeforeZtitleskip 7, <u>908</u>
\@secondoftwo 7	\cftafterlottitle 630, <u>645</u>	\cftchapafterpnum . <u>257</u>
\@secpenalty 111, 182, 286, 520, 622, 894	\cftafterlottitleskip 633, <u>636</u>	\cftchapaftersnum 250, <u>257</u>
\@starttoc 105, 513, 615, 879, 888, <u>1092</u>	\cftafterlottitle 630, <u>645</u>	\cftchapaftersnumb 251, <u>257</u>
\@starttoc@latex@orig 1092, 1102	\cftafterlottitleskip 633, <u>636</u>	\cftchapdotsep <u>257</u>
\@tempa 146, 148	\cftaftertoctitle 119, <u>133</u>	\cftchapfillnum 254, <u>257</u>
\@tempb 147, 148	\cftaftertoctitleskip 122, <u>124</u>	\cftchapfont .. 253, <u>257</u>
\@tocextra 87, 498, 600	\cftafterZtitle . 7, <u>927</u>	\cftchapindent 242, 245, <u>257</u> , 1072
\@tocrmarg .. 3, 142, 187, 243, 290, 335, 375, 415, 455, 556, 658, 695, 738, 805, 1074	\cftafterZtitleskip 7, <u>908</u>	\cftchapleader <u>257</u>
A	\cftbeforechapskip 241, <u>257</u>	\cftchapname 933, <u>976</u> , 989, 1025
\addcontentsline 2, 84, 87, 495, 498, 597, 600	\cftbeforefigskip 554, <u>570</u>	\cftchapnumwidth 248, <u>257</u> , 1073
\addtocontents 4, 1071, 1078, 1079, 1081, 1085, 1087	\cftbeforeloftitleskip 522, 524, <u>534</u>	\cftchappagefont .. <u>257</u>
	\cftbeforelottitleskip 624, 626, <u>636</u>	\cftchappresnum 249, <u>257</u>
	\cftbeforeparaskip 413, <u>428</u>	\cftchapterprecis 16, <u>1065</u>
	\cftbeforepartskip 184, <u>212</u>	\cftchapterprecishere ... 16, 1066, <u>1068</u>
	\cftbeforesecskip 284, 287, <u>303</u>	\cftchapterprecistoc ... 16, 1067, <u>1071</u>
	\cftbeforesubfigskip 693, <u>712</u>	\cftdot 7, <u>144</u>
		\cftdotfill ... 144, 221, 268, 314, 322, 363, 403, 443, 483, 580, 682, 722, 764, 845
		\cftdotsep .. 8, <u>154</u> , 315, 364, 404, 444, 484, 581, 683, 723, 765, 844
		\cftfigafterpnum .. <u>570</u>

<code>\cftfigaftersnum</code> ..	<code>\cftparapresnum</code> 421 , 428	<code>\cftsubfigdotsep</code> .. 712
..... 563 , 570	<code>\cftparfillskip</code> 156 , 974	<code>\cftsubfigfillnum</code> ..
<code>\cftfigaftersnumb</code> .	<code>\cftparskip</code> .. 8 , 91 , 706 , 712
..... 564 , 570	100 , 508 , 610 , 886	<code>\cftsubfigfont</code> 705 , 712
<code>\cftfigdotsep</code>	<code>\cftpartafterpnum</code> . 212	<code>\cftsubfigindent</code> ..
570	<code>\cftpartaftersnum</code> 694 , 697 , 712
<code>\cftfigfillnum</code> 567 , 570 194 , 212	<code>\cftsubfigleader</code> .. 712
<code>\cftfigfont</code> ... 566 , 570	<code>\cftpartaftersnumb</code> .	<code>\cftsubfigname</code>
<code>\cftfigindent</code> 195 , 212	948 , 976 , 1004 , 1040
.... 555 , 558 , 570	<code>\cftpartdotsep</code>	<code>\cftsubfignumwidth</code> .
<code>\cftfigleader</code>	212 700 , 712
<code>\cftfigname</code>	<code>\cftpartfillnum</code> 198 , 212	<code>\cftsubfigpagefont</code> . 712
946 , 976 , 1002 , 1038	<code>\cftpartfont</code> .. 197 , 212	<code>\cftsubfigpresnum</code> .
<code>\cftfignumwidth</code> 561 , 570	<code>\cftpartindent</code> 701 , 712
<code>\cftfigpagefont</code> ... 570 186 , 189 , 212	<code>\cftsubparaafterpnum</code>
<code>\cftfigpresnum</code> 562 , 570	<code>\cftpartleader</code> ... 212 468
<code>\cftlocalchange</code> 16 , 1077	<code>\cftpartnumwidth</code> ..	<code>\cftsubparaaftersnum</code>
<code>\cftlofposthook</code> .. 1119 192 , 212 462 , 468
<code>\cftlofprehook</code> ... 1116	<code>\cftpartpagefont</code> .. 212	<code>\cftsubparaaftersnumb</code>
<code>\cftloftitlefont</code> ..	<code>\cftpartpresnum</code> 463 , 468
..... 528 , 543 193 , 197 , 212	<code>\cftsubparadotsep</code> . 468
<code>\cftlotposthook</code> .. 1120	<code>\cftpnumalign</code> 8 , 143 ,	<code>\cftsubparafillnum</code> .
<code>\cftlotprehook</code> ... 1117	229 , 274 , 329 , 466 , 468
<code>\cftlottitlefont</code> ..	369 , 409 , 449 ,	<code>\cftsubparafont</code> 465 , 468
..... 630 , 645	489 , 586 , 688 ,	<code>\cftsubparaindent</code> .
<code>\cftmarklof</code> 52 , 529	729 , 771 , 851 , 1062 454 , 457 , 468
<code>\cftmarklot</code> 52 , 631	<code>\cftsecafterpnum</code> .. 303	<code>\cftsubparaleader</code> . 468
<code>\cftmarktoc</code> 52 , 120	<code>\cftsecaftersnum</code> ..	<code>\cftsubparaname</code> ...
<code>\cftmarkZ</code> 297 , 303	944 , 976 , 1000 , 1036
<code>\cftmarkZ</code>	<code>\cftsecaftersnumb</code> .	<code>\cftsubparanumwidth</code>
<code>\cftnodots</code> .. 8 , 147 , 298 , 303 460 , 468
154 , 222 , 269 , 323	<code>\cftsecdotsep</code>	<code>\cftsubparapagefont</code> 468
<code>\cftpagenumbersoff</code> .	303	<code>\cftsubparapresnum</code> .
..... 12 , 987	<code>\cftsecfillnum</code> 301 , 303 461 , 468
<code>\cftpagenumberson</code> .	<code>\cftsecfont</code> ... 300 , 303	<code>\cftsubsecafterpnum</code>
..... 12 , 1023	<code>\cftsecindent</code> 348 , 369
<code>\cftparaafterpnum</code> . 428 289 , 292 , 303	<code>\cftsubsecaftersnum</code>
<code>\cftparaaftersnum</code> .	<code>\cftsecleader</code> 342 , 348
..... 422 , 428	<code>\cftsecname</code>	<code>\cftsubsecaftersnumb</code>
<code>\cftparaaftersnumb</code> .	936 , 976 , 992 , 1028 343 , 348
..... 423 , 428	<code>\cftsecnumwidth</code> 295 , 303	<code>\cftsubsecdotsep</code> .. 348
<code>\cftparadotsep</code> 428	<code>\cftsecpagefont</code> ... 303	<code>\cftsubsecfillnum</code> .
<code>\cftparafillnum</code> 426 , 428	<code>\cftsecpresnum</code> 296 , 303 346 , 367
<code>\cftparafont</code> .. 425 , 428	<code>\cftsetindents</code> . 10 , 931	<code>\cftsubsecfont</code> 345 , 348
<code>\cftparaindent</code>	<code>\cftsetpnumwidth</code> ..	<code>\cftsubsecindent</code> ..
.... 414 , 417 , 428 8 , 141 , 1078 334 , 337 , 348
<code>\cftparaleader</code> 428	<code>\cftsetrmarg</code> 8 , 141 , 1078	<code>\cftsubsecleader</code> ..
<code>\cftparaname</code>	<code>\cftsubfigafterpnum</code> 712 348 , 368
942 , 976 , 998 , 1034	<code>\cftsubfigaftersnum</code>	<code>\cftsubsecname</code>
<code>\cftparanumwidth</code> 702 , 712	938 , 976 , 994 , 1030
..... 420 , 428	<code>\cftsubfigaftersnumb</code>	
<code>\cftparapagefont</code> .. 428 703 , 712	

<code>\cftsubsecnumwidth</code> .	<code>\cfttabaftersnum</code> ..	E
..... 340 , 348 665 , 672	<code>\edef</code> 794
<code>\cftsubsecpagefont</code> .	<code>\cfttabaftersnumb</code> .	<code>\endcsname</code> 4 , 794 , 795 ,
..... 348 , 369 666 , 672	811 , 812 , 813 ,
<code>\cftsubsecpresnum</code> .	<code>\cfttabdotsep</code> 672	820 , 822 , 823 ,
..... 341 , 348	<code>\cfttabfillnum</code> 669 , 672	908 , 909 , 1097 ,
<code>\cftsubsubsecafterpnum</code>	<code>\cfttabfont</code> ... 668 , 672	1098 , 1104 ,
..... 388	<code>\cfttabindent</code>	1109 , 1110 , 1114
<code>\cftsubsubsecaftersnum</code> 657 , 660 , 672	<code>\endinput</code> 23
..... 382 , 388	<code>\cfttableader</code> 672	<code>\ext@Z</code> 860
<code>\cftsubsubsecaftersnumb</code>	<code>\cfttabname</code>	H
..... 383 , 388	950 , 976 , 1006 , 1042	<code>\hb@xt@</code> 158
<code>\cftsubsubsecdotsep</code> 388	<code>\cfttabnumwidth</code> 663 , 672	<code>\Huge</code> 134 ,
<code>\cftsubsubsecfillnum</code>	<code>\cfttabpagefont</code> ... 672	544 , 646 , 916 , 924
..... 386 , 388	<code>\cfttabpresnum</code> 664 , 672	I
<code>\cftsubsubsecfont</code> .	<code>\cfttocposthook</code> .. 1118	<code>\if@bibchapter</code>
..... 385 , 388	<code>\cfttocprehook</code> ... 1115 82 , 493 , 595
<code>\cftsubsubsecindent</code>	<code>\cfttoctitlefont</code> ..	<code>\if@cftdopart</code> 162
..... 374 , 377 , 388 119 , 133	<code>\if@cfthaschapter</code> 9 ,
<code>\cftsubsubsecleader</code> 388	<code>\cftXafterpnum</code> . 10 , 840	67 , 75 , 112 , 126 ,
<code>\cftsubsubsecname</code> .	<code>\cftXaftersnum</code> .. 9 , 840	133 , 169 , 173 ,
940 , 976 , 996 , 1032	<code>\cftXaftersnumb</code> . 9 , 840	179 , 200 , 237 ,
<code>\cftsubsubsecnumwidth</code>	<code>\cftXdotsep</code> 10 , 840	257 , 283 , 309 ,
..... 380 , 388	<code>\cftXfillnum</code> 849	352 , 392 , 432 ,
<code>\cftsubsubsecpagefont</code>	<code>\cftXfont</code> 9 , 840	472 , 521 , 536 ,
..... 388	<code>\cftXindent</code> 9 , 822	543 , 623 , 638 ,
<code>\cftsubsubsecpresnum</code>	<code>\cftXleader</code> 10 , 840	645 , 873 , 895 , 910
..... 381 , 388	<code>\cftXnumwidth</code> ... 9 , 822	<code>\if@cfthaspart</code>
<code>\cftsubtabafterpnum</code> 754	<code>\cftXpagefont</code> .. 10 , 840 163 , 165 , 212
<code>\cftsubtabaftersnum</code>	<code>\cftXpresnum</code> 9 , 840	<code>\if@cftkoma</code>
..... 745 , 754	<code>\cftZposthook</code> .. 17 , 928	11 , 58 , 135 , 138 ,
<code>\cftsubtabaftersnumb</code>	<code>\cftZprehook</code> ... 17 , 928	197 , 231 , 276 ,
..... 746 , 754	<code>\cftZtitlefont</code> .. 7 , 908	545 , 548 , 647 ,
<code>\cftsubtabdotsep</code> ..	<code>\chapter</code> 874	650 , 863 , 913 , 921
..... 722 , 754	<code>\contentsline</code> 2 , 1080 ,	<code>\if@cftnctoc</code>
<code>\cftsubtabfillnum</code> .	1082 , 1086 , 1088	40 , 95 , 503 , 605 , 870
..... 749 , 754	<code>\contentsname</code>	<code>\if@cftsubfigopt</code> ..
<code>\cftsubtabfont</code> 748 , 754	53 , 60 , 84 , 87 , 119 43 , 588 , 776 , 780
<code>\cftsubtabindent</code> ..	<code>\csname</code> . 4 , 794 , 795 ,	<code>\if@cfttitlesec</code> 16 , 197
..... 737 , 740 , 754	811 , 812 , 813 ,	<code>\if@cfttocbibind</code> ..
<code>\cftsubtableader</code> .. 754	820 , 822 , 823 , 28 , 102 , 510 , 612
<code>\cftsubtabname</code>	908 , 909 , 1097 ,	<code>\if@compatibility</code> . 204
952 , 976 , 1008 , 1044	1098 , 1104 ,	<code>\if@dotoclof</code> 492
<code>\cftsubtabnumwidth</code> .	1109 , 1110 , 1114	<code>\if@dotoclot</code> 594
..... 743 , 754	D	<code>\if@dotocloc</code> 81
<code>\cftsubtabpagefont</code> . 754	<code>\DeclareRobustCommand</code>	<code>\if@files</code> . 1096 , 1108
<code>\cftsubtabpresnum</code> 987 , 1023	<code>\if@restonecol</code> 76
..... 744 , 754		<code>\if@twocolumn</code> 68
<code>\cfttabafterpnum</code> .. 672		

<code>\immediate</code> .	1098, 1110		
<code>\interlinepenalty</code> .	118, 190, 246, 293, 338, 378, 418, 458, 527, 559, 629, 661, 698, 741, 808, 901		
L			
<code>\l@</code>	818		
<code>\l@chapter</code>	237		
<code>\l@figure</code>	552		
<code>\l@paragraph</code>	411		
<code>\l@part</code>	162		
<code>\l@section</code>	281		
<code>\l@subfigure</code>	691		
<code>\l@subparagraph</code>	451		
<code>\l@subsection</code>	331		
<code>\l@subsubsection</code>	371		
<code>\l@subtable</code>	734		
<code>\l@table</code>	654		
<code>\l@X</code>	14, 801		
<code>\Large</code>	137, 547, 649		
<code>\large</code>	217, 221, 223, 233, 234		
<code>\leaders</code>	151		
<code>\leftskip</code>	186, 196, 242, 252, 289, 299, 334, 344, 374, 384, 414, 424, 454, 464, 555, 565, 657, 667, 694, 704, 737, 747, 804, 814, 1072, 1073		
<code>\listfigurename</code>	55, 62, 495, 498, 528		
<code>\listoffigures</code>	6, 502		
<code>\listoftables</code>	6, 604		
<code>\listofX</code>	13, 870		
<code>\listtablename</code>	57, 64, 597, 600, 630		
<code>\lofdepth</code>	588		
<code>\lotdepth</code>	588		
M			
<code>\makeatletter</code>	1094, 1106		
<code>\makebox</code>	229, 274, 329, 369, 409, 449, 489, 586, 688, 729, 771, 851, 1062		
<code>\MakeUppercase</code>	53, 55, 57, 868		
<code>\MessageBreak</code>	34, 35, 36		
<code>\mkern</code>	151		
N			
<code>\newif</code>	10, 11, 16, 28, 40, 43, 162, 163		
<code>\newlistentry</code>	13, 785, 856, 858		
<code>\newlistof</code>	12, 854		
<code>\newwrite</code>	1097, 1109		
<code>\noexpand</code>	795		
<code>\normalfont</code>	134, 137, 313, 314, 316, 359, 363, 365, 399, 403, 405, 439, 443, 445, 479, 483, 485, 544, 547, 576, 580, 582, 646, 649, 678, 682, 684, 718, 722, 724, 760, 764, 766, 840, 845, 846, 916, 924		
<code>\numberline</code>	3, 157		
O			
<code>\onecolumn</code>	69		
<code>\openout</code>	1098, 1110		
P			
<code>\PackageError</code>	799		
<code>\PackageInfo</code>	24, 26		
<code>\PackageWarning</code>	21, 33, 790, 1122		
<code>\parindent</code>	100, 189, 245, 292, 337, 377, 417, 457, 508, 558, 610, 660, 697, 740, 807, 886		
<code>\parskip</code>	100, 508, 610, 886		
<code>\phantomsection</code>	78, 83, 86, 494, 497, 596, 599		
<code>\providecommand</code>	66, 74, 78, 144, 145, 154, 726, 768		
S			
<code>\sectfont</code>	135, 138, 233, 234, 277, 545, 548, 647, 650, 914, 922		
<code>\section</code>	876		
<code>\size@chapter</code>	135, 545, 647, 914		
<code>\size@section</code>	138, 548, 650, 922		
<code>\string</code>	1122		
T			
<code>\tableofcontents</code>	6, 94		
<code>\textit</code>	1070, 1075		
<code>\theX</code>	14, 786		
<code>\thispagestyle</code>	49		
<code>\toclevel@subfig</code>	712		
<code>\toclevel@subfigure</code>	692, 726		
<code>\toclevel@subtable</code>	735, 754		
<code>\toclevel@X</code>	848		
<code>\tocloftpagestyle</code>	6, 48		
<code>\twocolumn</code>	76		
Z			
<code>\Zdepth</code>	13, 860		