The tocloft package*

Author: Peter Wilson, Herries Press
Maintainer: Will Robertson
will dot robertson at latex-project dot org

2013/05/02

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

List of Figures

*This file (tocloft.dtx) has version number v2.3f, last revised 2013/05/02.

List of Tables

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 commands[2] 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 (numberered) 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` `\@tocrmarg` `\@dotsep` The general layout of a typeset entry is illustrated in Figure 1. There are three

[1] Thanks 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

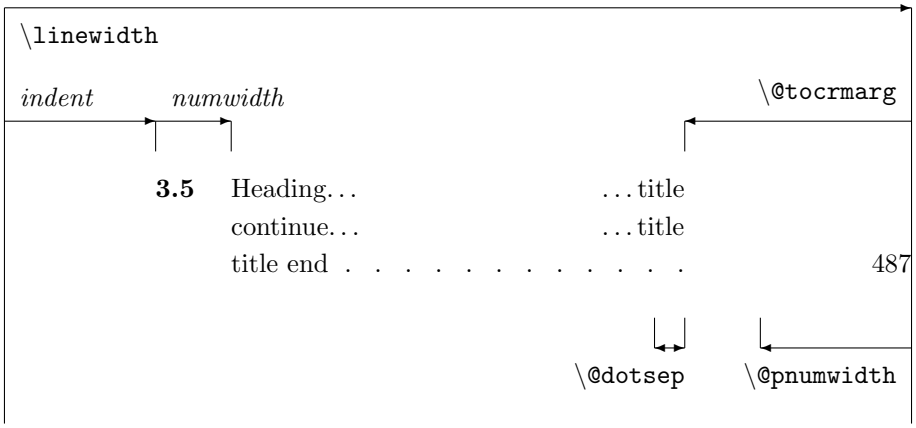

Figure 1: Layout of a ToC (LoF, LoT) entry

internal LaTeX commands that are used in the typesetting. The page number is typeset flushright in a box of width `\@pnumwidth`, and the box is at the righthand margin. If the page number is too long to fit into the box it will stick out into the righthand margin. The title text is indented from the righthand margin by an amount given by `\@tocrmarg`. Note that `\@tocrmarg` should be greater than `\@pnumwidth`. Some entries are typeset with a dotted leader between the end of the title 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.

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

[3]There are 18mu to 1em.

Table 1: Indents and Numwidths (in ems)

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

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

`\@dottedtocline`

Most of the `\l@kind` commands are defined in terms of the `\@dottedtocline` command. This command takes three arguments:
`\@dottedtocline{`⟨*seclevel*⟩`}{`⟨*indent*⟩`}{`⟨*numwidth*⟩`}`.
For example, one definition of the `\l@section` command is:

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

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

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

`\addtocontents`

LaTeX 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 `\protect`ed.

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

- If your page numbers stick out into the righthand margin

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

but using lengths appropriate to your document.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2 The tocloft package

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

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

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

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

`\tocloftpagestyle`

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

2.1 Package options

The package takes the following options:

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

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

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

2.2 Changing the titles

Commands are provided for controlling the appearance of the titles. Following LaTeX custom, the title texts are the values of the `\contentsname`, `\listfigurename` and `\listtablename` commands.

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

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

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

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

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

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

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

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

will put the word 'Page' flushright on the line following the title. (If you do this, then you may need to decrease `\cftafterZtitleskip`).

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

2.3 Typesetting the entries

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

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

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

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

`\cftdotsep` `\cftnodots` 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` `\cftsetrmarg` The page numbers are typeset in a fixed width box. The command `\cftsetpnumwidth{`⟨*length*⟩`}` can be used to change the width of the box (LaTeX's internal `\@pnumwidth`). The title texts will end before reaching the righthand margin. `\cftsetrmarg{`⟨*length*⟩`}` can be used to set this distance (LaTeX'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\cftaftersnum\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` `\cftXaftersnum` `\cftXaftersnumb` Normally 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.

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 LaTeX ideosyncracies, \cftpartpresnum may become doubled in the output if a third-party package behaves differently to that of the default internal LaTeX 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
\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.

\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}
```

[4] Lyndon Dudding (lyndon.dudding@totalise.co.uk) discovered this.
[5] With thanks to David Holz (lbda@earthlink.net) for requesting this.

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

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

```
\renewcommand{\cftXfont}{\hfill\bfseries}
\renewcommand{\cftXleader}{}
```

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

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

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

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

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

`\cftpagenumbersoff`
`\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}%                                              level
   {\parindent}%                     indent from left margin
   {3.25ex \@plus1ex \@minus .2ex}%       skip above heading
   {-1em}%       runin heading with 1em between title & text
   {\normalfont\normalsize\itshape}% italic number and title
```

```
}
\newlistentry[subparagraph]{subsubpara}{toc}{5}
\cftsetindents{subsubpara}{14.0em}{7.0em}
\newcommand*{\subsubparamark}[1]{}     % gobble heading mark
```

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

```
No room for a new write
```

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

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

2.5 Experimental utilities

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

`\cftchapterprecis`

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

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

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

`\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

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

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 LaTeX values for `\@pnumwidth` and `\@tocrmarg` are 1.55em and 2.55em respectively, one might do the following for a figure on the frontispiece page.

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

2.6 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`, `\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
\@cftasnum
\@cftasnumb

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

The tocloft and minitoc packages have an unfortunate interaction,[7] which fortunately can be fixed. In the normal course of events, when minitoc is used in a chaptered document it will typeset section entries in the 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 LaTeX 2$_\varepsilon$ but no extra packages.

```
1 ⟨*usc⟩
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{tocloft}[2013/05/02 v2.3f parameterised ToC, etc., typesetting]
```

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

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

`\@cftquit`
`\if@cfthaschapter`

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

```
\newcommand{\@cftquit}{}
\newif\if@cfthaschapter
```

`\if@cftkoma`

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

```
\newif\if@cftkoma
\@cftkomafalse
\@ifclassloaded{scrartcl}{\@cftkomatrue}{}
\@ifclassloaded{scrreprt}{\@cftkomatrue}{}
\@ifclassloaded{scrbook}{\@cftkomatrue}{}
```

`\if@cfttitlesec`

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

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

```
\@ifundefined{chapter}{%
  \@cfthaschapterfalse
  \@ifundefined{section}{%
    \PackageWarning{tocloft}%
      {I don't recognize any sectional divisions so I'll do nothing}
    \renewcommand{\@cftquit}{\endinput}
    }{\PackageInfo{tocloft}{The document has section divisions}}
  }{\@cfthaschaptertrue
    \PackageInfo{tocloft}{The document has chapter divisions}}
```

Perhaps quit now.

```
\@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.

```
\newif\if@cfttocbibind
\AtBeginDocument{%
  \@ifpackageloaded{tocbibind}{\@cfttocbibindtrue}{\@cfttocbibindfalse}
  \if@cfttocbibind
    \@ifpackagelater{tocbibind}{1998/11/16}{}{%
      \PackageWarning{tocloft}{%
You are using a version of the tocbibind package\MessageBreak
that is not compatible with tocloft.\MessageBreak
The results may be surprising.\MessageBreak
Consider installing the current version of tocbibind.}}
  \fi
}
```

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

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

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

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

Process the options.

```

\ProcessOptions\relax

```

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

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

```

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

```
\newcommand{\cftmarktoc}{%
  \@mkboth{\MakeUppercase\contentsname}{\MakeUppercase\contentsname}}
\newcommand{\cftmarklof}{%
  \@mkboth{\MakeUppercase\listfigurename}{\MakeUppercase\listfigurename}}
\newcommand{\cftmarklot}{%
  \@mkboth{\MakeUppercase\listtablename}{\MakeUppercase\listtablename}}
\if@cftkoma
  \renewcommand{\cftmarktoc}{%
    \@mkboth{\contentsname}{\contentsname}}
  \renewcommand{\cftmarklof}{%
    \@mkboth{\listfigurename}{\listfigurename}}
  \renewcommand{\cftmarklot}{%
    \@mkboth{\listtablename}{\listtablename}}
\fi
```

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

```
\providecommand{\@cfttocstart}{%
  \if@cfthaschapter
    \if@twocolumn
      \@restonecoltrue\onecolumn
    \else
```

```
      \@restonecolfalse
    \fi
  \fi}
```

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

```
\providecommand{\@cfttocfinish}{%
  \if@cfthaschapter
    \if@restonecol\twocolumn\fi
  \fi}
```

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

```
\providecommand{\phantomsection}{}

```

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

```
\newcommand{\@cftdobibtoc}{%
  \if@dotoctoc
    \if@bibchapter
      \phantomsection
      \addcontentsline{toc}{chapter}{\contentsname}
    \else
      \phantomsection
      \addcontentsline{toc}{\@tocextra}{\contentsname}
    \fi
  \fi}

```

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

```
\newlength{\cftparskip}
\setlength{\cftparskip}{0pt}

```

`\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.

```
\AtBeginDocument{%
\if@cftnctoc\else
  \renewcommand{\tableofcontents}{%
    \@cfttocstart
```

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

```
    \par
    \begingroup
      \parindent\z@ \parskip\cftparskip
      \@cftmaketoctitle
```

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

```
      \if@cfttocbibind
        \@cftdobibtoc
      \fi
```

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

```
      \@starttoc{toc}%
    \endgroup
    \@cfttocfinish}
\fi
}
```

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

```
\newcommand{\@cftmaketoctitle}{%
  \addpenalty\@secpenalty
  \if@cfthaschapter
    \vspace*{\cftbeforetoctitleskip}%
  \else
    \vspace{\cftbeforetoctitleskip}%
  \fi
  \@cftpagestyle
  {\interlinepenalty\@M
  {\cfttoctitlefont\contentsname}{\cftaftertoctitle}%
  \cftmarktoc
  \par\nobreak
  \vskip \cftaftertoctitleskip
  \@afterheading}}
```

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

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

```
\if@cfthaschapter
  \setlength{\cftbeforetoctitleskip}{50pt}
  \setlength{\cftaftertoctitleskip}{40pt}
\else
  \setlength{\cftbeforetoctitleskip}{3.5ex \@plus 1ex \@minus .2ex}
  \setlength{\cftaftertoctitleskip}{2.3ex \@plus.2ex}
\fi
```

`\cfttoctitlefont` `\cftaftertoctitle` The ToC title is typeset in the style given by `\cfttoctitlefont`. The macro `\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).

```
\if@cfthaschapter
  \newcommand{\cfttoctitlefont}{\normalfont\Huge\bfseries}
```

```
  \if@cftkoma\renewcommand{\cfttoctitlefont}{\size@chapter\sectfont}\fi
\else
  \newcommand{\cfttoctitlefont}{\normalfont\Large\bfseries}
  \if@cftkoma\renewcommand{\cfttoctitlefont}{\size@section\sectfont}\fi
\fi
\newcommand{\cftaftertoctitle}{}
```

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

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

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

```
\newcommand{\cftpnumalign}{r}
```

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

```
\providecommand{\cftdot}{.}
\providecommand{\cftdotfill}[1]{%
  \def\@tempa{#1}%
  \def\@tempb{\cftnodots}%
  \ifx\@tempa\@tempb
    \hfill
  \else
    \leaders\hbox{$\m@th\mkern #1 mu\hbox{\cftdot}\mkern #1 mu$}\hfill
  \fi
}
```

`\cftdotsep` `\cftnodots` `\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.)

```
\providecommand{\cftdotsep}{4.5}
\newcommand{\cftnodots}{5000}
```

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

A `.toc` (also `.lof` and `.lot`) file consists of a list of `\contentsline{`⟨*kind*⟩`}{`⟨*title*⟩`}{`⟨*page*⟩`}` commands, where ⟨*kind*⟩ is the kind of heading (e.g., `part` or `section` or `figure`), ⟨*title*⟩ is the title text (including the number), and ⟨*page*⟩ is the page number. The entries are inserted into the file by calling the `\addcontentsline{`⟨*file*⟩`}{`⟨*kind*⟩`}{`⟨*title*⟩`}` command, where ⟨*file*⟩ is the file extension (e.g., `toc`, `lot`) and the other arguments are the same as for the `\contentsline` command. (Arbitrary stuff may also be put into the file via the `\addtocontents{`⟨*file*⟩`}{`⟨*text*⟩`}` command). The typesetting of the `\contentsline` entries is performed by commands of the form

\l@kind. The sectioning and captioning commands call \addcontentsline to insert their titles into the .toc etc., files.

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

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

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

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

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

\numberline The purpose of the \numberline{⟨*secnum*⟩} command is to typeset ⟨*secnum*⟩ left justified in a box of width \@tempdima. I redefine it to add three additional parameters, namely \@cftbsnum, \@cftasnum and \@cftasnumb (see ltsect.dtx for the original definition).

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

\@cftbsnum \@cftasnum \@cftasnumb 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]

```
\newcommand{\@cftbsnum}{}
\newcommand{\@cftasnum}{}
\newcommand{\@cftasnumb}{}
```

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

```
\newif\if@cftdopart
\newif\if@cfthaspart
\@ifundefined{part}{\@cfthaspartfalse}{\@cfthasparttrue}
\if@cfthaspart
```

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

```
\renewcommand*{\l@part}[2]{%
  \@cftdopartfalse
  \ifnum \c@tocdepth >-2\relax
    \if@cfthaschapter
      \@cftdoparttrue
    \fi
    \ifnum \c@tocdepth >\m@ne
      \if@cfthaschapter\else
        \@cftdoparttrue
      \fi
    \fi
  \fi
  \if@cftdopart
    \if@cfthaschapter
      \addpenalty{-\@highpenalty}%
    \else
      \addpenalty\@secpenalty
    \fi
    \addvspace{\cftbeforepartskip}%
    \begingroup
      {\leftskip \cftpartindent\relax
       \rightskip \@tocrmarg
       \parfillskip -\rightskip
       \parindent \cftpartindent\relax\@afterindenttrue
       \interlinepenalty\@M
       \leavevmode
       \@tempdima \cftpartnumwidth\relax
       \let\@cftbsnum \cftpartpresnum
       \let\@cftasnum \cftpartaftersnum
       \let\@cftasnumb \cftpartaftersnumb
       \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
```

In default LaTeX, 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.

```
       {\cftpartfont \if@cftkoma\else\if@cfttitlesec\else\cftpartpresnum\fi\fi #1}%
       \cftpartfillnum{#2}}
      \nobreak
      \if@cfthaschapter
        \global\@nobreaktrue
        \everypar{\global\@nobreakfalse\everypar{}}%
      \else
        \if@compatibility
          \global\@nobreaktrue
          \everypar{\global\@nobreakfalse\everypar{}}%
        \fi
      \fi
    \endgroup
  \fi}
\fi
```

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

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

```
\if@cfthaspart
  \newlength{\cftbeforepartskip}
    \setlength{\cftbeforepartskip}{2.25em \@plus\p@}
  \newlength{\cftpartnumwidth}
    \setlength{\cftpartnumwidth}{0em}
  \newcommand{\cftpartfont}{\large\bfseries}
  \newcommand{\cftpartpresnum}{}
  \newcommand{\cftpartaftersnum}{}
  \newcommand{\cftpartaftersnumb}{}
  \newcommand{\cftpartleader}{\large\bfseries\cftdotfill{\cftpartdotsep}}
  \newcommand{\cftpartdotsep}{\cftnodots}
  \newcommand{\cftpartpagefont}{\large\bfseries}
  \newcommand{\cftpartafterpnum}{}
  \newlength{\cftpartindent}
    \setlength{\cftpartindent}{0em}
  \newcommand{\cftpartfillnum}[1]{%
    {\cftpartleader}%
    {\makebox[\@pnumwidth][\cftpnumalign]{\cftpartpagefont #1}\cftpartafterpnum\par}%
  }
```

koma classes use some different settings.

```
  \if@cftkoma
    \setlength{\cftpartnumwidth}{2em}
    \renewcommand{\cftpartfont}{\sectfont\large}
    \renewcommand{\cftpartpagefont}{\sectfont\large}
  \fi
\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.

```
\if@cfthaschapter
\renewcommand*{\l@chapter}[2]{%
  \ifnum \c@tocdepth >\m@ne
    \addpenalty{-\@highpenalty}%
    \vskip \cftbeforechapskip
    {\leftskip \cftchapindent\relax
     \rightskip \@tocrmarg
     \parfillskip -\rightskip
     \parindent \cftchapindent\relax\@afterindenttrue
     \interlinepenalty\@M
     \leavevmode
     \@tempdima \cftchapnumwidth\relax
     \let\@cftbsnum \cftchappresnum
     \let\@cftasnum \cftchapaftersnum
     \let\@cftasnumb \cftchapaftersnumb
     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
     {\cftchapfont #1}\nobreak
```

```
    \cftchapfillnum{#2}}%
  \fi}%
\fi
```

\cftbeforechapskip
\cftchapindent
\cftchapnumwidth
\cftchapfont
\cftchappresnum
\cftchapaftersnum
\cftchapaftersnumb
\cftchapleader
\cftchapdotsep
\cftchappagefont
\cftchapafterpnum
\cftchapfillnum

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

```
\if@cfthaschapter
  \newlength{\cftbeforechapskip}
    \setlength{\cftbeforechapskip}{1.0em \@plus\p@}
  \newlength{\cftchapindent}
    \setlength{\cftchapindent}{0em}
  \newlength{\cftchapnumwidth}
    \setlength{\cftchapnumwidth}{1.5em}
  \newcommand{\cftchapfont}{\bfseries}
  \newcommand{\cftchappresnum}{}
  \newcommand{\cftchapaftersnum}{}
  \newcommand{\cftchapaftersnumb}{}
  \newcommand{\cftchapleader}{\bfseries\cftdotfill{\cftchapdotsep}}
  \newcommand{\cftchapdotsep}{\cftnodots}
  \newcommand{\cftchappagefont}{\bfseries}
  \newcommand{\cftchapafterpnum}{}
  \newcommand{\cftchapfillnum}[1]{%
    {\cftchapleader}\nobreak
    \makebox[\@pnumwidth][\cftpnumalign]{\cftchappagefont #1}\cftchapafterpnum\par
  }
```

koma classes have different chapter settings.

```
  \if@cftkoma
    \renewcommand{\cftchapfont}{\sectfont}
  \fi
\fi
```

\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).

```
\renewcommand*{\l@section}[2]{%
  \ifnum \c@tocdepth >\z@
    \if@cfthaschapter
      \vskip \cftbeforesecskip
    \else
      \addpenalty\@secpenalty
      \addvspace{\cftbeforesecskip}
    \fi
    {\leftskip \cftsecindent\relax
     \rightskip \@tocrmarg
     \parfillskip -\rightskip
     \parindent \cftsecindent\relax\@afterindenttrue
     \interlinepenalty\@M
     \leavevmode
```

```
    \@tempdima \cftsecnumwidth\relax
    \let\@cftbsnum \cftsecpresnum
    \let\@cftasnum \cftsecaftersnum
    \let\@cftasnumb \cftsecaftersnumb
    \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
    {\cftsecfont #1}\nobreak
    \cftsecfillnum{#2}}%
  \fi}
```

\cftbeforesecskip
\cftsecindent
\cftsecnumwidth
\cftsecfont
\cftsecpresnum
\cftsecaftersnum
\cftsecaftersnumb
\cftsecleader
\cftsecdotsep
\cftsecpagefont
\cftsecafterpnum
\cftsecfillnum

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

```
\newlength{\cftbeforesecskip}
\newlength{\cftsecindent}
\newlength{\cftsecnumwidth}
\newcommand{\cftsecpresnum}{}
\newcommand{\cftsecaftersnum}{}
\newcommand{\cftsecaftersnumb}{}
\if@cfthaschapter
  \setlength{\cftbeforesecskip}{\z@ \@plus.2\p@}
  \setlength{\cftsecindent}{1.5em}
  \setlength{\cftsecnumwidth}{2.3em}
  \newcommand{\cftsecfont}{\normalfont}
  \newcommand{\cftsecleader}{\normalfont\cftdotfill{\cftsecdotsep}}
  \newcommand{\cftsecdotsep}{\cftdotsep}
  \newcommand{\cftsecpagefont}{\normalfont}
\else
  \setlength{\cftbeforesecskip}{1.0em \@plus\p@}
  \setlength{\cftsecindent}{0em}
  \setlength{\cftsecnumwidth}{1.5em}
  \newcommand{\cftsecfont}{\bfseries}
  \newcommand{\cftsecleader}{\bfseries\cftdotfill{\cftsecdotsep}}
  \newcommand{\cftsecdotsep}{\cftnodots}
  \newcommand{\cftsecpagefont}{\bfseries}
\fi
\newcommand{\cftsecafterpnum}{}
\newcommand{\cftsecfillnum}[1]{%
  {\cftsecleader}\nobreak
  \makebox[\@pnumwidth][\cftpnumalign]{\cftsecpagefont #1}\cftsecafterpnum\par
}
```

\l@subsection

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

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

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

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

Add some vertical space.

```
    \vskip \cftbeforesubsecskip
```

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

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

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

```
     \rightskip \@tocrmarg
```

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

```
     \parfillskip -\rightskip
```

Set the paragraph indent to the entry's indentation.

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

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

```
     \interlinepenalty\@M
```

Make sure that we have left vertical mode.

```
     \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.

```
     \@tempdima \cftsubsecnumwidth\relax
     \let\@cftbsnum \cftsubsecpresnum
     \let\@cftasnum \cftsubsecaftersnum
     \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.

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

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

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

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

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

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

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

```
\newlength{\cftbeforesubsecskip}
  \setlength{\cftbeforesubsecskip}{\z@ \@plus.2\p@}
\newlength{\cftsubsecindent}
\newlength{\cftsubsecnumwidth}
\if@cfthaschapter
  \setlength{\cftsubsecindent}{3.8em}
  \setlength{\cftsubsecnumwidth}{3.2em}
\else
  \setlength{\cftsubsecindent}{1.5em}
  \setlength{\cftsubsecnumwidth}{2.3em}
\fi
```

```
\newcommand{\cftsubsecfont}{\normalfont}
\newcommand{\cftsubsecpresnum}{}
\newcommand{\cftsubsecaftersnum}{}
\newcommand{\cftsubsecaftersnumb}{}
\newcommand{\cftsubsecleader}{\normalfont\cftdotfill{\cftsubsecdotsep}}
\newcommand{\cftsubsecdotsep}{\cftdotsep}
\newcommand{\cftsubsecpagefont}{\normalfont}
\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.

```
\newcommand{\cftsubsecfillnum}[1]{%
  {\cftsubsecleader}\nobreak
  \makebox[\@pnumwidth][\cftpnumalign]{\cftsubsecpagefont #1}\cftsubsecafterpnum\par
}
```

`\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).

```
\renewcommand*{\l@subsubsection}[2]{%
  \ifnum \c@tocdepth >\tw@
    \vskip \cftbeforesubsubsecskip
    {\leftskip \cftsubsubsecindent\relax
     \rightskip \@tocrmarg
     \parfillskip -\rightskip
     \parindent \cftsubsubsecindent\relax\@afterindenttrue
     \interlinepenalty\@M
     \leavevmode
     \@tempdima \cftsubsubsecnumwidth\relax
     \let\@cftbsnum \cftsubsubsecpresnum
     \let\@cftasnum \cftsubsubsecaftersnum
     \let\@cftasnumb \cftsubsubsecaftersnumb
     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
     {\cftsubsubsecfont #1}\nobreak
     \cftsubsubsecfillnum{#2}}%
  \fi}
```

`\cftbeforesubsubsecskip`
`\cftsubsubsecindent`
`\cftsubsubsecnumwidth`
`\cftsubsubsecfont`
`\cftsubsubsecpresnum`
`\cftsubsubsecaftersnum`
`\cftsubsubsecaftersnumb`
`\cftsubsubsecleader`
`\cftsubsubsecdotsep`
`\cftsubsubsecpagefont`
`\cftsubsubsecafterpnum`
`\cftsubsubsecfillnum`

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

```
\newlength{\cftbeforesubsubsecskip}
  \setlength{\cftbeforesubsubsecskip}{\z@ \@plus.2\p@}
\newlength{\cftsubsubsecindent}
\newlength{\cftsubsubsecnumwidth}
\if@cfthaschapter
  \setlength{\cftsubsubsecindent}{7.0em}
  \setlength{\cftsubsubsecnumwidth}{4.1em}
\else
```

```
  \setlength{\cftsubsubsecindent}{3.8em}
  \setlength{\cftsubsubsecnumwidth}{3.2em}
\fi
\newcommand{\cftsubsubsecfont}{\normalfont}
\newcommand{\cftsubsubsecpresnum}{}
\newcommand{\cftsubsubsecaftersnum}{}
\newcommand{\cftsubsubsecaftersnumb}{}
\newcommand{\cftsubsubsecleader}{\normalfont\cftdotfill{\cftsubsubsecdotsep}}
\newcommand{\cftsubsubsecdotsep}{\cftdotsep}
\newcommand{\cftsubsubsecpagefont}{\normalfont}
\newcommand{\cftsubsubsecafterpnum}{}
\newcommand{\cftsubsubsecfillnum}[1]{%
  {\cftsubsubsecleader}\nobreak
  \makebox[\@pnumwidth][\cftpnumalign]{\cftsubsubsecpagefont #1}\cftsubsubsecafterpnum\par
}
```

`\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).

```
\renewcommand*{\l@paragraph}[2]{%
  \ifnum \c@tocdepth >3\relax
    \vskip \cftbeforeparaskip
    {\leftskip \cftparaindent\relax
     \rightskip \@tocrmarg
     \parfillskip -\rightskip
     \parindent \cftparaindent\relax\@afterindenttrue
     \interlinepenalty\@M
     \leavevmode
     \@tempdima \cftparanumwidth\relax
     \let\@cftbsnum \cftparapresnum
     \let\@cftasnum \cftparaaftersnum
     \let\@cftasnumb \cftparaaftersnumb
     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
     {\cftparafont #1}\nobreak
     \cftparafillnum{#2}}%
  \fi}
```

`\cftbeforeparaskip` `\cftparaindent` `\cftparanumwidth` `\cftparafont` `\cftparapresnum` `\cftparaaftersnum` `\cftparaaftersnumb` `\cftparaleader` `\cftparadotsep` `\cftparapagefont` `\cftparaafterpnum` `\cftparafillnum`

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

```
\newlength{\cftbeforeparaskip}
  \setlength{\cftbeforeparaskip}{\z@ \@plus.2\p@}
\newlength{\cftparaindent}
\newlength{\cftparanumwidth}
\if@cfthaschapter
  \setlength{\cftparaindent}{10em}
  \setlength{\cftparanumwidth}{5em}
\else
  \setlength{\cftparaindent}{7.0em}
  \setlength{\cftparanumwidth}{4.1em}
```

```
\fi
\newcommand{\cftparafont}{\normalfont}
\newcommand{\cftparapresnum}{}
\newcommand{\cftparaaftersnum}{}
\newcommand{\cftparaaftersnumb}{}
\newcommand{\cftparaleader}{\normalfont\cftdotfill{\cftparadotsep}}
\newcommand{\cftparadotsep}{\cftdotsep}
\newcommand{\cftparapagefont}{\normalfont}
\newcommand{\cftparaafterpnum}{}
\newcommand{\cftparafillnum}[1]{%
  {\cftparaleader}\nobreak
  \makebox[\@pnumwidth][\cftpnumalign]{\cftparapagefont #1}\cftparaafterpnum\par
}
```

`\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).

```
\renewcommand*{\l@subparagraph}[2]{%
  \ifnum \c@tocdepth >4\relax
    \vskip \cftbeforesubparaskip
    {\leftskip \cftsubparaindent\relax
     \rightskip \@tocrmarg
     \parfillskip -\rightskip
     \parindent \cftsubparaindent\relax\@afterindenttrue
     \interlinepenalty\@M
     \leavevmode
     \@tempdima \cftsubparanumwidth\relax
     \let\@cftbsnum \cftsubparapresnum
     \let\@cftasnum \cftsubparaaftersnum
     \let\@cftasnumb \cftsubparaaftersnumb
     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
     {\cftsubparafont #1}\nobreak
     \cftsubparafillnum{#2}}%
  \fi}
```

`\cftbeforesubparaskip`
`\cftsubparaindent`
`\cftsubparanumwidth`
`\cftsubparafont`
`\cftsubparapresnum`
`\cftsubparaaftersnum`
`\cftsubparaaftersnumb`
`\cftsubparaleader`
`\cftsubparadotsep`
`\cftsubparapagefont`
`\cftsubparaafterpnum`
`\cftsubparafillnum`

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

```
\newlength{\cftbeforesubparaskip}
  \setlength{\cftbeforesubparaskip}{\z@ \@plus.2\p@}
\newlength{\cftsubparaindent}
\newlength{\cftsubparanumwidth}
\if@cfthaschapter
  \setlength{\cftsubparaindent}{12em}
  \setlength{\cftsubparanumwidth}{6em}
\else
  \setlength{\cftsubparaindent}{10em}
  \setlength{\cftsubparanumwidth}{5em}
\fi
\newcommand{\cftsubparafont}{\normalfont}
```

```
\newcommand{\cftsubparapresnum}{}
\newcommand{\cftsubparaaftersnum}{}
\newcommand{\cftsubparaaftersnumb}{}
\newcommand{\cftsubparaleader}{\normalfont\cftdotfill{\cftsubparadotsep}}
\newcommand{\cftsubparadotsep}{\cftdotsep}
\newcommand{\cftsubparapagefont}{\normalfont}
\newcommand{\cftsubparaafterpnum}{}
\newcommand{\cftsubparafillnum}[1]{%
  {\cftsubparaleader}\nobreak
  \makebox[\@pnumwidth][\cftpnumalign]{\cftsubparapagefont #1}\cftsubparaafterpnum\par
}
```

`\@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.

```
\newcommand{\@cftdobiblof}{%
  \if@dotoclof
    \if@bibchapter
      \phantomsection
      \addcontentsline{toc}{chapter}{\listfigurename}
    \else
      \phantomsection
      \addcontentsline{toc}{\@tocextra}{\listfigurename}
    \fi
  \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).

```
\AtBeginDocument{%
\if@cftnctoc\else
\renewcommand{\listoffigures}{%
  \@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.

```
  \par
  \begingroup
    \parindent\z@ \parskip\cftparskip
    \@cftmakeloftitle
    \if@cfttocbibind
      \@cftdobiblof
    \fi
    \@starttoc{lof}%
  \endgroup
```

Finally, restore any multicolumn typesetting.

```
  \@cfttocfinish}%
\fi
}
```

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

```
\newcommand{\@cftmakeloftitle}{%
  \addpenalty\@secpenalty
  \if@cfthaschapter
    \vspace*{\cftbeforeloftitleskip}
  \else
    \vspace{\cftbeforeloftitleskip}
  \fi
  \@cftpagestyle
  {\interlinepenalty\@M
  {\cftloftitlefont\listfigurename}{\cftafterloftitle}
  \cftmarklof
  \par\nobreak
  \vskip \cftafterloftitleskip
  \@afterheading}}
```

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

```
\newlength{\cftbeforeloftitleskip}
\newlength{\cftafterloftitleskip}
```

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

```
\if@cfthaschapter
  \setlength{\cftbeforeloftitleskip}{50pt}
  \setlength{\cftafterloftitleskip}{40pt}
\else
  \setlength{\cftbeforeloftitleskip}{3.5ex \@plus 1ex \@minus .2ex}
  \setlength{\cftafterloftitleskip}{2.3ex \@plus.2ex}
\fi
```

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

```
\if@cfthaschapter
  \newcommand{\cftloftitlefont}{\normalfont\Huge\bfseries}
  \if@cftkoma\renewcommand{\cftloftitlefont}{\size@chapter\sectfont}\fi
\else
  \newcommand{\cftloftitlefont}{\normalfont\Large\bfseries}
  \if@cftkoma\renewcommand{\cftloftitlefont}{\size@section\sectfont}\fi
\fi
\newcommand{\cftafterloftitle}{}
```

\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).

```
\renewcommand*{\l@figure}[2]{%
  \ifnum \c@lofdepth >\z@
    \vskip \cftbeforefigskip
    {\leftskip \cftfigindent\relax
     \rightskip \@tocrmarg
     \parfillskip -\rightskip
     \parindent \cftfigindent\relax\@afterindenttrue
     \interlinepenalty\@M
     \leavevmode
     \@tempdima \cftfignumwidth\relax
     \let\@cftbsnum \cftfigpresnum
     \let\@cftasnum \cftfigaftersnum
     \let\@cftasnumb \cftfigaftersnumb
     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
     {\cftfigfont #1}\nobreak
     \cftfigfillnum{#2}}%
   \fi
  }
```

\cftbeforefigskip
\cftfigindent
\cftfignumwidth
\cftfigfont
\cftfigpresnum
\cftfigaftersnum
\cftfigaftersnumb
\cftfigleader
\cftfigdotsep
\cftfigpagefont
\cftfigafterpnum
\cftfigfillnum

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

```
\newlength{\cftbeforefigskip}
  \setlength{\cftbeforefigskip}{\z@ \@plus.2\p@}
\newlength{\cftfigindent}
  \setlength{\cftfigindent}{1.5em}
\newlength{\cftfignumwidth}
  \setlength{\cftfignumwidth}{2.3em}
\newcommand{\cftfigfont}{\normalfont}
\newcommand{\cftfigpresnum}{}
\newcommand{\cftfigaftersnum}{}
\newcommand{\cftfigaftersnumb}{}
\newcommand{\cftfigleader}{\normalfont\cftdotfill{\cftfigdotsep}}
\newcommand{\cftfigdotsep}{\cftdotsep}
\newcommand{\cftfigpagefont}{\normalfont}
\newcommand{\cftfigafterpnum}{}
\newcommand{\cftfigfillnum}[1]{%
  {\cftfigleader}\nobreak
  \makebox[\@pnumwidth][\cftpnumalign]{\cftfigpagefont #1}\cftfigafterpnum\par
}
```

lofdepth
lotdepth

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

```
\if@cftsubfigopt\else
  \newcounter{lofdepth}\setcounter{lofdepth}{1}
  \newcounter{lotdepth}\setcounter{lotdepth}{1}
\fi
```

```
587
```

`\@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.

```
588 \newcommand{\@cftdobiblot}{%
589   \if@dotoclot
590     \if@bibchapter
591       \phantomsection
592       \addcontentsline{toc}{chapter}{\listtablename}
593     \else
594       \phantomsection
595       \addcontentsline{toc}{\@tocextra}{\listtablename}
596     \fi
597   \fi}
598
```

`\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).

```
599 \AtBeginDocument{%
600 \if@cftnctoc\else
601 \renewcommand{\listoftables}{%
602   \@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.

```
603   \par
604   \begingroup
605     \parindent\z@ \parskip\cftparskip
606     \@cftmakelottitle
607     \if@cfttocbibind
608       \@cftdobiblot
609     \fi
610     \@starttoc{lot}%
611   \endgroup
```

Finally, restore any multicolumn typesetting.

```
612   \@cfttocfinish}%
613 \fi
614 }
615
```

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

```
616 \newcommand{\@cftmakelottitle}{%
617   \addpenalty\@secpenalty
618   \if@cfthaschapter
619     \vspace*{\cftbeforelottitleskip}
```

```
  \else
    \vspace{\cftbeforelottitleskip}
  \fi
  \@cftpagestyle
  {\interlinepenalty\@M
  {\cftlottitlefont\listtablename}{\cftafterlottitle}
  \cftmarklot
  \par\nobreak
  \vskip \cftafterlottitleskip
  \@afterheading}}
```

`\cftbeforelottitleskip` `\cftafterlottitleskip`

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

```
\newlength{\cftbeforelottitleskip}
\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*`.

```
\if@cfthaschapter
  \setlength{\cftbeforelottitleskip}{50pt}
  \setlength{\cftafterlottitleskip}{40pt}
\else
  \setlength{\cftbeforelottitleskip}{3.5ex \@plus 1ex \@minus .2ex}
  \setlength{\cftafterlottitleskip}{2.3ex \@plus.2ex}
\fi
```

`\cftlottitlefont` `\cftafterlottitle`

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).

```
\if@cfthaschapter
  \newcommand{\cftlottitlefont}{\normalfont\Huge\bfseries}
  \if@cftkoma\renewcommand{\cftlottitlefont}{\size@chapter\sectfont}\fi
\else
  \newcommand{\cftlottitlefont}{\normalfont\Large\bfseries}
  \if@cftkoma\renewcommand{\cftlottitlefont}{\size@section\sectfont}\fi
\fi
\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).

```
\renewcommand*{\l@table}[2]{%
  \ifnum\c@lotdepth >\z@
    \vskip \cftbeforetabskip
    {\leftskip \cfttabindent\relax
     \rightskip \@tocrmarg
```

```
    \parfillskip -\rightskip
    \parindent \cfttabindent\relax\@afterindenttrue
    \interlinepenalty\@M
    \leavevmode
    \@tempdima \cfttabnumwidth\relax
    \let\@cftbsnum \cfttabpresnum
    \let\@cftasnum \cfttabaftersnum
    \let\@cftasnumb \cfttabaftersnumb
    \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
    {\cfttabfont #1}\nobreak
    \cfttabfillnum{#2}}%
   \fi
  }
```

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

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

```
\newlength{\cftbeforetabskip}
  \setlength{\cftbeforetabskip}{\z@ \@plus.2\p@}
\newlength{\cfttabindent}
  \setlength{\cfttabindent}{1.5em}
\newlength{\cfttabnumwidth}
  \setlength{\cfttabnumwidth}{2.3em}
\newcommand{\cfttabfont}{\normalfont}
\newcommand{\cfttabpresnum}{}
\newcommand{\cfttabaftersnum}{}
\newcommand{\cfttabaftersnumb}{}
\newcommand{\cfttableader}{\normalfont\cftdotfill{\cfttabdotsep}}
\newcommand{\cfttabdotsep}{\cftdotsep}
\newcommand{\cfttabpagefont}{\normalfont}
\newcommand{\cfttabafterpnum}{}
\newcommand{\cfttabfillnum}[1]{%
  {\cfttableader}\nobreak
  \makebox[\@pnumwidth][\cftpnumalign]{\cfttabpagefont #1}\cfttabafterpnum\par
}
```

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.

```
\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`.

```
\renewcommand*{\l@subfigure}[2]{%
  \ifnum \c@lofdepth > \toclevel@subfigure
```

```
    \vskip \cftbeforesubfigskip
    {\leftskip \cftsubfigindent\relax
     \rightskip \@tocrmarg
     \parfillskip -\rightskip
     \parindent \cftsubfigindent\relax\@afterindenttrue
     \interlinepenalty\@M
     \leavevmode
     \@tempdima \cftsubfignumwidth\relax
     \let\@cftbsnum \cftsubfigpresnum
     \let\@cftasnum \cftsubfigaftersnum
     \let\@cftasnumb \cftsubfigaftersnumb
     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
     {\cftsubfigfont ##1}\nobreak
     \cftsubfigfillnum{##2}}%
  \fi
  }%
}

```

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

```
\newcommand{\@cftsetsubfig}{%

```

`\cftbeforesubfigskip` `\cftsubfigindent` `\cftsubfignumwidth` `\cftsubfigfont` `\cftsubfigpresnum` `\cftsubfigaftersnum` `\cftsubfigaftersnumb` `\cftsubfigleader` `\cftsubfigdotsep` `\cftsubfigpagefont` `\cftsubfigafterpnum` `\toclevel@subfig` `\cftsubfigfillnum`

```
\newlength{\cftbeforesubfigskip}
  \setlength{\cftbeforesubfigskip}{\z@ \@plus.2\p@}
\newlength{\cftsubfigindent}
  \setlength{\cftsubfigindent}{3.8em}
\newlength{\cftsubfignumwidth}
  \setlength{\cftsubfignumwidth}{2.5em}
\newcommand{\cftsubfigfont}{\normalfont}
\newcommand{\cftsubfigpresnum}{}
\newcommand{\cftsubfigaftersnum}{}
\newcommand{\cftsubfigaftersnumb}{}
\newcommand{\cftsubfigleader}{\normalfont\cftdotfill{\cftsubtabdotsep}}
\newcommand{\cftsubfigdotsep}{\cftdotsep}
\newcommand{\cftsubfigpagefont}{\normalfont}
\newcommand{\cftsubfigafterpnum}{}
\providecommand{\toclevel@subfigure}{1}
\newcommand{\cftsubfigfillnum}[1]{%
  {\cftsubfigleader}\nobreak
  \makebox[\@pnumwidth][\cftpnumalign]{\cftsubfigpagefont ##1}\cftsubfigafterpnum\par
}
```

This is the end of `\@cftsetsubfig`.

```
}

```

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

```
\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`.

```
\renewcommand*{\l@subtable}[2]{%
  \ifnum \c@lotdepth > \toclevel@subtable
    \vskip \cftbeforesubtabskip
    {\leftskip \cftsubtabindent\relax
     \rightskip \@tocrmarg
     \parfillskip -\rightskip
     \parindent \cftsubtabindent\relax\@afterindenttrue
     \interlinepenalty\@M
     \leavevmode
     \@tempdima \cftsubtabnumwidth\relax
     \let\@cftbsnum \cftsubtabpresnum
     \let\@cftasnum \cftsubtabaftersnum
     \let\@cftasnumb \cftsubtabaftersnumb
     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
     {\cftsubtabfont ##1}\nobreak
     \cftsubtabfillnum{##2}}%
  \fi
  }%
}
```

\@cftsetsubtab

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

```
\newcommand{\@cftsetsubtab}{%
```

\cftbeforesubtabskip
\cftsubtabindent
\cftsubtabnumwidth
\cftsubtabfont
\cftsubtabpresnum
\cftsubtabaftersnum
\cftsubtabaftersnumb
\cftsubtableader
\cftsubtabdotsep
\cftsubtabpagefont
\cftsubtabafterpnum
\toclevel@subtable
\cftsubtabfillnum

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

```
\newlength{\cftbeforesubtabskip}
  \setlength{\cftbeforesubtabskip}{\z@ \@plus.2\p@}
\newlength{\cftsubtabindent}
  \setlength{\cftsubtabindent}{3.8em}
\newlength{\cftsubtabnumwidth}
  \setlength{\cftsubtabnumwidth}{2.5em}
\newcommand{\cftsubtabfont}{\normalfont}
\newcommand{\cftsubtabpresnum}{}
\newcommand{\cftsubtabaftersnum}{}
\newcommand{\cftsubtabaftersnumb}{}
\newcommand{\cftsubtableader}{\normalfont\cftdotfill{\cftsubtabdotsep}}
\newcommand{\cftsubtabdotsep}{\cftdotsep}
\newcommand{\cftsubtabpagefont}{\normalfont}
\newcommand{\cftsubtabafterpnum}{}
\providecommand{\toclevel@subtable}{1}
\newcommand{\cftsubtabfillnum}[1]{%
  {\cftsubtableader}\nobreak
  \makebox[\@pnumwidth][\cftpnumalign]{\cftsubtabpagefont ##1}\cftsubtabafterpnum\par
}
```

This is the end of `\@cftsetsubtab`.

```
}
```

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.

```

\if@cftsubfigopt
  \@cftsetsubfig\@cftsetsubtab
  \AtBeginDocument{\@cftl@subfig\@cftl@subtab}
\fi
%%  \AtBeginDocument{\if@cftsubfigopt
%%    \@cftsetsubfig\@cftsetsubtab
%%    \@cftl@subfig\@cftl@subtab
%%  \fi}

```

3.2 New list of...

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

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

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

```
  \@ifundefined{c@#2}{%    check & set the counter
    \ifx \@empty#1\relax
      \newcounter{#2}
    \else
      \@ifundefined{c@#1}{\PackageWarning{tocloft}%
                          {#1 has no counter for use as a `within'}
        \newcounter{#2}}%
      {\newcounter{#2}[#1]%
       \expandafter\edef\csname the#2\endcsname{%
         \expandafter\noexpand\csname the#1\endcsname.\noexpand\arabic{#2}}}
    \fi
    \setcounter{#2}{0}
  }
  {\PackageError{tocloft}{#2 has been previously defined}{\@eha}}

```

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

`\l@X` `\l@X{⟨title⟩}{⟨page⟩}` typesets the entry.

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

Only typeset if the `\Zdepth` is greater than ⟨*level-1*⟩.

```
    \ifnum \@nameuse{c@#3depth} > #4\relax
```

```
      \vskip \@nameuse{cftbefore#2skip}
      {\leftskip \@nameuse{cft#2indent}\relax
       \rightskip \@tocrmarg
       \parfillskip -\rightskip
       \parindent \@nameuse{cft#2indent}\relax\@afterindenttrue
       \interlinepenalty\@M
       \leavevmode
       \@tempdima \@nameuse{cft#2numwidth}\relax
       \expandafter\let\expandafter\@cftbsnum\csname cft#2presnum\endcsname
       \expandafter\let\expandafter\@cftasnum\csname cft#2aftersnum\endcsname
       \expandafter\let\expandafter\@cftasnumb\csname cft#2aftersnumb\endcsname
       \advance\leftskip\@tempdima \null\nobreak\hskip -\leftskip
       {\@nameuse{cft#2font}##1}\nobreak
       \@nameuse{cft#2fillnum}{##2}}%
    \fi
  }  % end of \l@#2

```

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`

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

`\cftXindent`
`\cftXnumwidth`

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

```
  \ifcase #4\relax  % 0
    \setlength{\@nameuse{cft#2indent}}{0em}
    \setlength{\@nameuse{cft#2numwidth}}{1.5em}
  \or               % 1
    \setlength{\@nameuse{cft#2indent}}{1.5em}
    \setlength{\@nameuse{cft#2numwidth}}{2.3em}
  \or               % 2
    \setlength{\@nameuse{cft#2indent}}{3.8em}
    \setlength{\@nameuse{cft#2numwidth}}{3.2em}
  \or               % 3
    \setlength{\@nameuse{cft#2indent}}{7.0em}
    \setlength{\@nameuse{cft#2numwidth}}{4.1em}
  \else             % anything else
    \setlength{\@nameuse{cft#2indent}}{10.0em}
    \setlength{\@nameuse{cft#2numwidth}}{5.0em}
  \fi
```

`\cftXfont`
`\cftXpresnum`
`\cftXaftersnum`
`\cftXaftersnumb`
`\cftXdotsep`
`\cftXleader`
`\cftXpagefont`
`\cftXafterpnum`

And the remaining commands.

```
  \@namedef{cft#2font}{\normalfont}
```

```
  \@namedef{cft#2presnum}{}
  \@namedef{cft#2aftersnum}{}
  \@namedef{cft#2aftersnumb}{}
  \@namedef{cft#2dotsep}{\cftdotsep}
  \@namedef{cft#2leader}{\normalfont\cftdotfill{\@nameuse{cft#2dotsep}}}
  \@namedef{cft#2pagefont}{\normalfont}
  \@namedef{cft#2afterpnum}{}
```

`\toclevel@X` The hyperref package needs a command `\toclevel@X`, holding the ⟨*level-1*⟩ value.

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

`\cftXfillnum` Typeset the leader and page number.

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

This ends the definition of `\newlistentry`.

```
}
```

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

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

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

```
  \ifx \@empty#1\relax
    \newlistentry{#2}{#3}{0}
  \else
    \newlistentry[#1]{#2}{#3}{0}
  \fi
```

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

`\Zdepth`

```
  \@namedef{ext@#2}{#3}
  \newcounter{#3depth}
  \setcounter{#3depth}{1}
```

`\cftmarkZ` The heading marks for the listing.

```
  \if@cftkoma
    \@namedef{cftmark#3}{%
      \@mkboth{#4}{#4}}
  \else
    \@namedef{cftmark#3}{%
      \@mkboth{\MakeUppercase{#4}}{\MakeUppercase{#4}}}
  \fi
```

`\listofX` Typeset the listing title and entries.

```
 \if@cftnctoc
```

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

```
\@namedef{listof#2}{%
  \@cfttocstart
  \if@cfthaschapter
    \chapter*{#4}
  \else
    \section*{#4}
  \fi
  \@nameuse{cftmark#3}
  \@starttoc{#3}%
  \@cfttocfinish}
\else
```

Otherwise use the fully parameterised definition.

```
  \@namedef{listof#2}{%
    \@cfttocstart
    \par
    \begingroup
      \parindent\z@ \parskip\cftparskip
      \@nameuse{@cftmake#3title}
      \@starttoc{#3}%
    \endgroup
    \@cfttocfinish}
\fi

```

`\@cftmakeZtitle` Typeset the title.

```
  \@namedef{@cftmake#3title}{%
    \addpenalty\@secpenalty
    \if@cfthaschapter
      \vspace*{\@nameuse{cftbefore#3titleskip}}%
    \else
      \vspace{\@nameuse{cftbefore#3titleskip}}%
    \fi
    \@cftpagestyle
    {\interlinepenalty\@M
    {\@nameuse{cft#3titlefont}#4}{\@nameuse{cftafter#3title}}%
    \@nameuse{cftmark#3}%
    \par\nobreak
    \vskip \@nameuse{cftafter#3titleskip}%
    \@afterheading}}

```

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

```
  \expandafter\newlength\csname cftbefore#3titleskip\endcsname
  \expandafter\newlength\csname cftafter#3titleskip\endcsname
  \if@cfthaschapter
     \setlength{\@nameuse{cftbefore#3titleskip}}{50pt}
```

```
      \setlength{\@nameuse{cftafter#3titleskip}}{40pt}
      \if@cftkoma
        \@namedef{cft#3titlefont}{\size@chapter\sectfont}
      \else
        \@namedef{cft#3titlefont}{\normalfont\Huge\bfseries}
      \fi
    \else
      \setlength{\@nameuse{cftbefore#3titleskip}}{3.5ex \@plus 1ex \@minus .2ex}
      \setlength{\@nameuse{cftafter#3titleskip}}{2.3ex \@plus .2ex}
      \if@cftkoma
        \@namedef{cft#3titlefont}{\size@section\sectfont}
      \else
        \@namedef{cft#3titlefont}{\normalfont\Huge\bfseries}
      \fi
    \fi
```

`\cftafterZtitle` Something to go after the title.

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

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

```
}
```

`\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.

```
\newcommand{\cftsetindents}[3]{%
  \def\@cftemp{#1}
  \ifx\@cftemp\cftchapname
    \@cftsetindents{chap}{#2}{#3}
  \else
    \ifx\@cftemp\cftsecname \@cftsetindents{sec}{#2}{#3}
    \else
      \ifx\@cftemp\cftsubsecname \@cftsetindents{subsec}{#2}{#3}
      \else
        \ifx\@cftemp\cftsubsubsecname \@cftsetindents{subsubsec}{#2}{#3}
        \else
          \ifx\@cftemp\cftparaname \@cftsetindents{para}{#2}{#3}
          \else
            \ifx\@cftemp\cftsubparaname \@cftsetindents{subpara}{#2}{#3}
            \else
              \ifx\@cftemp\cftfigname \@cftsetindents{fig}{#2}{#3}
              \else
                \ifx\@cftemp\cftsubfigname \@cftsetindents{subfig}{#2}{#3}
                \else
                  \ifx\@cftemp\cfttabname \@cftsetindents{tab}{#2}{#3}
                  \else
                    \ifx\@cftemp\cftsubtabname \@cftsetindents{subtab}{#2}{#3}
                    \else
                      \@cftsetindents{#1}{#2}{#3}
```

```
                    \fi
                  \fi
                \fi
              \fi
            \fi
          \fi
        \fi
      \fi
    \fi
  \fi
}
```

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

```
\newcommand{\@cftsetindents}[3]{%
  \setlength{\@nameuse{cft#1indent}}{#2}
  \setlength{\@nameuse{cft#1numwidth}}{#3}
}
```

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.

```
\newcommand{\@cftpnumoff}[1]{%
  \@namedef{cft#1fillnum}##1{%
    \cftparfillskip\@nameuse{cft#1afterpnum}\par}}
```

`\cftchapname` `\cftsecname` `\cftsubsecname` `\cftsubsubsecname` `\cftparaname` `\cftsubparaname` `\cftfigname` `\cftsubfigname` `\cfttabname` `\cftsubtabname`

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

```
\newcommand*{\cftchapname}{chapter}
\newcommand*{\cftsecname}{section}
\newcommand*{\cftsubsecname}{subsection}
\newcommand*{\cftsubsubsecname}{subsubsection}
\newcommand*{\cftparaname}{paragraph}
\newcommand*{\cftsubparaname}{subparagraph}
\newcommand*{\cftfigname}{figure}
\newcommand*{\cftsubfigname}{subfigure}
\newcommand*{\cfttabname}{table}
\newcommand*{\cftsubtabname}{subtable}
```

`\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 opf 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.

```
\DeclareRobustCommand{\cftpagenumbersoff}[1]{%
  \def\@cftemp{#1}
  \ifx\@cftemp\cftchapname
    \@cftpnumoff{chap}
  \else
    \ifx\@cftemp\cftsecname \@cftpnumoff{sec}
    \else
      \ifx\@cftemp\cftsubsecname \@cftpnumoff{subsec}
      \else
        \ifx\@cftemp\cftsubsubsecname \@cftpnumoff{subsubsec}
        \else
          \ifx\@cftemp\cftparaname \@cftpnumoff{para}
          \else
            \ifx\@cftemp\cftsubparaname \@cftpnumoff{subpara}
            \else
              \ifx\@cftemp\cftfigname \@cftpnumoff{fig}
              \else
                \ifx\@cftemp\cftsubfigname \@cftpnumoff{subfig}
                \else
                  \ifx\@cftemp\cfttabname \@cftpnumoff{tab}
                  \else
                    \ifx\@cftemp\cftsubtabname \@cftpnumoff{subtab}
                    \else
                      \@cftpnumoff{#1}
                    \fi
                  \fi
                \fi
              \fi
            \fi
          \fi
        \fi
      \fi
    \fi
  \fi
}

```

`\cftpagenumberson` `\cftpagenumberson{`⟨*entry*⟩`}` is the user level command for reversing the corresponding `\cftpagenumbersoff`.

```
\DeclareRobustCommand{\cftpagenumberson}[1]{%
  \def\@cftemp{#1}
  \ifx\@cftemp\cftchapname
    \@cftpnumon{chap}
  \else
    \ifx\@cftemp\cftsecname \@cftpnumon{sec}
    \else
```

```
      \ifx\@cftemp\cftsubsecname \@cftpnumon{subsec}
      \else
        \ifx\@cftemp\cftsubsubsecname \@cftpnumon{subsubsec}
        \else
          \ifx\@cftemp\cftparaname \@cftpnumon{para}
          \else
            \ifx\@cftemp\cftsubparaname \@cftpnumon{subpara}
            \else
              \ifx\@cftemp\cftfigname \@cftpnumon{fig}
              \else
                \ifx\@cftemp\cftsubfigname \@cftpnumon{subfig}
                \else
                  \ifx\@cftemp\cfttabname \@cftpnumon{tab}
                  \else
                    \ifx\@cftemp\cftsubtabname \@cftpnumon{subtab}
                    \else
                      \@cftpnumon{#1}
                    \fi
                  \fi
                \fi
              \fi
            \fi
          \fi
        \fi
      \fi
    \fi
  \fi
}
```

`\@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.

```
\newcommand{\@cftpnumon}[1]{%
  \@namedef{cft#1fillnum}##1{%
    {\@nameuse{cft#1leader}}\nobreak
    \makebox[\@pnumwidth][\cftpnumalign]{\@nameuse{cft#1pagefont}##1}\@nameuse{cft#1afterpnum}
  }%
}
```

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 expects to be called immediately after a `\chapter` command.

```
\newcommand{\cftchapterprecis}[1]{%
  \cftchapterprecishere{#1}
  \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.

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

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

```
  \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.

```
\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.

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

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

```
   \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.

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

```

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

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

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

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

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

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

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

```
\AtBeginDocument{%
  \@ifpackageloaded{hyperref}{%
    \renewcommand{\cftaddtitleline}[4]{\addtocontents{#1}{%
```

```
      \protect\contentsline{#2}{#3}{#4}{\@currentHref}}}
    \renewcommand{\cftaddnumtitleline}[5]{\addtocontents{#1}{%
      \protect\contentsline{#2}{\protect\numberline{#3}#4}{#5}{\@currentHref}}}
  }{}
}
```

The end of this package.

⟨/usc⟩

References

[Coc95] Steven Douglas Cochran. *The subfigure package*. March 1995. (Available from CTAN as file `subfigure.dtx`)

[Dru99] Jean-Pierre Drucbert. *The minitoc package*. August 1999. (Available from CTAN in subdirectory `/minitoc`)

[GMS94] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley Publishing Company, 1994.

[Lin97] Ulf A. Lindgren. *FncyChap V1.11*. April 1997. (Available from CTAN in subdirectory `/fncychap`)

[Lin95] Anselm Lingnau. *An Improved Environment for Floats*. March 1995. (Available from CTAN in subdirectory `/float`)

[Wil96a] Peter Williams. *Algorithms*. April 1996. (Available from CTAN in subdirectory `/algorithm`)

[Wil96b] Peter R. Wilson. *LaTeX for standards: The LaTeX package files user manual*. NIST Report NISTIR, June 1996.

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

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

Index

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

A

B

C

S

T

Z