

The `tocloft` package*

Peter Wilson
Catholic University of America
Now at `peter.r.wilson@boeing.com`

2000/02/11

Abstract

The `tocloft` package provides means of controlling the typographic design of the Table of Contents, List of Figures and List of Tables. Use of this package requires the `stdclsdv` package.

The package has been tested with the `tocbibind`, `minitoc`, `subfigure`, `float` and `fnycchap` packages, and maybe others.

Contents

1	Introduction	1
1.1	<code>L</code> A _T E _X 's methods	2
2	The <code>tocloft</code> package	5
2.1	Package options	5
2.2	Changing the titles	5
2.3	Typesetting the entries	6
2.4	Experimental utilities	10
2.5	Usage with other packages	11
3	The package code	12
3.1	Support for the <code>subfigure</code> package	29
3.2	Experimental utilities	31

List of Figures

1	Layout of a ToC (LoF, LoT) entry	2
---	--	---

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.

*This file has version number v1.1, last revised 2000/02/11.

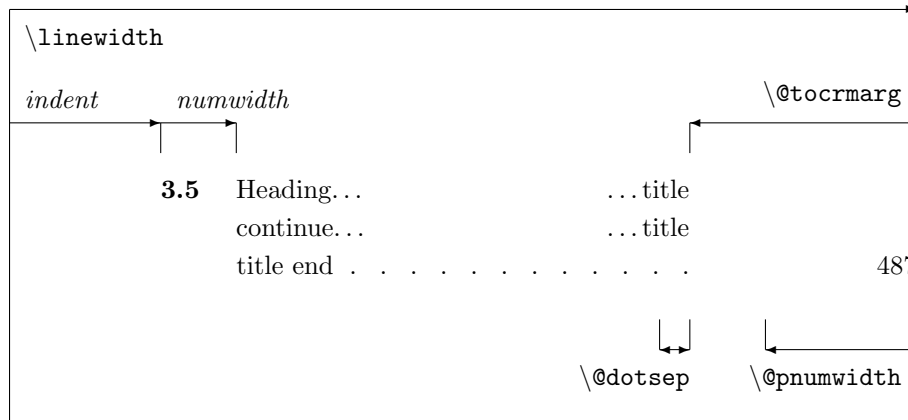


Figure 1: Layout of a ToC (LoF, LoT) entry

Elements of the package were developed as part of a class and package bundle for typesetting ISO standards [Wil96]. This manual is typeset according to the conventions of the L^AT_EX DOCSTRIP utility which enables the automatic extraction of the L^AT_EX macro source files [GMS94].

Section 2 describes the usage of the package. Commented source code for the package is in Section 3. The package requires the `stdclsdv` package.

The package has been tested in combination with at least the `tocbibind` package, the `minitoc` package, the `subfigure` package (version 2.1), the `algorithm` package (which, in turn, calls the `float` package) and the `fncychap` 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 L^AT_EX's methods

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

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

`\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>`, L^AT_EX provides a command `\l@kind{<title>}{<page>}` which performs the actual typesetting of the `\contentsline` entry.

`\@pnumwidth` `\@toctrmarg` `\@dotsep` The general layout of a typeset entry is illustrated in Figure 1. There are three internal L^AT_EX commands that are used in the typesetting. The page number is

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

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

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

`\@dottedtocline`

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

`\@dottedtocline{<secllevel>}{<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}`

³There are 18mu to 1em.

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
```

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

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

2 The tocloft package

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

```
\tableofcontents
\listoffigures
\listoftables
```

The ToC, LoF, and LoT are printed at the point in the document where these commands are called, as per normal L^AT_EX. However, there is one difference between the standard L^AT_EX behaviour and the behaviour with the tocloft package. In the standard L^AT_EX 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
...
```

2.1 Package options

The package takes two options, namely subfigure and titles. The subfigure option is required if the tocloft and subfigure packages are being used together. These two packages can be specified in any order.

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 ‘X’ to stand for ‘toc’ or ‘lof’ or ‘lot’. For example, \cftmarkX stands for \cftmarktoc or \cftmarklof or \cftmarklot.

\cftmarkX These macros set the appearance of the running heads on the ToC, LoF, and

LoT pages. You probably don't need to change these.

`\cftbeforeXtitleskip` These lengths control the vertical spacing before and after the titles. You can change them from their default values by using `\setlength`.

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

```
\cftXtitlefont
\cftafterXtitle
{\cfttoctitlefont \contentsname}{\cftaftertocitle}\par
```

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

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

```
\renewcommand{\cftafterXtitle}{%
  \\\[\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 `\cftafterXtitleskip`).

2.3 Typesetting the entries

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

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

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

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

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

`\cftsetpnumwidth` The page numbers are typeset in a fixed width box. The command `\cftsetpnumwidth{\length}` can be used to change the width of the box (L^AT_EX's

internal `\@pnumwidth`). The title texts will end before reaching the righthand margin. `\cftsetrmarg{<length>}` can be used to set this distance (L^AT_EX's internal `\@tocrmarg`). Note that the length used in `\cftsetrmarg` should be greater than the length set in `\cftsetpnumwidth`. These values should remain constant in any given document.

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`. NOTE: This command is not provided for `part` as their entries will have no indentation (they are the highest level document divisions).

`\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\cftaftersnum\hfil}
```

is typeset within a box of width `\cftXnumwidth`. Note also that for a Part entry the unnumbered style is used, whether or not the Part has a number.

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

`\cftXpresnum`
`\cftXaftersnum`
`\cftXaftersnumb` The section number is typeset within a box of width `\cftXnumwidth`. Within the box the macro `\cftXpresnum` is first called, then the number is typeset, 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.

`\cftXleader`
`\cftXdotsep` `\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`.

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}

```

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 title 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 title instead of at the righthand margin:

```

\renewcommand{\cftXleader}{\}
\renewcommand{\cftXafterpnum}{\cftparfillskip}

```

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

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

By default the `\parfillskip` value is locally set to fill up the last line of a paragraph. Just changing `\cftXleader` puts horrible interword spaces into the last line of the title. The `\cftparfillskip` command⁶ is part of the `tocloft` package and is provided just so that the above effect can be achieved.

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. This is one way of using `tocloft` to do this (assuming that the document has `\chapters`).

```
...
\appendix
\addtocontents{toc}{%
  % ensure no leaders
  \protect\renewcommand{\protect\cftchapleader}{} %
  % ensure last title line is ragged right
  \protect\renewcommand{\protect\cftchapafterpnum}{\cftparfillskip}
  % gobble the page number
  \protect\renewcommand{\protect\cftchappagefont}[1]{}
\chapter{First appendix}
```

If there are other headings to go into the ToC after the appendices, then it will probably be necessary to do a similar `\addtocontents` restoring the renewed commands to their pre-appendices state.

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:

```
\renewcommand{\cftsubfigdotsep}{\cftnodots}
\renewcommand{\cftsubfigpagefont}[1]{}
\renewcommand{\cftsubfigafterpnum}{\cftparfillskip}
```

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 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,⁷ 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` The `\cftchapterprecis` command calls these two commands to print the text in the document (the `\...here{<text>}` command) and to put it into the ToC (the `\...toc{<text>}` command). These can be used individually if required.

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

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

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

Any fragile commands used in the arguments must be protected.

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

`\contentsline{<kind>}{\numberline{<num> title}{<page>}`

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`.

2.5 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:

⁶Is there a better name for this?

⁷For example, Robert Sedgewick, *Algorithms*, Addison-Wesley, 1983.

- The 1998/11/15 version of `tocbibind` may give surprising results if the `\toctocname`, `\toclotname` 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 other kinds of *List of . . .*. The `tocloft` package is only minimally able to change the formatting of these, principally because the packages are independent of each other and, in the case of the `float` package, new kinds of float environments and their associated lists can be created on the fly at any point in a document. Some aspects of the typesetting are controlled by `\@cftbsnum`, `\@cftasnum` and `\@cftasnumb` commands. These are equivalent to the `\cftXpresnum`, `\cftXaftersnum` and `\cftXaftersnumb` commands described earlier. By default they are defined to do nothing, but may be renewed to do something.

The `tocloft` and `minitoc` packages have an unfortunate interaction,⁸ which fortunately can be fixed. In the normal course of events, when `minitoc` is used in a chaptered document it will typeset section entries in the `minitocs` in bold font. If `tocloft` is used in conjunction with `minitoc`, then the `minitoc` section entries are typeset in the normal font, except for the page numbers which are in bold font, while the ToC section entries are all in normal font.

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

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

in the preamble.

Otherwise, the cure is the following incantation:

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

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

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

3 The package code

Announce the name and version of the package, which requires L^AT_EX 2_ε and the `stdclsdv` package.

⁸Discovered by Lyndon Dudding (lyndon.dudding@totalise.co.uk).

```

1 (*usc)
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{tocloft}[2000/02/11 v1.1 parameterised ToC, etc., typesetting]
4 \RequirePackage{stdclsdv}

```

`\PRWPackageNote` and `\PRWPackageNoteNoLine` These two commands write a Package Note to the terminal and the log file. Use as: `\PRWPackageNote{<package name>}{<note text>}`. The NoLine version does not show the line number. The commands are intermediate between the kernel `\PackageWarning` and `\PackageInfo` commands. I have `\provided` the `\PRW...` commands as other packages (of mine) may also incorporate them. The code is based on `lterror.dtx`.

```

5 \providecommand{\PRWPackageNote}[2]{%
6   \GenericWarning{%
7     (#1)\@spaces\@spaces\@spaces\@spaces
8   }{%
9     Package #1 Note: #2%
10  }%
11 }
12 \providecommand{\PRWPackageNoteNoLine}[2]{%
13   \PRWPackageNote{#1}{#2@gobble}%
14 }

```

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

The `stdclsdv` package is used to determine what sectioning commands may be used in the document. We assume that if the class is known and it has `\chapter` commands, then it is either a book or report class; if it has `\section` commands then it is an article class or one of its derivatives.

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

```

15 \ifSCDnodivs
16   \PackageWarning{tocloft}%
17     {I don't recognize any sectional divisions so I'll do nothing}
18   \renewcommand{\SCDquit}{\endinput}
19 \fi

```

Perhaps quit now.

```
20 \SCDquit
```

Issue a warning if the class is unknown. Use chapter style if `\chapter` is defined, else use the `\section` style if it is defined, otherwise quit.

```

21 \ifSCDknownclass\else
22   \PackageWarning{tocloft}%
23     {I don't recognize the class but I'll do my best}
24   \ifSCDnodivs
25     \PackageWarning{tocloft}%
26       {I don't recognize any sectional divisions so I'll do nothing}
27     \renewcommand{\SCDquit}{\endinput}
28   \else
29     \ifSCDchapter
30       \PRWPackageNoteNoLine{tocloft}{The document class has chapter divisions}
31     \else
32       \ifSCDsection

```

```

33     \PRWPackageNoteNoLine{tocloft}{The document class has section divisions}
34     \else
35     \PackageWarning{tocloft}%
36     {The class has neither chapters nor sections, so I'll do nothing}
37     \renewcommand{\SCDquit}{\endinput}
38     \fi
39     \fi
40     \fi
41 \fi
42 \SCDquit

```

`\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`.

```

43 \newif\if@cfttocbibind
44 \AtBeginDocument{%
45   \ifpackageloaded{tocbibind}{\@cfttocbibindtrue}{\@cfttocbibindfalse}
46   \if@cfttocbibind
47     \ifpackageolder{tocbibind}{1998/11/16}{\@cfttocbibindfalse}
48     \PackageWarning{tocloft}{%
49 You are using a version of the tocbibind package\MessageBreak
50 that is not compatible with tocloft.\MessageBreak
51 The results may be surprising.\MessageBreak
52 Consider installing the current version of tocbibind.}}
53   \fi
54 }

```

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

```

55 \newif\if@cftnctoc\@cftnctocfalse
56 \DeclareOption{titles}{\@cftnctoctrue}
57 %% \ProcessOptions\relax

```

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

```

58 \newcommand{\cftmarktoc}{%
59   \@mkboth{\MakeUppercase\contentsname}{\MakeUppercase\contentsname}}
60 \newcommand{\cftmarklof}{%
61   \@mkboth{\MakeUppercase\listfigurename}{\MakeUppercase\listfigurename}}
62 \newcommand{\cftmarklot}{%
63   \@mkboth{\MakeUppercase\listtablename}{\MakeUppercase\listtablename}}

```

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

```

64 \newcommand{\@cfttocstart}{%
65   \ifSCDchapter
66     \if@twocolumn
67       \@restonecoltrue\onecolumn
68     \else
69       \@restonecolfalse
70     \fi
71   \fi}

```

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

```
72 \newcommand{\cfttocfinish}{%
73   \ifSCDchapter
74   \if@restonecol\twocolumn\fi
75   \fi}
```

`\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`.

```
76 \newcommand{\cftdobibtoc}{%
77   \if@dotoc
78   \if@bibchapter
79     \addcontentsline{toc}{chapter}{\contentsname}
80   \else
81     \addcontentsline{toc}{\@tocextra}{\contentsname}
82   \fi
83   \fi}
```

`\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.

```
84 \AtBeginDocument{%
85 \if@cftnctoc\else
86   \renewcommand{\tableofcontents}{%
87     \cfttocstart
```

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

```
88   \par
89   \begingroup
90   \parindent\z@ \parskip\z@
91   \cftmaketoctitle
```

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

```
92   \if@cfttocbibind
93     \cftdobibtoc
94   \fi
```

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

```
95   \@starttoc{toc}%
96   \endgroup
97   \cfttocfinish}
98 \fi
99 }
```

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

```
100 \newcommand{\cftmaketoctitle}{%
101   \vspace*{\cftbeforetoctitleskip}
102   \interlinepenalty\@M
103   {\cfttoctitlefont\contentsname}{\cftaftertoctitle}
104   \cftmarktoc
105   \par\nobreak
106   \vskip \cftaftertoctitleskip
107   \@afterheading}
```

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

`\cftaftertoctitleskip`

```

108 \newlength{\cftbeforetoctitleskip}
109 \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*`.

```

110 \ifSCDchapter
111   \setlength{\cftbeforetoctitleskip}{50pt}
112   \setlength{\cftaftertoctitleskip}{40pt}
113 \else
114   \setlength{\cftbeforetoctitleskip}{3.5ex \@plus 1ex \@minus .2ex}
115   \setlength{\cftaftertoctitleskip}{2.3ex \@plus .2ex}
116 \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).

```

117 \ifSCDchapter
118   \newcommand{\cfttoctitlefont}{\normalfont\Huge\bfseries}
119 \else
120   \newcommand{\cfttoctitlefont}{\normalfont\Large\bfseries}
121 \fi
122 \newcommand{\cftaftertoctitle}{}

```

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

`\cftsetrmarg`

```

123 \newcommand{\cftsetpnumwidth}[1]{\renewcommand{\@pnumwidth}{#1}}
124 \newcommand{\cftsetrmarg}[1]{\renewcommand{\@tocrmarg}{#1}}

```

`\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`.

```

125 \newcommand{\cftdot}{.}
126 \newcommand{\cftdotfill}[1]{%
127   \leaders\hbox{$\m@th\mkern #1 mu\hbox{\cftdot}\mkern #1 mu$}\hfill}

```

`\cftdotsep` `\cftdotsep` holds the default dot separation. If the kerns in `\cftdotfill` are

`\cftnodots` large enough, then no dots will be printed. `\cftnodots` should be ‘large enough’.

```

128 \newcommand{\cftdotsep}{4.5}
129 \newcommand{\cftnodots}{10000}

```

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

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

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

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

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

`\numberline` The purpose of the `\numberline{\langle secnum \rangle}` command is to typeset `\langle secnum \rangle` 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).

```
131 \renewcommand{\numberline}[1]{%
```

```
132   \hb@xt@\@tempdima{\@cftbsnum #1\@cftasnum\hfil}\@cftasnumb}
```

`\@cftbsnum` Originally these were not defined but were `\let` to appropriate commands in the `\l@...` commands, but they have to be defined in case something unexpected calls `\numberline`, for example through use of the `float` package.⁹

`\@cftasnum`
`\@cftasnumb`

```
133 \newcommand{\@cftbsnum}{}
134 \newcommand{\@cftasnum}{}
135 \newcommand{\@cftasnumb}{}

```

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

`\if@cftdopart`

```
136 \newif\if@cftdopart
137 \ifSCDpart
138 \renewcommand*{\l@part}[2]{%
139   \@cftdopartfalse
140   \ifnum \c@tocdepth >-2\relax
141     \ifSCDchapter
142       \@cftdoparttrue
143     \fi

```

⁹This bug was discovered by Andrew Thurber when using the `tocloft` and `algorithm` packages together.

```

144 \ifnum \c@tocdepth >\m@ne
145 \ifSCDchapter\else
146 \cftdoparttrue
147 \fi
148 \fi
149 \fi
150 \if@cftdopart
151 \ifSCDchapter
152 \addpenalty{-\@highpenalty}%
153 \else
154 \addpenalty\@secpenalty
155 \fi
156 \addvspace{\cftbeforepartskip}%
157 \@tempdima \cftpartnumwidth\relax
158 \let\@cftbsnum \cftpartpresnum
159 \let\@cftasnum \cftpartaftersnum
160 \let\@cftasnumb \cftpartaftersnumb
161 \begingroup
162 \parindent \z@ \rightskip \@pnumwidth
163 \parfillskip -\@pnumwidth
164 {\leavevmode
165 {\cftpartfont #1}{\cftpartleader}
166 {\hb@xt@\@pnumwidth{\hss {\cftpartpagefont #2}}}\cftpartafterpnum}\par
167 \nobreak
168 \ifSCDchapter
169 \global\@nobreaktrue
170 \everypar{\global\@nobreakfalse\everypar{}}%
171 \else
172 \if@compatibility
173 \global\@nobreaktrue
174 \everypar{\global\@nobreakfalse\everypar{}}%
175 \fi
176 \fi
177 \endgroup
178 \fi}
179 \fi

```

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

```

\cftpartnumwidth
\cftpartfont 180 \newlength{\cftbeforepartskip}
\cftpartpresnum 181 \setlength{\cftbeforepartskip}{2.25em \@plus\p@}
\cftpartaftersnum 182 \newlength{\cftpartnumwidth}
\cftpartaftersnumb 183 \setlength{\cftpartnumwidth}{3em}
\cftpartleader 184 \newcommand{\cftpartfont}{\large\bfseries}
\cftpartdotsep 185 \newcommand{\cftpartpresnum}{}
\cftpartpagefont 186 \newcommand{\cftpartaftersnum}{}
\cftpartafterpnum 187 \newcommand{\cftpartaftersnumb}{}
188 \newcommand{\cftpartleader}{\large\bfseries\cftdotfill{\cftpartdotsep}}
189 \newcommand{\cftpartdotsep}{\cftnodots}
190 \newcommand{\cftpartpagefont}{\large\bfseries}
191 \newcommand{\cftpartafterpnum}{}

```

`\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.

```

192 \ifSCDchapter
193 \renewcommand*{\l@chapter}[2]{%
194   \ifnum \c@tocdepth >\m@ne
195     \addpenalty{-\@highpenalty}%
196     \vskip \cftbeforechapskip
197     {\leftskip \cftchapindent\relax
198       \rightskip \@tocmarg
199       \parfillskip -\rightskip
200       \parindent \cftchapindent\relax\@afterindenttrue
201       \interlinepenalty\@M
202       \leavevmode
203       \@tempdima \cftchapnumwidth\relax
204       \let\@cftbsnum \cftchappresnum
205       \let\@cftasnum \cftchapaftersnum
206       \let\@cftasnumb \cftchapaftersnumb
207       \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
208       {\cftchapfont #1}\nobreak
209       {\cftchapleader}
210       \nobreak
211       \hb@xt@\@pnumwidth{\hfil\cftchappagefont #2}\cftchapafterpnum\par}%
212   \fi}
213 \fi

```

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

```

\cftchapnumwidth 214 \newlength{\cftbeforechapskip}
\cftchapfont      215 \setlength{\cftbeforechapskip}{1.0em \@plus\p@}
\cftchappresnum  216 \newlength{\cftchapindent}
\cftchapaftersnum 217 \setlength{\cftchapindent}{0em}
\cftchapaftersnumb 218 \newlength{\cftchapnumwidth}
\cftchapleader    219 \setlength{\cftchapnumwidth}{1.5em}
\cftchapdotsep    220 \newcommand{\cftchapfont}{\bfseries}
\cftchappagefont  221 \newcommand{\cftchappresnum}{}
\cftchapafterpnum 222 \newcommand{\cftchapaftersnum}{}
                  223 \newcommand{\cftchapaftersnumb}{}
                  224 \newcommand{\cftchapleader}{\bfseries\cftdotfill{\cftchapdotsep}}
                  225 \newcommand{\cftchapdotsep}{\cftnodots}
                  226 \newcommand{\cftchappagefont}{\bfseries}
                  227 \newcommand{\cftchapafterpnum}{}

```

`\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).

```

228 \renewcommand*{\l@section}[2]{%
229   \ifnum \c@tocdepth >\z@
230     \ifSCDchapter
231       \vskip \cftbeforesecskip
232     \else
233       \addpenalty\@secpenalty
234       \addvspace{\cftbeforesecskip}
235     \fi
236     {\leftskip \cftsecindent\relax

```

```

237 \rightskip \@tocrmarg
238 \parfillskip -\rightskip
239 \parindent \cftsecindent\relax\@afterindenttrue
240 \interlinepenalty\@M
241 \leavevmode
242 \@tempdima \cftsecnumwidth\relax
243 \let\@cftbsnum \cftsecpresnum
244 \let\@cftasnum \cftsecaftersnum
245 \let\@cftasnumb \cftsecaftersnumb
246 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
247 {\cftsecfont #1}\nobreak
248 {\cftsecleader}
249 \nobreak
250 \hb@xt@\@pnumwidth{\hfil\cftsecpagefont #2}\cftsecafterpnum\par}%
251 \fi}

```

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

```

\cftsecindent
\cftsecnumwidth 252 \newlength{\cftbeforesecskip}
\cftsecfont 253 \ifSCDchapter
\cftsecpresnum 254 \setlength{\cftbeforesecskip}{\z@ \@plus.2\p@}
\cftsecaftersnum 255 \else
\cftsecaftersnumb 256 \setlength{\cftbeforesecskip}{1.0em \@plus\p@}
\cftsecleader 257 \fi
\cftsecdotsep 258 \newlength{\cftsecindent}
\cftsecpagefont 259 \ifSCDchapter
\cftsecafterpnum 260 \setlength{\cftsecindent}{1.5em}
261 \else
262 \setlength{\cftsecindent}{0em}
263 \fi
264 \newlength{\cftsecnumwidth}
265 \ifSCDchapter
266 \setlength{\cftsecnumwidth}{2.3em}
267 \else
268 \setlength{\cftsecnumwidth}{1.5em}
269 \fi
270 \ifSCDchapter
271 \newcommand{\cftsecfont}{\normalfont}
272 \else
273 \newcommand{\cftsecfont}{\bfseries}
274 \fi
275 \newcommand{\cftsecpresnum}{}
276 \newcommand{\cftsecaftersnum}{}
277 \newcommand{\cftsecaftersnumb}{}
278 \ifSCDchapter
279 \newcommand{\cftsecleader}{\normalfont\cftdotfill{\cftsecdotsep}}
280 \else
281 \newcommand{\cftsecleader}{\bfseries\cftdotfill{\cftsecdotsep}}
282 \fi
283 \ifSCDchapter
284 \newcommand{\cftsecdotsep}{\cftdotsep}
285 \else
286 \newcommand{\cftsecdotsep}{\cftnodots}
287 \fi

```

```

288 \ifSCDchapter
289 \newcommand{\cftsecpagefont}{\normalfont}
290 \else
291 \newcommand{\cftsecpagefont}{\bfseries}
292 \fi
293 \newcommand{\cftsecafterpnum}{}

\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).
294 \renewcommand*{\l@section}[2]{%
    Only typeset the entry if it falls within the tocdepth.
295 \ifnum \c@tocdepth >\@ne
    Add some vertical space.
296 \vskip \cftbeforesubsecskip
    Start a group to keep paragraphing changes local. Set the \leftskip to the entry's indentation.
297 {\leftskip \cftsubsecindent\relax
    Set the \rightskip to \@tocrmarg to leave room for the page number.
298 \rightskip \@tocrmarg
    Ensure that the last line of the entry will be filled. Setting \parfillskip to a negative number prevents any overfull box messages.
299 \parfillskip -\rightskip
    Set the paragraph indent to the entry's indentation.
300 \parindent \cftsubsecindent\relax\@afterindenttrue
    Try and prevent breaks between lines in a multiple line entry.
301 \interlinepenalty\@M
    Make sure that we have left vertical mode.
302 \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.
303 \@tempdima \cftsubsecnumwidth\relax
304 \let\@cftbsnum \cftsubsecpresnum
305 \let\@cftasnum \cftsubsecaftersnum
306 \let\@cftasnumb \cftsubsecaftersnum
    Arrange that the (section number and) first line of the title is set at the current indent, and any further lines are further indented.
307 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
    Print the (number and) title, prohibiting any breaking.
308 {\cftsubsecfont #1}\nobreak
    Print the leader between the title and the page number, again asking for no breaks.
309 {\cftsubsecleader}
310 \nobreak

```

Finally, set the page number flushright in a box of width \@pnumwidth, not forgetting to finish the paragraph and close the group.

```
311     \hb@xt@\@pnumwidth{\hfil\cftsubsecpagefont #2}\cftsubsecafterpnum\par}%
312     \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 313 \newlength{\cftbeforesubsecskip}
\cftsubsecfont      314 \setlength{\cftbeforesubsecskip}{\z@ \@plus.2\p@}
\cftsubsecpresnum   315 \newlength{\cftsubsecindent}
\cftsubsecftersnum  316 \ifSCDchapter
\cftsubsecaftersnum 317 \setlength{\cftsubsecindent}{3.8em}
\cftsubsecleader    318 \else
\cftsubsecdotsep    319 \setlength{\cftsubsecindent}{1.5em}
\cftsubsecpagefont  320 \fi
\cftsubsecafterpnum 321 \newlength{\cftsubsecnumwidth}
                    322 \ifSCDchapter
                    323 \setlength{\cftsubsecnumwidth}{3.2em}
                    324 \else
                    325 \setlength{\cftsubsecnumwidth}{2.3em}
                    326 \fi
                    327 \newcommand{\cftsubsecfont}{\normalfont}
                    328 \newcommand{\cftsubsecpresnum}{}
                    329 \newcommand{\cftsubsecaftersnum}{}
                    330 \newcommand{\cftsubsecaftersnumb}{}
                    331 \newcommand{\cftsubsecleader}{\normalfont\cftdotfill{\cftsubsecdotsep}}
                    332 \newcommand{\cftsubsecdotsep}{\cftdotsep}
                    333 \newcommand{\cftsubsecpagefont}{\normalfont}
                    334 \newcommand{\cftsubsecafterpnum}{}

```

`\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).

```
335 \renewcommand*{\l@subsubsection}[2]{%
336   \ifnum \c@tocdepth >\tw@
337     \vskip \cftbeforesubsubsecskip
338     {\leftskip \cftsubsubsecindent\relax
339      \rightskip \@tocrmarg
340      \parfillskip -\rightskip
341      \parindent \cftsubsubsecindent\relax\@afterindenttrue
342      \interlinepenalty\@M
343      \leavevmode
344      \@tempdima \cftsubsubsecnumwidth\relax
345      \let\@cftbsnum \cftsubsubsecpresnum
346      \let\@cftasnum \cftsubsubsecaftersnum
347      \let\@cftasnumb \cftsubsubsecaftersnumb
348      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
349      {\cftsubsubsecfont #1}\nobreak
350      {\cftsubsubsecleader}
351      \nobreak
352      \hb@xt@\@pnumwidth{\hfil\cftsubsubsecpagefont #2}\cftsubsubsecafterpnum\par}%
353     \fi}
```

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

```

\cftsubsubsecnumwidth 354 \newlength{\cftbeforesubsubsecskip}
\cftsubsubsecfont      355 \setlength{\cftbeforesubsubsecskip}{\z@ \@plus.2\p@}
\cftsubsubsecpresnum   356 \newlength{\cftsubsubsecindent}
\cftsubsubsecaftersnum 357 \ifSCDchapter
\cftsubsubsecaftersnumb 358 \setlength{\cftsubsubsecindent}{7.0em}
\cftsubsubsecleader     359 \else
\cftsubsubsecdotsep     360 \setlength{\cftsubsubsecindent}{3.8em}
\cftsubsubsecpagefont   361 \fi
\cftsubsubsecafterpnum 362 \newlength{\cftsubsubsecnumwidth}
                        363 \ifSCDchapter
                        364 \setlength{\cftsubsubsecnumwidth}{4.1em}
                        365 \else
                        366 \setlength{\cftsubsubsecnumwidth}{3.2em}
                        367 \fi
                        368 \newcommand{\cftsubsubsecfont}{\normalfont}
                        369 \newcommand{\cftsubsubsecpresnum}{\relax}
                        370 \newcommand{\cftsubsubsecaftersnum}{\relax}
                        371 \newcommand{\cftsubsubsecaftersnumb}{\relax}
                        372 \newcommand{\cftsubsubsecleader}{\normalfont\cftdotfill{\cftsubsubsecdotsep}}
                        373 \newcommand{\cftsubsubsecdotsep}{\cftdotsep}
                        374 \newcommand{\cftsubsubsecpagefont}{\normalfont}
                        375 \newcommand{\cftsubsubsecafterpnum}{\relax}

```

`\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).

```

376 \renewcommand*{\l@paragraph}[2]{%
377 \ifnum \c@tocdepth >3\relax
378 \vskip \cftbeforeparaskip
379 {\leftskip \cftparaindent\relax
380 \rightskip \@tocrmarg
381 \parfillskip -\rightskip
382 \parindent \cftparaindent\relax\@afterindenttrue
383 \interlinepenalty\@M
384 \leavevmode
385 \@tempdima \cftparanumwidth\relax
386 \let\cftbsnum \cftparapresnum
387 \let\cftasnum \cftparaaftersnum
388 \let\cftasnumb \cftparaaftersnumb
389 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
390 {\cftparafont #1}\nobreak
391 {\cftparaleader}
392 \nobreak
393 \hb@xt@\@pnumwidth{\hfil\cftparapagefont #2}\cftparaafterpnum\par}%
394 \fi}

```

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

```

\cftparanumwidth 395 \newlength{\cftbeforeparaskip}
\cftparafont      396 \setlength{\cftbeforeparaskip}{\z@ \@plus.2\p@}
\cftparapresnum   397 \newlength{\cftparaindent}
\cftparaaftersnum
\cftparaaftersnumb
\cftparaleader
\cftparadotsep
\cftparapagefont
\cftparaafterpnum

```

```

398 \ifSCDchapter
399   \setlength{\cftparaindent}{10em}
400 \else
401   \setlength{\cftparaindent}{7.0em}
402 \fi
403 \newlength{\cftparanumwidth}
404 \ifSCDchapter
405   \setlength{\cftparanumwidth}{5em}
406 \else
407   \setlength{\cftparanumwidth}{4.1em}
408 \fi
409 \newcommand{\cftparafont}{\normalfont}
410 \newcommand{\cftparapresnum}{}
411 \newcommand{\cftparaaftersnum}{}
412 \newcommand{\cftparaaftersnumb}{}
413 \newcommand{\cftparaleader}{\normalfont\cftdotfill{\cftparadotsep}}
414 \newcommand{\cftparadotsep}{\cftdotsep}
415 \newcommand{\cftparapagefont}{\normalfont}
416 \newcommand{\cftparaafterpnum}{}

```

`\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).

```

417 \renewcommand*{\l@subparagraph}[2]{%
418   \ifnum \c@tocdepth >4\relax
419     \vskip \cftbeforesubparaskip
420     {\leftskip \cftsubparaindent\relax
421      \rightskip \@tocrmarg
422      \parfillskip -\rightskip
423      \parindent \cftsubparaindent\relax\@afterindenttrue
424      \interlinepenalty\@M
425      \leavevmode
426      \@tempdima \cftsubparanumwidth\relax
427      \let\@cftbsnum \cftsubparapresnum
428      \let\@cftasnum \cftsubparaaftersnum
429      \let\@cftasnumb \cftsubparaaftersnumb
430      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
431      {\cftsubparafont #1}\nobreak
432      {\cftsubparaleader}
433      \nobreak
434      \hb@xt@\@pnumwidth{\hfil\cftsubparapagefont #2}\cftsubparaafterpnum\par}%
435   \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` 436 `\newlength{\cftbeforesubparaskip}`
`\cftsubparafont` 437 `\setlength{\cftbeforesubparaskip}{\z@ \@plus.2\p@}`
`\cftsubparapresnum` 438 `\newlength{\cftsubparaindent}`
`\cftsubparaaftersnum` 439 `\ifSCDchapter`
`\cftsubparaaftersnumb` 440 `\setlength{\cftsubparaindent}{12em}`
`\cftsubparaleader` 441 `\else`
`\cftsubparadotsep` 442 `\setlength{\cftsubparaindent}{10em}`
`\cftsubparapagefont` 443 `\fi`
`\cftsubparaafterpnum` 444 `\newlength{\cftsubparanumwidth}`


```

445 \ifSCDchapter
446   \setlength{\cftsubparanumwidth}{6em}
447 \else
448   \setlength{\cftsubparanumwidth}{5em}
449 \fi
450 \newcommand{\cftsubparafont}{\normalfont}
451 \newcommand{\cftsubparapresnum}{}
452 \newcommand{\cftsubparaaftersnum}{}
453 \newcommand{\cftsubparaaftersnumb}{}
454 \newcommand{\cftsubparaleader}{\normalfont\cftdotfill{\cftsubparadotsep}}
455 \newcommand{\cftsubparadotsep}{\cftdotsep}
456 \newcommand{\cftsubparapagefont}{\normalfont}
457 \newcommand{\cftsubparaafterpnum}{}

```

`\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`.

```

458 \newcommand{\cftdobiblof}{%
459   \if@dotoclof
460     \if@bibchapter
461       \addcontentsline{toc}{chapter}{\listfigurename}
462     \else
463       \addcontentsline{toc}{\@tocextra}{\listfigurename}
464     \fi
465   \fi}

```

`\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).

```

466 \AtBeginDocument{
467 \ifcftnctoc\else
468 \renewcommand{\listoffigures}{%
469   \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.

```

470   \par
471   \begingroup
472     \parindent\z@ \parskip\z@
473     \@cftmakeloftitle
474     \ifcfttocbibind
475       \cftdobiblof
476     \fi
477     \@starttoc{lof}%
478   \endgroup

```

Finally, restore any multicolumn typesetting.

```

479   \@cfttocfinish}
480 \fi
481 }

```

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

```

482 \newcommand{\@cftmakelofttitle}{%
483   \vspace*{\cftbeforelofttitleskip}
484   \interlinepenalty\@M
485   {\cftlofttitlefont\listfigurename}{\cftafterlofttitle}
486   \cftmarklof
487   \par\nobreak
488   \vskip \cftafterlofttitleskip
489   \@afterheading}

```

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

```

\cftafterlofttitleskip 490 \newlength{\cftbeforelofttitleskip}
491 \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*`.

```

492 \ifSCDchapter
493   \setlength{\cftbeforelofttitleskip}{50pt}
494   \setlength{\cftafterlofttitleskip}{40pt}
495 \else
496   \setlength{\cftbeforelofttitleskip}{3.5ex \@plus 1ex \@minus .2ex}
497   \setlength{\cftafterlofttitleskip}{2.3ex \@plus .2ex}
498 \fi

```

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

```

499 \ifSCDchapter
500   \newcommand{\cftlofttitlefont}{\normalfont\Huge\bfseries}
501 \else
502   \newcommand{\cftlofttitlefont}{\normalfont\Large\bfseries}
503 \fi
504 \newcommand{\cftafterlofttitle}{}

```

`\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).

```

505 \renewcommand*{\l@figure}[2]{%
506   \vskip \cftbeforefigskip
507   {\leftskip \cftfigindent\relax
508     \rightskip \@tocrmarg
509     \parfillskip -\rightskip
510     \parindent \cftfigindent\relax\@afterindenttrue
511     \interlinepenalty\@M
512     \leavevmode
513     \@tempdima \cftfignumwidth\relax
514     \let\@cftbsnum \cftfigpresnum
515     \let\@cftasnum \cftfigaftersnum
516     \let\@cftasnumb \cftfigaftersnumb
517     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
518     {\cftfigfont #1}\nobreak
519     {\cftfigleader}
520     \nobreak

```

```

521     \hb@xt@\@pnumwidth{\hfil\cftfigpagefont #2}\cftfigafterpnum\par}%
522   }

```

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

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

```

\cftfignumwidth 523 \newlength{\cftbeforefigskip}
\cftfigfont      524 \setlength{\cftbeforefigskip}{\z@ \@plus.2\p@}
\cftfigpresnum  525 \newlength{\cftfigindent}
\cftfigaftersnum 526 \setlength{\cftfigindent}{1.5em}
\cftfigaftersnumb 527 \newlength{\cftfignumwidth}
\cftfigleader    528 \setlength{\cftfignumwidth}{2.3em}
\cftfigdotsep    529 \newcommand{\cftfigfont}{\normalfont}
\cftfigpagefont  530 \newcommand{\cftfigpresnum}{ }
\cftfigafterpnum 531 \newcommand{\cftfigaftersnum}{ }
\cftfigafterpnum 532 \newcommand{\cftfigaftersnumb}{ }
533 \newcommand{\cftfigleader}{\normalfont\cftdotfill{\cftfigdotsep}}
534 \newcommand{\cftfigdotsep}{\cftdotsep}
535 \newcommand{\cftfigpagefont}{\normalfont}
536 \newcommand{\cftfigafterpnum}{ }

```

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

```

537 \newcommand{\@cftdobiblot}{%
538   \if@dotoclot
539     \if@bibchapter
540       \addcontentsline{toc}{chapter}{\listtablename}
541     \else
542       \addcontentsline{toc}{\@tocextra}{\listtablename}
543     \fi
544   \fi}

```

`\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).

```

545 \AtBeginDocument{
546 \if@cftnctoc\else
547 \renewcommand{\listoftables}{%
548 \@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.

```

549 \par
550 \begingroup
551 \parindent\z@ \parskip\z@
552 \@cftmakelottitle
553 \if@cfttocbibind
554 \@cftdobiblot
555 \fi
556 \@starttoc{lot}%
557 \endgroup

```

Finally, restore any multicolumn typesetting.

```
558 \cfttocfinish}
559 \fi
560 }
```

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

```
561 \newcommand{\cftmakelottitle}{%
562 \vspace*{\cftbeforelottitleskip}
563 \interlinepenalty\@M
564 {\cftlottitlefont\listtablename}{\cftafterlottitle}
565 \cftmarklot
566 \par\nobreak
567 \vskip \cftafterlottitleskip
568 \@afterheading}
```

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

```
\cftafterlottitleskip 569 \newlength{\cftbeforelottitleskip}
570 \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*`.

```
571 \ifSCDchapter
572 \setlength{\cftbeforelottitleskip}{50pt}
573 \setlength{\cftafterlottitleskip}{40pt}
574 \else
575 \setlength{\cftbeforelottitleskip}{3.5ex \@plus 1ex \@minus .2ex}
576 \setlength{\cftafterlottitleskip}{2.3ex \@plus .2ex}
577 \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).

```
578 \ifSCDchapter
579 \newcommand{\cftlottitlefont}{\normalfont\Huge\bfseries}
580 \else
581 \newcommand{\cftlottitlefont}{\normalfont\Large\bfseries}
582 \fi
583 \newcommand{\cftafterlottitle}{}
```

`\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).

```
584 \renewcommand*{\l@table}[2]{%
585 \vskip \cftbeforetabskip
586 {\leftskip \cfttabindent\relax
587 \rightskip \@tocrmarg
588 \parfillskip -\rightskip
589 \parindent \cfttabindent\relax\@afterindenttrue
590 \interlinepenalty\@M
591 \leavevmode
592 \@tempdima \cfttabnumwidth\relax
```

```

593 \let\@cftbsnum \cfttabpresnum
594 \let\@cftasnum \cfttabaftersnum
595 \let\@cftasnumb \cfttabaftersnumb
596 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
597 {\cfttabfont #1}\nobreak
598 {\cfttableader}
599 \nobreak
600 \hb@xt@\@pnumwidth{\hfil\cfttabpagefont #2}\cfttabafterpnum\par}%
601 }

```

`\cftbeforetabskip` These are the user commands to control the typesetting of Table caption entries.

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

```

\cfttabnumwidth 602 \newlength{\cftbeforetabskip}
\cfttabfont 603 \setlength{\cftbeforetabskip}{\z@ \@plus.2\p@}
\cfttabpresnum 604 \newlength{\cfttabindent}
\cfttabaftersnum 605 \setlength{\cfttabindent}{1.5em}
\cfttabaftersnumb 606 \newlength{\cfttabnumwidth}
\cfttableader 607 \setlength{\cfttabnumwidth}{2.3em}
\cfttabdotsep 608 \newcommand{\cfttabfont}{\normalfont}
\cfttabpagefont 609 \newcommand{\cfttabpresnum}{}
\cfttabafterpnum 610 \newcommand{\cfttabaftersnum}{}
611 \newcommand{\cfttabaftersnumb}{}
612 \newcommand{\cfttableader}{\normalfont\cftdotfill{\cfttabdotsep}}
613 \newcommand{\cfttabdotsep}{\cftdotsep}
614 \newcommand{\cfttabpagefont}{\normalfont}
615 \newcommand{\cfttabafterpnum}{}

```

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.

```

616 \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`.

```

617 \renewcommand*{\l@subfigure}[2]{%
618 \ifnum \c@lofdepth > 1\relax
619 \vskip \cftbeforesubfigskip
620 {\leftskip \cftsubfigindent\relax
621 \rightskip \@tocrmarg
622 \parfillskip -\rightskip
623 \parindent \cftsubfigindent\relax\@afterindenttrue
624 \interlinepenalty\@M
625 \leavevmode
626 \@tempdima \cftsubfignumwidth\relax
627 \let\@cftbsnum \cftsubfigpresnum
628 \let\@cftasnum \cftsubfigaftersnum
629 \let\@cftasnumb \cftsubfigaftersnumb
630 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip

```

```

631     {\cftsubfigfont ##1}\nobreak
632     {\cftsubfigleader}
633     \nobreak
634     \hb@xt@{\cftsubfigwidth}{\hfil\cftsubfigpagefont ##2}\cftsubfigafterpnum\par}%
635     \fi
636   }
637 }

```

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

```

638 \newcommand{\@cftsetsubfig}{

```

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

```

\cftsubfigindent
\cftsubfignumwidth 639 \newlength{\cftbeforesubfigskip}
\cftsubfigfont      640 \setlength{\cftbeforesubfigskip}{\z@ \@plus.2\p@}
\cftsubfigpresnum  641 \newlength{\cftsubfigindent}
\cftsubfigaftersnum 642 \setlength{\cftsubfigindent}{3.8em}
\cftsubfigaftersnumb 643 \newlength{\cftsubfignumwidth}
\cftsubfigleader    644 \setlength{\cftsubfignumwidth}{2.5em}
\cftsubfigdotsep    645 \newcommand{\cftsubfigfont}{\normalfont}
\cftsubfigpagefont  646 \newcommand{\cftsubfigpresnum}{}
\cftsubfigafterpnum 647 \newcommand{\cftsubfigaftersnum}{}
648 \newcommand{\cftsubfigaftersnumb}{}
649 \newcommand{\cftsubfigleader}{\normalfont\cftdotfill{\cftsubfigdotsep}}
650 \newcommand{\cftsubfigdotsep}{\cftdotsep}
651 \newcommand{\cftsubfigpagefont}{\normalfont}
652 \newcommand{\cftsubfigafterpnum}{}

```

This is the end of `\@cftsetsubfig`.

```

653 }
654

```

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

```

655 \newcommand{\@cftl@subtab}{

```

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

```

656 \renewcommand*{\l@subtable}[2]{%
657   \ifnum \c@lotdepth > 1\relax
658     \vskip \cftbeforesubtabskip
659     {\leftskip \cftsubtabindent\relax
660     \rightskip \@tocrmarg
661     \parfillskip -\rightskip
662     \parindent \cftsubtabindent\relax\@afterindenttrue
663     \interlinepenalty\@M
664     \leavevmode
665     \@tempdima \cftsubtabnumwidth\relax
666     \let\@cftbsnum \cftsubtabpresnum
667     \let\@cftasnum \cftsubtabaftersnum
668     \let\@cftasnumb \cftsubtabaftersnumb
669     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
670     {\cftsubtabfont ##1}\nobreak

```

```

671     {\cftsubtableader}
672     \nobreak
673     \hb@xt@\@pnumwidth{\hfil\cftsubtabpagefont ##2}\cftsubtabafterpnum\par}%
674 \fi
675 }
676 }

```

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

```
677 \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 678 \newlength{\cftbeforesubtabskip}
\cftsubtabfont      679 \setlength{\cftbeforesubtabskip}{\z@ \@plus.2\p@}
\cftsubtabpresnum   680 \newlength{\cftsubtabindent}
\cftsubtabaftersnum 681 \setlength{\cftsubtabindent}{3.8em}
\cftsubtabaftersnumb 682 \newlength{\cftsubtabnumwidth}
\cftsubtableader    683 \setlength{\cftsubtabnumwidth}{2.5em}
\cftsubtabdotsep    684 \newcommand{\cftsubtabfont}{\normalfont}
\cftsubtabpresnum   685 \newcommand{\cftsubtabpresnum}{}
\cftsubtabpagefont  686 \newcommand{\cftsubtabaftersnum}{}
\cftsubtabafterpnum 687 \newcommand{\cftsubtabaftersnumb}{}
688 \newcommand{\cftsubtableader}{\normalfont\cftdotfill{\cftsubtabdotsep}}
689 \newcommand{\cftsubtabdotsep}{\cftdotsep}
690 \newcommand{\cftsubtabpagefont}{\normalfont}
691 \newcommand{\cftsubtabafterpnum}{}

```

This is the end of `\@cftsetsubtab`.

```
692 }
693
```

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.

```

694 \DeclareOption{subfigure}{%
695   \@cftsetsubfig\@cftsetsubtab
696   \AtBeginDocument{\@cftl@subfig\@cftl@subtab}
697 }
698 \ProcessOptions\relax
699

```

3.2 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.

```

700 \newcommand{\cftchapterprecis}[1]{%
701   \cftchapterprecishere{#1}
702   \cftchapterprecistoc{#1}}

```

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

```
703 \newcommand{\cftchapterprecishere}[1]{%
704   \vspace*{-2\baselineskip}
   Typeset its argument using italic font in a quote environment.
705   \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 the title text of a `\chapter` heading, using an italic font.

```
706 \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.
707   {\leftskip \cftchapindent\relax
708    \advance\leftskip \cftchapnumwidth\relax
   Set the right hand margin to \@tocrmarg.
709    \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.
710    \textit{#1}\protect\par}}}
```

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

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

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

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

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

```
715 \newcommand{\cftaddnumtitleline}[5]{%
716   \addtocontents{#1}{%
717     {\protect\contentsline{#2}{\protect\numberline{#3}{\protect\ignorespaces #4}}{#5}}}
```

The end of this package.

```
718 </usc>
```

References

- [GMS94] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley Publishing Company, 1994.
- [Wil96] Peter R. Wilson. *LaTeX for standards: The LaTeX package files user manual*. NIST Report NISTIR, June 1996.