

The `tocloft` package*

Author: Peter Wilson, Herries Press

Maintainer: Will Robertson

`will dot robertson at latex-project dot org`

2017/08/29

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 v2.3h, last revised 2017/08/29.

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

`\@tocrmarg`
`\@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

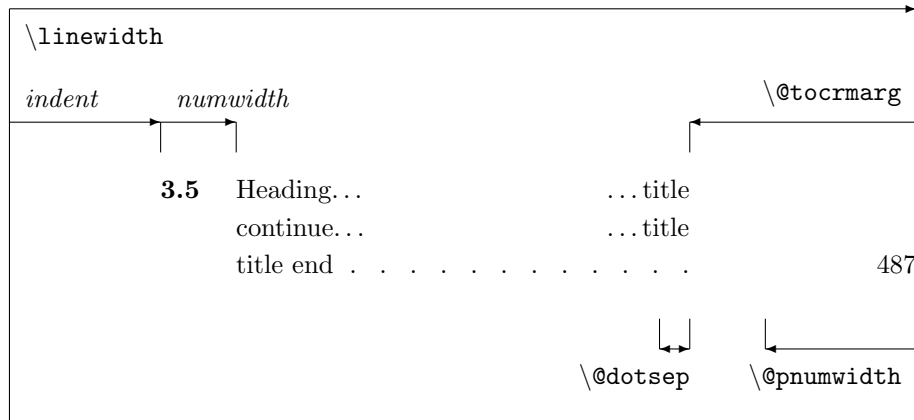


Figure 1: Layout of a ToC (LoF, LoT) entry

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 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{<seclvl>}{<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\nneedspace{2\baselineskip}}` to specify that if there is not a vertical space of two baselines left on the page then start a new page (the `\needspace` macro is defined in the `needspace` package).

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

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

2 The `tocloft` package

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

`\tableofcontents` The ToC, LoF, and LoT are printed at the point in the document where these
`\listoffigures` commands are called, as per normal \LaTeX . However, there is one difference be-
`\listoftables` tween 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`.

<code>\cftmarkZ</code>	These macros set the appearance of the running heads on the ToC, LoF, and LoT pages. You probably don’t need to change these.
<code>\cftbeforeZtitleskip</code>	These lengths control the vertical spacing before and after the titles. You can change them from their default values by using <code>\setlength</code> .
<code>\cftafterZtitleskip</code>	
<code>\cftZtitlefont</code>	The code used for typesetting the ToC title looks like
<code>\cftafterZtitle</code>	<code>{\cfttoctitlefont \contentsname}{\cftaftertoctitle}\par</code>

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
`\cftnodots` (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
`\cftsetrmarg` `\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.

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

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

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

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

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

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

```

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

- To eliminate the dots in the leader:

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

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

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

- To add a colon after the section number:

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

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

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

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

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

- To set the section numbers flushright:⁵

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

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

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

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

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

`\cftpagenumbersoff`
`\cftpagenumberson`

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

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

...

```

\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}{\cftpagenumberson{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}%
  {\normalfont\normalsize\itshape}%
  level
  indent from left margin
  skip above heading
  runin heading with 1em between title & text
  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} \langle title \rangle}{\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` Some packages, like the `float` package by Anselm Lingnau, enable the creation of
`\cftasnum` other kinds of *List of . . .*. The `tocloft` package is only minimally able to change the
`\cftasnumb` 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 `minitocs` in bold font. If `tocloft` is used in conjunction with `minitoc`, then the `minitoc` section entries are typeset in the normal font, except for the page numbers which are in bold font, while the ToC section entries are all in normal font.

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

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

Announce the name and version of the package, which requires L^AT_EX 2_ε but no extra packages.

```
1 \usc
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{tocloft}[2017/08/29 v2.3h parameterised ToC, etc., typesetting]
```

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

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

```
10 \newcommand{\@cftquit}{}
11 \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.
```

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

```

`\if@cfttitlesec`

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

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

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

Perhaps quit now.

```
28 \@cftquit
```

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

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

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

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

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

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

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

Process the options.

```
46
47 \ProcessOptions\relax
48
```

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

`\@cftpagestyle`

```

49 \newcommand{\tocloftpagestyle}[1]{%
50   \def\@cftpagestyle{\thispagestyle{#1}}
51 \tocloftpagestyle{plain}
52

```

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

`\cftmarklof`

`\cftmarklot`

```

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

```

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

`\@cfttocfinish`

```

67 \providecommand{\@cfttocstart}{%
68   \if@cfthaschapter
69     \if@twocolumn
70       \@restonecoltrue\onecolumn
71     \else
72       \@restonecolfalse
73     \fi
74   \fi}

```

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

```

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

```

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

```

79 \providecommand{\phantomsection}{}
80

```

`\@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`.

```

81 \newcommand{\@cftdobibtoc}{%
82   \if@dotoc
83     \if@bibchapter
84       \phantomsection
85       \addcontentsline{toc}{chapter}{\contentsname}
86     \else
87       \phantomsection
88       \addcontentsline{toc}{\@tocextra}{\contentsname}
89     \fi
90   \fi}
91

```

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

```

92 \newlength{\cftparskip}
93 \setlength{\cftparskip}{0pt}
94

```

`\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.

```

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

```

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

```

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

```

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

```

103     \if@cfttocbibind
104       \@cftdobibtoc
105     \fi

```

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

```

106     \@starttoc{toc}%
107   \endgroup
108   \@cfttocfinish}
109 \fi
110 }

```

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

```

111 \newcommand{\@cftmaketoctitle}{%
112   \addpenalty\@secpenalty
113   \if@cfthaschapter

```

```

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

```

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

`\cftaftertoctitleskip`

```

125 \newlength{\cftbeforetoctitleskip}
126 \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*`.

```

127 \if@cfthaschapter
128 \setlength{\cftbeforetoctitleskip}{50pt}
129 \setlength{\cftaftertoctitleskip}{40pt}
130 \else
131 \setlength{\cftbeforetoctitleskip}{3.5ex \@plus 1ex \@minus .2ex}
132 \setlength{\cftaftertoctitleskip}{2.3ex \@plus .2ex}
133 \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).

```

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

```

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

`\cftsetrmarg`

```

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

```

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

```

144 \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.

```

145 \providecommand{\cftdot}{.}
146 \providecommand{\cftdotfill}[1]{%
147   \def\@tempa{#1}%
148   \def\@tempb{\cftnodots}%
149   \ifx\@tempa\@tempb
150     \hfill
151   \else
152     \leaders\hbox{$\m@th\mkern #1 mu\hbox{\cftdot}\mkern #1 mu$}\hfill
153   \fi
154 }
```

`\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.)

```

155 \providecommand{\cftdotsep}{4.5}
156 \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`.

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

```
157 \newcommand{\cftparrfillskip}{\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).

```
158 \renewcommand{\numberline}[1]{%
159   \hbext@{\@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

`\@cftasnum`

`\@cftasnumb`

```
160 \newcommand{\@cftbsnum}{}
161 \newcommand{\@cftasnum}{}
162 \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@section` for an explanation of most of this code). By default, Parts (and Chapters) do not have dotted leaders. This package provides for all entries to have dotted leaders.

`\ifcftdopart`

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

```

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

```

183     \addpenalty\@secpenalty
184     \fi
185     \addvspace{\cftbeforepartskip}%
186     \beginingroup
187     {\leftskip \cftpartindent\relax
188     \rightskip \@tocrmarg
189     \parfillskip -\rightskip
190     \parindent \cftpartindent\relax\@afterindenttrue
191     \interlinepenalty\@M
192     \leavevmode
193     \@tempdima \cftpartnumwidth\relax
194     \let\@cftbsnum \cftpartpresnum
195     \let\@cftasnum \cftpartaftersnum
196     \let\@cftasnumb \cftpartaftersnumb
197     \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.

```

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

```

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

```

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

```

```

226 \newlength{\cftpartindent}
227 \setlength{\cftpartindent}{0em}
228 \newcommand{\cftpartfillnum}[1]{%
229   {\cftpartleader}%
230   {\makebox[\@pnumwidth][\cftpnumalign]{\cftpartpagefont #1}\cftpartafterpnum\par}%
231 }

```

koma classes use some different settings.

```

232 \if@cfstkoma
233   \setlength{\cftpartnumwidth}{2em}
234   \renewcommand{\cftpartfont}{\sectfont\large}
235   \renewcommand{\cftpartpagefont}{\sectfont\large}
236 \fi
237 \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.

```

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

```

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

```

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

```

```

267 \newcommand{\cftchapaftersnum}{ }
268 \newcommand{\cftchapaftersnumb}{ }
269 \newcommand{\cftchapleader}{\bfseries\cftdotfill{\cftchapdotsep}}
270 \newcommand{\cftchapdotsep}{\cftnodots}
271 \newcommand{\cftchappagefont}{\bfseries}
272 \newcommand{\cftchapafterpnum}{ }
273 \newcommand{\cftchapfillnum}[1]{%
274   {\cftchapleader}\nobreak
275   \makebox[\@pnumwidth][\cftpnumalign]{\cftchappagefont #1}\cftchapafterpnum\par
276 }

```

koma classes have different chapter settings.

```

277 \if@cftkoma
278   \renewcommand{\cftchapfont}{\sectfont}
279 \fi
280 \fi
281

```

`\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).

```

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

```

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

```

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

```

```

308 \newcommand{\cftsecaftersnum}{\z@ \plus.2\p@}
309 \newcommand{\cftsecaftersnumb}{\z@ \plus.2\p@}
310 \if@cfthaschapter
311   \setlength{\cftbeforesecskip}{\z@ \plus.2\p@}
312   \setlength{\cftsecindent}{1.5em}
313   \setlength{\cftsecnumwidth}{2.3em}
314   \newcommand{\cftsecfont}{\normalfont}
315   \newcommand{\cftsecleader}{\normalfont\cftdotfill{\cftsecdotsep}}
316   \newcommand{\cftsecdotsep}{\cftdotsep}
317   \newcommand{\cftsecpagefont}{\normalfont}
318 \else
319   \setlength{\cftbeforesecskip}{1.0em \plus\p@}
320   \setlength{\cftsecindent}{0em}
321   \setlength{\cftsecnumwidth}{1.5em}
322   \newcommand{\cftsecfont}{\bfseries}
323   \newcommand{\cftsecleader}{\bfseries\cftdotfill{\cftsecdotsep}}
324   \newcommand{\cftsecdotsep}{\cftnodots}
325   \newcommand{\cftsecpagefont}{\bfseries}
326 \fi
327 \newcommand{\cftsecafterpnum}{\z@}
328 \newcommand{\cftsecfillnum}[1]{\%
329   {\cftsecleader}\nobreak
330   \makebox[\@pnumwidth][\cftpnumalign]{\cftsecpagefont #1}\cftsecafterpnum\par
331 }

```

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

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

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

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

Add some vertical space.

```
334     \vskip \cftbeforesubsecskip
```

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

```
335     {\leftskip \cftsubsecindent\relax
```

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

```
336     \rightskip \@tocrmarg
```

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

```
337     \parfillskip -\rightskip
```

Set the paragraph indent to the entry's indentation.

```
338     \parindent \cftsubsecindent\relax\@afterindenttrue
```

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

```
339     \interlinepenalty\@M
```

Make sure that we have left vertical mode.

```
340 \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.

```
341 \@tempdima \cftsubsecnumwidth\relax
342 \let\@cftbsnum \cftsubsecpresnum
343 \let\@cftasnum \cftsubsecaftersnum
344 \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.

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

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

```
346 {\cftsubsecfont #1}\nobreak
```

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

```
347 \cftsubsecfillnum{#2}}%
348 \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 349 \newlength{\cftbeforesubsecskip}
\cftsubsecfont      350 \setlength{\cftbeforesubsecskip}{\z@ \@plus.2\p@}
\cftsubsecpresnum   351 \newlength{\cftsubsecindent}
\cftsubsecaftersnum 352 \newlength{\cftsubsecnumwidth}
\cftsubsecaftersnumb 353 \if@cfthaschapter
\cftsubsecleader     354 \setlength{\cftsubsecindent}{3.8em}
\cftsubsecdotsep     355 \setlength{\cftsubsecnumwidth}{3.2em}
\cftsubsecpagefont   356 \else
\cftsubsecafterpnum  357 \setlength{\cftsubsecindent}{1.5em}
358 \setlength{\cftsubsecnumwidth}{2.3em}
359 \fi
360 \newcommand{\cftsubsecfont}{\normalfont}
361 \newcommand{\cftsubsecpresnum}{}
362 \newcommand{\cftsubsecaftersnum}{}
363 \newcommand{\cftsubsecaftersnumb}{}
364 \newcommand{\cftsubsecleader}{\normalfont\cftdotfill{\cftsubsecdotsep}}
365 \newcommand{\cftsubsecdotsep}{\cftdotsep}
366 \newcommand{\cftsubsecpagefont}{\normalfont}
367 \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.

```
368 \newcommand{\cftsubsecfillnum}[1]{%
369 {\cftsubsecleader}\nobreak
```

```

370 \makebox[\@pnumwidth][\cftpnumalign]{\cftsubsecpagefont #1}\cftsubsecafterpnum\par
371 }

```

`\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).

```

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

```

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

```

\cftsubsubsecindent
\cftsubsubsecnumwidth
\cftsubsubsecfont
\cftsubsubsecpresnum
\cftsubsubsecaftersnum
\cftsubsubsecaftersnumb
\cftsubsubsecleader
\cftsubsubsecdotsep
\cftsubsubsecpagefont
\cftsubsubsecafterpnum
\cftsubsubsecfillnum
389 \newlength{\cftbeforesubsubsecskip}
390 \setlength{\cftbeforesubsubsecskip}{\z@ \@plus.2\p@}
391 \newlength{\cftsubsubsecindent}
392 \newlength{\cftsubsubsecnumwidth}
393 \if@cfthaschapter
394   \setlength{\cftsubsubsecindent}{7.0em}
395   \setlength{\cftsubsubsecnumwidth}{4.1em}
396 \else
397   \setlength{\cftsubsubsecindent}{3.8em}
398   \setlength{\cftsubsubsecnumwidth}{3.2em}
399 \fi
400 \newcommand{\cftsubsubsecfont}{\normalfont}
401 \newcommand{\cftsubsubsecpresnum}{}
402 \newcommand{\cftsubsubsecaftersnum}{}
403 \newcommand{\cftsubsubsecaftersnumb}{}
404 \newcommand{\cftsubsubsecleader}{\normalfont\cftdotfill{\cftsubsubsecdotsep}}
405 \newcommand{\cftsubsubsecdotsep}{\cftdotsep}
406 \newcommand{\cftsubsubsecpagefont}{\normalfont}
407 \newcommand{\cftsubsubsecafterpnum}{}
408 \newcommand{\cftsubsubsecfillnum}[1]{%
409   {\cftsubsubsecleader}\nobreak
410   \makebox[\@pnumwidth][\cftpnumalign]{\cftsubsubsecpagefont #1}\cftsubsubsecafterpnum\par
411 }

```

`\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).

```

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

```

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

```

\cftparanumwidth 429 \newlength{\cftbeforeparaskip}
\cftparafont      430 \setlength{\cftbeforeparaskip}{\z@ \@plus.2\p@}
\cftparapresnum  431 \newlength{\cftparaindent}
\cftparaaftersnum 432 \newlength{\cftparanumwidth}
\cftparaaftersnumb 433 \if@cfthaschapter
\cftparaleader    434 \setlength{\cftparaindent}{10em}
\cftparadotsep    435 \setlength{\cftparanumwidth}{5em}
\cftparapagefont  436 \else
\cftparaafterpnum 437 \setlength{\cftparaindent}{7.0em}
\cftparafillnum   438 \setlength{\cftparanumwidth}{4.1em}
                  439 \fi
                  440 \newcommand{\cftparafont}{\normalfont}
                  441 \newcommand{\cftparapresnum}{}
                  442 \newcommand{\cftparaaftersnum}{}
                  443 \newcommand{\cftparaaftersnumb}{}
                  444 \newcommand{\cftparaleader}{\normalfont\cftdotfill{\cftparadotsep}}
                  445 \newcommand{\cftparadotsep}{\cftdotsep}
                  446 \newcommand{\cftparapagefont}{\normalfont}
                  447 \newcommand{\cftparaafterpnum}{}
                  448 \newcommand{\cftparafillnum}[1]{%
                  449   {\cftparaleader}\nobreak
                  450   \makebox[\@pnumwidth][\cftpnumalign]{\cftparapagefont #1}\cftparaafterpnum\par
                  451 }

```

`\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).

```

452 \renewcommand*{\l@subparagraph}[2]{%
453   \ifnum \c@tocdepth >4\relax
454     \vskip \cftbeforesubparaskip
455     {\leftskip \cftsubparaindent\relax
456      \rightskip \@tocrmarg
457      \parfillskip -\rightskip
458      \parindent \cftsubparaindent\relax\@afterindenttrue
459      \interlinepenalty\@M
460      \leavevmode
461      \@tempdima \cftsubparanumwidth\relax
462      \let\@cftbsnum \cftsubparapresnum
463      \let\@cftasnum \cftsubparaaftersnum
464      \let\@cftasnumb \cftsubparaaftersnumb
465      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
466      {\cftsubparafont #1}\nobreak
467      \cftsubparafillnum{#2}}}%
468   \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 469 \newlength{\cftbeforesubparaskip}
\cftsubparafont      470 \setlength{\cftbeforesubparaskip}{\z@ \@plus.2\p@}
\cftsubparapresnum   471 \newlength{\cftsubparaindent}
\cftsubparaaftersnum 472 \newlength{\cftsubparanumwidth}
\cftsubparaaftersnumb 473 \if@cfthaschapter
\cftsubparaleader     474 \setlength{\cftsubparaindent}{12em}
\cftsubparadotsep     475 \setlength{\cftsubparanumwidth}{6em}
\cftsubparapagefont   476 \else
\cftsubparaafterpnum  477 \setlength{\cftsubparaindent}{10em}
\cftsubparafillnum    478 \setlength{\cftsubparanumwidth}{5em}
                     479 \fi
                     480 \newcommand{\cftsubparafont}{\normalfont}
                     481 \newcommand{\cftsubparapresnum}{\relax}
                     482 \newcommand{\cftsubparaaftersnum}{\relax}
                     483 \newcommand{\cftsubparaaftersnumb}{\relax}
                     484 \newcommand{\cftsubparaleader}{\normalfont\cftdotfill{\cftsubparadotsep}}
                     485 \newcommand{\cftsubparadotsep}{\cftdotsep}
                     486 \newcommand{\cftsubparapagefont}{\normalfont}
                     487 \newcommand{\cftsubparaafterpnum}{\relax}
                     488 \newcommand{\cftsubparafillnum}[1]{%
                     489   {\cftsubparaleader}\nobreak
                     490   \makebox[\@pnumwidth][\cftpnumalign]{\cftsubparapagefont #1}\cftsubparaafterpnum\par
                     491 }

```

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

```

492 \newcommand{\@cftdobiblof}{%
493   \if@dotoclof

```

```

494 \if@bibchapter
495 \phantomsection
496 \addcontentsline{toc}{chapter}{\listfigurename}
497 \else
498 \phantomsection
499 \addcontentsline{toc}{\@tocextra}{\listfigurename}
500 \fi
501 \fi}
502

```

`\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).

```

503 \AtBeginDocument{%
504 \if@cftnctoc\else
505 \renewcommand{\listoffigures}{%
506 \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.

```

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

```

Finally, restore any multicolumn typesetting.

```

516 \cfttocfinish}%
517 \fi
518 }
519

```

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

```

520 \newcommand{\@cftmakeloftitle}{%
521 \addpenalty\@secpenalty
522 \if@cfthaschapter
523 \vspace*{\cftbeforelofttitleskip}
524 \else
525 \vspace{\cftbeforelofttitleskip}
526 \fi
527 \cftpagestyle
528 {\interlinepenalty\@M
529 {\cftlofttitlefont\listfigurename}{\cftafterlofttitle}

```

```

530 \cftmarklof
531 \par\nobreak
532 \vskip \cftafterloftitleskip
533 \@afterheading}}
534

```

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

```

\cftafterlofttitleskip 535 \newlength{\cftbeforelofttitleskip}
536 \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*`.

```

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

```

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

```

544 \ifcfthaschapter
545 \newcommand{\cftlofttitlefont}{\normalfont\Huge\bfseries}
546 \ifcftkoma\renewcommand{\cftlofttitlefont}{\size@chapter\sectfont}\fi
547 \else
548 \newcommand{\cftlofttitlefont}{\normalfont\Large\bfseries}
549 \ifcftkoma\renewcommand{\cftlofttitlefont}{\size@section\sectfont}\fi
550 \fi
551 \newcommand{\cftafterlofttitle}{}
552

```

`\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).

```

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

```

```

564 \let\cftasnum \cftfigaftersnum
565 \let\cftasnumb \cftfigaftersnumb
566 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
567 {\cftfigfont #1}\nobreak
568 \cftfigfillnum{#2}}%
569 \fi
570 }

```

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

```

\cftfignumwidth 571 \newlength{\cftbeforefigskip}
\cftfigfont      572 \setlength{\cftbeforefigskip}{\z@ \@plus.2\p@}
\cftfigpresnum   573 \newlength{\cftfigindent}
\cftfigaftersnum 574 \setlength{\cftfigindent}{1.5em}
\cftfigaftersnumb 575 \newlength{\cftfignumwidth}
\cftfigleader     576 \setlength{\cftfignumwidth}{2.3em}
\cftfigdotsep    577 \newcommand{\cftfigfont}{\normalfont}
\cftfigpagefont  578 \newcommand{\cftfigpresnum}{\cftfigfont}
\cftfigafterpnum 579 \newcommand{\cftfigaftersnum}{\cftfigfont}
\cftfigfillnum   580 \newcommand{\cftfigaftersnumb}{\cftfigfont}
                  581 \newcommand{\cftfigleader}{\normalfont\cftdotfill{\cftfigdotsep}}
                  582 \newcommand{\cftfigdotsep}{\cftdotsep}
                  583 \newcommand{\cftfigpagefont}{\normalfont}
                  584 \newcommand{\cftfigafterpnum}{\cftfigfont}
                  585 \newcommand{\cftfigfillnum}[1]{%
                  586 {\cftfigleader}\nobreak
                  587 \makebox[\@pnumwidth][\cftpnumalign]{\cftfigpagefont #1}\cftfigafterpnum\par
                  588 }

```

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

```

589 \if@cftsubfigopt\else
590 \newcounter{lofdepth}\setcounter{lofdepth}{1}
591 \newcounter{lotdepth}\setcounter{lotdepth}{1}
592 \fi
593

```

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

```

594 \newcommand{\@cftdobiblot}{%
595 \if@dotoclot
596 \if@bibchapter
597 \phantomsection
598 \addcontentsline{toc}{chapter}{\listtablename}
599 \else
600 \phantomsection
601 \addcontentsline{toc}{\@tocextra}{\listtablename}
602 \fi
603 \fi}
604

```

`\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).

```
605 \AtBeginDocument{%
606 \if@cftnctoc\else
607 \renewcommand{\listoftables}{%
608   \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.

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

Finally, restore any multicolumn typesetting.

```
618   \@cfttocfinish}%
619 \fi
620 }
621
```

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

```
622 \newcommand{\@cftmakelottitle}{%
623   \addpenalty\@secpenalty
624   \if@cfthaschapter
625     \vspace*{\cftbeforelottitleskip}
626   \else
627     \vspace{\cftbeforelottitleskip}
628   \fi
629   \@cftpagestyle
630   {\interlinepenalty\M
631    {\cftlottitlefont\listtablename}{\cftafterlottitle}
632    \cftmarklot
633    \par\nobreak
634    \vskip \cftafterlottitleskip
635    \@afterheading}}
636
```

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

```
\cftafterlottitleskip 637 \newlength{\cftbeforelottitleskip}
638 \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*`.

```
639 \if@cfthaschapter
640   \setlength{\cftbeforelottitleskip}{50pt}
641   \setlength{\cftafterlottitleskip}{40pt}
642 \else
643   \setlength{\cftbeforelottitleskip}{3.5ex \@plus 1ex \@minus .2ex}
644   \setlength{\cftafterlottitleskip}{2.3ex \@plus .2ex}
645 \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).

```
646 \if@cfthaschapter
647   \newcommand{\cftlottitlefont}{\normalfont\Huge\bfseries}
648   \if@cfstkoma\renewcommand{\cftlottitlefont}{\size@chapter\sectfont}\fi
649 \else
650   \newcommand{\cftlottitlefont}{\normalfont\Large\bfseries}
651   \if@cfstkoma\renewcommand{\cftlottitlefont}{\size@section\sectfont}\fi
652 \fi
653 \newcommand{\cftafterlottitle}{}
654
```

`\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).

```
655 \renewcommand*{\l@table}[2]{%
656   \ifnum\c@lotdepth >\z@
657     \vskip \cftbeforetabskip
658     {\leftskip \cfttabindent\relax
659      \rightskip \@tocrmarg
660      \parfillskip -\rightskip
661      \parindent \cfttabindent\relax\@afterindenttrue
662      \interlinepenalty\M
663      \leavevmode
664      \@tempdima \cfttabnumwidth\relax
665      \let\@cftbsnum \cfttabpresnum
666      \let\@cftasnum \cfttabaftersnum
667      \let\@cftasnumb \cfttabaftersnumb
668      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
669      {\cfttabfont #1}\nobreak
670      \cfttabfillnum{#2}}%
671   \fi
672 }
```

`\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
```

```

673 \newlength{\cftbeforetabskip}
674 \setlength{\cftbeforetabskip}{\z@ \@plus.2\p@}
675 \newlength{\cfttabindent}
676 \setlength{\cfttabindent}{1.5em}
677 \newlength{\cfttabnumwidth}
678 \setlength{\cfttabnumwidth}{2.3em}
679 \newcommand{\cfttabfont}{\normalfont}
680 \newcommand{\cfttabpresnum}{}
681 \newcommand{\cfttabaftersnum}{}
682 \newcommand{\cfttabaftersnumb}{}
683 \newcommand{\cfttableader}{\normalfont\cftdotfill{\cfttabdotsep}}
684 \newcommand{\cfttabdotsep}{\cftdotsep}
685 \newcommand{\cfttabpagefont}{\normalfont}
686 \newcommand{\cfttabafterpnum}{}
687 \newcommand{\cfttabfillnum}[1]{%
688   {\cfttableader}\nobreak
689   \makebox[\@pnumwidth][\cftpnumalign]{\cfttabpagefont #1}\cfttabafterpnum\par
690 }

```

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

\l@subfigure \l@subfigure{\<title>}{\<page>} typesets the LoF entry for a subfigure caption
heading. It is essentially the same as the parameterised code for \l@figure except
that account has to be taken of lofdepth.
692 \renewcommand*{\l@subfigure}[2]{%
693   \ifnum \c@lofdepth > \toclevel@subfigure
694     \vskip \cftbeforesubfigskip
695     {\leftskip \cftsubfigindent\relax
696       \rightskip \@tocrmarg
697       \parfillskip -\rightskip
698       \parindent \cftsubfigindent\relax\@afterindenttrue
699       \interlinepenalty\@M
700       \leavevmode
701       \@tempdima \cftsubfignumwidth\relax
702       \let\@cftbsnum \cftsubfigpresnum
703       \let\@cftasnum \cftsubfigaftersnum
704       \let\@cftasnumb \cftsubfigaftersnumb
705       \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
706       {\cftsubfigfont ##1}\nobreak
707       \cftsubfigfillnum{##2}}%
708   \fi
709   }%

```

```

710 }
711

\@cftsetsubfig This command initialises the setup for subfigure captions in the LoF.
712 \newcommand{\@cftsetsubfig}{%

\cftbeforesubfigskip
\cftsubfigindent 713 \newlength{\cftbeforesubfigskip}
\cftsubfignumwidth 714 \setlength{\cftbeforesubfigskip}{\z@ \@plus.2\p@}
\cftsubfigfont 715 \newlength{\cftsubfigindent}
\cftsubfigpresnum 716 \setlength{\cftsubfigindent}{3.8em}
\cftsubfigaftersnum 717 \newlength{\cftsubfignumwidth}
\cftsubfigaftersnumb 718 \setlength{\cftsubfignumwidth}{2.5em}
\cftsubfigleader 719 \newcommand{\cftsubfigfont}{\normalfont}
\cftsubfigdotsep 720 \newcommand{\cftsubfigpresnum}{\cftdotsep}
\cftsubfigpagefont 721 \newcommand{\cftsubfigaftersnum}{\cftdotsep}
\cftsubfigafterpnum 722 \newcommand{\cftsubfigaftersnumb}{\cftdotsep}
\toclevel@subfig 723 \newcommand{\cftsubfigleader}{\normalfont\cftdotfill{\cftsubtabdotsep}}
\cftsubfigfillnum 724 \newcommand{\cftsubfigdotsep}{\cftdotsep}
725 \newcommand{\cftsubfigpagefont}{\normalfont}
726 \newcommand{\cftsubfigafterpnum}{\cftdotsep}
727 \providecommand{\toclevel@subfigure}{1}
728 \newcommand{\cftsubfigfillnum}[1]{%
729 {\cftsubfigleader}\nobreak
730 \makebox[\@pnumwidth][\cftpnumalign]{\cftsubfigpagefont ##1}\cftsubfigafterpnum\par
731 }

This is the end of \@cftsetsubfig.
732 }
733

\@cftl@subtab This code redefines the code for \l@subtable.
734 \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 ex-
cept account has to be taken of lotdepth.
735 \renewcommand*{\l@subtable}[2]{%
736 \ifnum \c@lotdepth > \toclevel@subtable
737 \vskip \cftbeforesubtabskip
738 {\leftskip \cftsubtabindent\relax
739 \rightskip \@tocrmarg
740 \parfillskip -\rightskip
741 \parindent \cftsubtabindent\relax\@afterindenttrue
742 \interlinepenalty\@M
743 \leavevmode
744 \@tempdima \cftsubtabnumwidth\relax
745 \let\@cftbsnum \cftsubtabpresnum
746 \let\@cftasnum \cftsubtabaftersnum

```



```

747 \let\cftasnumb \cftsubtabaftersnumb
748 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
749 {\cftsubtabfont ##1}\nobreak
750 \cftsubtabfillnum{##2}}%
751 \fi
752 }%
753 }

```

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

```

754 \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
\cftsubtabfont
\cftsubtabpresnum
\cftsubtabaftersnum
\cftsubtabaftersnumb
\cftsubtableader
\cftsubtabdotsep
\cftsubtabpagefont
\cftsubtabafterpnum
\toclevel@subtable
\cftsubtabfillnum
755 \newlength{\cftbeforesubtabskip}
756 \setlength{\cftbeforesubtabskip}{\z@ \@plus.2\p@}
757 \newlength{\cftsubtabindent}
758 \setlength{\cftsubtabindent}{3.8em}
759 \newlength{\cftsubtabnumwidth}
760 \setlength{\cftsubtabnumwidth}{2.5em}
761 \newcommand{\cftsubtabfont}{\normalfont}
762 \newcommand{\cftsubtabpresnum}{}
763 \newcommand{\cftsubtabaftersnum}{}
764 \newcommand{\cftsubtabaftersnumb}{}
765 \newcommand{\cftsubtableader}{\normalfont\cftdotfill{\cftsubtabdotsep}}
766 \newcommand{\cftsubtabdotsep}{\cftdotsep}
767 \newcommand{\cftsubtabpagefont}{\normalfont}
768 \newcommand{\cftsubtabafterpnum}{}
769 \providecommand{\toclevel@subtable}{1}
770 \newcommand{\cftsubtabfillnum}[1]{%
771 {\cftsubtableader}\nobreak
772 \makebox[\@pnumwidth][\cftpnumalign]{\cftsubtabpagefont ##1}\cftsubtabafterpnum\par
773 }

```

This is the end of `\cftsetsubtab`.

```

774 }
775

```

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.

```

776
777 \if@cftsubfigopt
778 \cftsetsubfig\cftsetsubtab
779 \AtBeginDocument{\cftl@subfig\cftl@subtab}
780 \fi
781 %% \AtBeginDocument{\if@cftsubfigopt
782 %% \cftsetsubfig\cftsetsubtab
783 %% \cftl@subfig\cftl@subtab
784 %% \fi}
785

```

3.2 New list of...

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

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

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

```
787 \ifundefined{c@#2}{%      check & set the counter
788   \ifx \@empty#1\relax
789     \newcounter{#2}
790   \else
791     \ifundefined{c@#1}{\PackageWarning{tocloft}%
792       {#1 has no counter for use as a 'within'}
793     \newcounter{#2}}%
794     {\newcounter{#2}[#1}%
795     \expandafter\edef\csname the#2\endcsname{%
796       \expandafter\noexpand\csname the#1\endcsname.\noexpand\arabic{#2}}}%
797   \fi
798   \setcounter{#2}{0}
799 }
800 {\PackageError{tocloft}{#2 has been previously defined}{\@eha}}
801
```

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

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

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

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

```
803 \ifnum \@nameuse{c@#3depth} > #4\relax
804 \vskip \@nameuse{cftbefore#2skip}
805 {\leftskip \@nameuse{cft#2indent}\relax
806 \rightskip \@tocrmarg
807 \parfillskip -\rightskip
808 \parindent \@nameuse{cft#2indent}\relax\@afterindenttrue
809 \interlinepenalty\@M
810 \leavevmode
811 \@tempdima \@nameuse{cft#2numwidth}\relax
812 \expandafter\let\expandafter\@cftbsnum\csname cft#2presnum\endcsname
813 \expandafter\let\expandafter\@cftasnum\csname cft#2aftersnum\endcsname
814 \expandafter\let\expandafter\@cftasnumb\csname cft#2aftersnumb\endcsname
815 \advance\leftskip\@tempdima \null\nobreak\hskip -\leftskip
816 {\@nameuse{cft#2font}##1}\nobreak
817 \@nameuse{cft#2fillnum}{##2}}%
818 \fi
819 } % end of \l@#2
820
```

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

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

\cftXindent

\cftXnumwidth

```
823 \expandafter\newlength\csname cft#2indent\endcsname
824 \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.

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

\cftXfont

And the remaining commands.

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

\toclevel@X

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

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

\cftXfillnum

Typeset the leader and page number.

```
850 \@namedef{cft#2fillnum}##1{%
851 {\@nameuse{cft#2leader}}\nobreak
852 \makebox[\@pnumwidth][\cftpnumalign]{\@nameuse{cft#2pagefont}##1}\@nameuse{cft#2afterpnum}
853 }
```

This ends the definition of `\newlistentry`.

```
854 }
```

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

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

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

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

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

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

`\cftmarkZ` The heading marks for the listing.

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

`\listofX` Typeset the listing title and entries.

```
871   \if@cftnctoc
```

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

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

Otherwise use the fully parameterised definition.

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

```

887     \parindent\z@ \parskip\cftparskip
888     \@nameuse{@cftmake#3title}
889     \@starttoc{#3}%
890     \endgroup
891     \@cfttocfinish}
892 \fi
893

```

`\@cftmakeZtitle` Typeset the title.

```

894 \namedef{@cftmake#3title}{%
895   \addpenalty\@secpenalty
896   \if@cfthaschapter
897     \vspace*{\@nameuse{cftbefore#3titleskip}}%
898   \else
899     \vspace{\@nameuse{cftbefore#3titleskip}}%
900   \fi
901   \@cftpagestyle
902   {\interlinepenalty\@M
903    {\@nameuse{cft#3titlefont}#4}{\@nameuse{cftafter#3title}}%
904    \@nameuse{cftmark#3}%
905    \par\nobreak
906    \vskip \@nameuse{cftafter#3titleskip}%
907    \@afterheading}}
908

```

`\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`

```

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

```

`\cftafterZtitle` Something to go after the title.

```

928   \namedef{cftafter#3title}{}

```

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

`\cftZposthook`

```

929 \namedef{cft#3prehook}{}
930 \namedef{cft#3posthook}{}

    This is the end of the definition of \newlistof.
931 }

```

`\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.

```

932 \newcommand{\cftsetindents}[3]{%
933   \def\@cftemp{#1}
934   \ifx\@cftemp\cftchapname
935     \@cftsetindents{chap}{#2}{#3}
936   \else
937     \ifx\@cftemp\cftsecname \cftsetindents{sec}{#2}{#3}
938     \else
939       \ifx\@cftemp\cftsubsecname \cftsetindents{subsec}{#2}{#3}
940       \else
941         \ifx\@cftemp\cftsubsubsecname \cftsetindents{subsubsec}{#2}{#3}
942         \else
943           \ifx\@cftemp\cftparaname \cftsetindents{para}{#2}{#3}
944           \else
945             \ifx\@cftemp\cftsubparaname \cftsetindents{subpara}{#2}{#3}
946             \else
947               \ifx\@cftemp\cftfigname \cftsetindents{fig}{#2}{#3}
948               \else
949                 \ifx\@cftemp\cftsubfigname \cftsetindents{subfig}{#2}{#3}
950                 \else
951                   \ifx\@cftemp\cfttabname \cftsetindents{tab}{#2}{#3}
952                   \else
953                     \ifx\@cftemp\cftsubtabname \cftsetindents{subtab}{#2}{#3}
954                     \else
955                       \@cftsetindents{#1}{#2}{#3}
956                     \fi
957                   \fi
958                 \fi
959               \fi
960             \fi
961           \fi
962         \fi
963       \fi
964     \fi
965   \fi
966 }
967
\@cftsetindents \cftsetindents{⟨X⟩}{⟨indent⟩}{⟨numwidth⟩} is the internal version of \cftsetindents,
where in this case ⟨X⟩ is the internal (shorthand) name of the entry.

```

```

968 \newcommand{\@cftsetindents}[3]{%
969   \setlength{\@nameuse{cft#1indent}}{#2}
970   \setlength{\@nameuse{cft#1numwidth}}{#3}
971 }
972

```

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.

```

973 \newcommand{\@cftpnumoff}[1]{%
974   \@namedef{cft#1fillnum}##1{%
975     \cftparfillskip\@nameuse{cft#1afterpnum}\par}}
976

```

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

```

\cftsubsubsecname 977 \newcommand*{\cftchapname}{chapter}
\cftparaname       978 \newcommand*{\cftsecname}{section}
\cftsubparaname    979 \newcommand*{\cftsubsecname}{subsection}
\cftfigname        980 \newcommand*{\cftsubsubsecname}{subsubsection}
\cftsubfigname     981 \newcommand*{\cftparaname}{paragraph}
\cfttabname        982 \newcommand*{\cftsubparaname}{subparagraph}
\cftsubtabname     983 \newcommand*{\cftfigname}{figure}
                  984 \newcommand*{\cftsubfigname}{subfigure}
                  985 \newcommand*{\cfttabname}{table}
                  986 \newcommand*{\cftsubtabname}{subtable}
                  987

```

`\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.

```

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

```

```

999         \ifx\@cfttemp\cftparaname \@cftpnumoff{para}
1000     \else
1001         \ifx\@cfttemp\cftsubparaname \@cftpnumoff{subpara}
1002     \else
1003         \ifx\@cfttemp\cftfigname \@cftpnumoff{fig}
1004     \else
1005         \ifx\@cfttemp\cftsubfigname \@cftpnumoff{subfig}
1006     \else
1007         \ifx\@cfttemp\cfttabname \@cftpnumoff{tab}
1008     \else
1009         \ifx\@cfttemp\cftsubtabname \@cftpnumoff{subtab}
1010     \else
1011         \@cftpnumoff{#1}
1012     \fi
1013 \fi
1014 \fi
1015 \fi
1016 \fi
1017 \fi
1018 \fi
1019 \fi
1020 \fi
1021 \fi
1022 }
1023

```

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

```

1024 \DeclareRobustCommand{\cftpagenumberon}[1]{%
1025     \def\@cfttemp{#1}
1026     \ifx\@cfttemp\cftchapname
1027         \@cftpnumon{chap}
1028     \else
1029         \ifx\@cfttemp\cftsecname \@cftpnumon{sec}
1030     \else
1031         \ifx\@cfttemp\cftsubsecname \@cftpnumon{subsec}
1032     \else
1033         \ifx\@cfttemp\cftsubsubsecname \@cftpnumon{subsubsec}
1034     \else
1035         \ifx\@cfttemp\cftparaname \@cftpnumon{para}
1036     \else
1037         \ifx\@cfttemp\cftsubparaname \@cftpnumon{subpara}
1038     \else
1039         \ifx\@cfttemp\cftfigname \@cftpnumon{fig}
1040     \else
1041         \ifx\@cfttemp\cftsubfigname \@cftpnumon{subfig}
1042     \else
1043         \ifx\@cfttemp\cfttabname \@cftpnumon{tab}
1044     \else
1045         \ifx\@cfttemp\cftsubtabname \@cftpnumon{subtab}

```



```

1046         \else
1047             \@cftpnumon{#1}
1048         \fi
1049     \fi
1050 \fi
1051 \fi
1052 \fi
1053 \fi
1054 \fi
1055 \fi
1056 \fi
1057 \fi
1058 }
1059

```

`\@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.

```

1060 \newcommand{\@cftpnumon}[1]{%
1061   \@namedef{cft#1fillnum}##1{%
1062     {\@nameuse{cft#1leader}}\nobreak
1063     \makebox[\@pnumwidth][\cftpnumalign]{\@nameuse{cft#1pagefont}##1}\@nameuse{cft#1afterpnum}
1064   }%
1065 }

```

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.

```

1066 \newcommand{\cftchapterprecis}[1]{%
1067   \cftchapterprecishere{#1}
1068   \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.

```

1069 \newcommand{\cftchapterprecishere}[1]{%
1070   \vspace*{-2\baselineskip}

```

Typeset its argument using italic font in a `quote` environment.

```

1071   \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.

1072 `\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.

1073 `{\leftskip \cftchapindent\relax`
 1074 `\advance\leftskip \cftchapnumwidth\relax`

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

1075 `\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.

1076 `\textit{#1}\protect\par}}}`
 1077

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

1078 `\newcommand{\cftlocalchange}[3]{%`
 1079 `\addtocontents{#1}{\protect\cftsetpnumwidth{#2} \protect\cftsetrmarg{#3}}}`

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

1080 `\newcommand{\cftaddtitleline}[4]{\addtocontents{#1}{%`
 1081 `\protect\contentsline{#2}{#3}{#4}}}`

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

1082 `\newcommand{\cftaddnumtitleline}[5]{\addtocontents{#1}{%`
 1083 `\protect\contentsline{#2}{\protect\numberline{#3}{#4}}{#5}}}`

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

1084 `\AtBeginDocument{%`
 1085 `\ifpackageloaded{hyperref}{%`
 1086 `\renewcommand{\cftaddtitleline}[4]{\addtocontents{#1}{%`
 1087 `\protect\contentsline{#2}{#3}{#4}{\@currentHref}}}`
 1088 `\renewcommand{\cftaddnumtitleline}[5]{\addtocontents{#1}{%`
 1089 `\protect\contentsline{#2}{\protect\numberline{#3}{#4}}{#5}{\@currentHref}}}`
 1090 `}{}`
 1091 `}`
 1092

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

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

- flowfram
- gmampulex (?), gmtypos
- hyperref, memhfixc
- ijmart
- scrartcl, scrbook, scrreprt
- 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.

```

1093 \def\@starttoc@latex@orig#1{%
1094   \begingroup
1095     \makeatletter
1096     \@input{\jobname.#1}%
1097     \if@filesw
1098       \expandafter\newwrite\csname tf@#1\endcsname
1099       \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
1100     \fi
1101     \@nobreakfalse
1102   \endgroup}
1103 \ifx\@starttoc\@starttoc@latex@orig
1104   \def\@starttoc#1{%

```

```

1105 \csname cft#1prehook\endcsname
1106 \begingroup
1107 \makeatletter
1108 \@input{\jobname.#1}%
1109 \if@files
1110 \expandafter\newwrite\csname tf@#1\endcsname
1111 \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
1112 \fi
1113 \@nobreakfalse
1114 \endgroup
1115 \csname cft#1posthook\endcsname}
1116 \newcommand\cfttocprehook{}
1117 \newcommand\cftlofprehook{}
1118 \newcommand\cftlotprehook{}
1119 \newcommand\cfttocposthook{}
1120 \newcommand\cftlofposthook{}
1121 \newcommand\cftlotposthook{}
1122 \else
1123 \PackageWarning{tocloft}{\string\starttoc\space has already been redefined; tocloft bailing}
1124 \fi

```

The end of this package.

```
1125 </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`)