The KOMA‑Script package
`tocstyle`[*]

Markus Kohm

2013/08/11

with = will be replaced by release process

While the main classes of the KOMA‑Script bundle were made, there where several ideas for formating the table of contents and lists of floats, but almost none of them where implemented. One reason was, that the KOMA‑Script author didn't like to change the LaTeX kernel at a class, because this may result in serveral problems with other packages. The package tocstyle will fill the gap. If it conflicts with another package, you simply may decide not to use it.

Contents

---

*This is version v0.2e-alpha of file `tocstyle.dtx`.

1. How It Works

Loading the package tocstyle will redefine the kernel macro `\@starttoc`. Using the redefined `\@starttoc` will redefine `\@dottedtocline`, `\l@part` down to `\l@subparagraph`, `\l@figure`, and `\l@table`, if and only if tocstyle wasn't deactivated for all TOCs or this TOC. Usage the redefined `\@dottedtocline` will redefine `\numberline`.

Redefining `\@starttoc`, `\@dottedtocline`, and `\numberline` will activate the features of tocstyle for all lists that uses these, e.g. table of contents, list of figures and list of tables at the standard or the KOMA-Script classes. But while not all classes uses `\@dottedtocline` and `\@numberline` for all entries to table of contents and list of floats the package redefines some other macros that are typically used for those entries. These are e.g.

`\l@part`, `\l@chapter` and some more. If the class even does not use those macros, you may not use tocstyle to change the lists. The term TOC will be used for all kind of list, that may be processed by tocstyle. The package tests wether the original kernel macros `\@starttoc`, `\@dottedtocline`, and `\numberline` were used or not and warns if not.

Package tocstyle needs some more information. For the standard and the KOMA-Script classes these informations may be detected by the package. If the result is not the expected, you may configure these informations manually.

The entries of every TOC hat a depth. See the counter `tocdepth` for more information about the depth. You may change several settings for the entries of either all depths of all TOCs, all depths of one TOC, or one depth of one TOC.

But most users will not need to set up tocstyle at this low level. They simply will select one of the predefined styles and maybe select one of the optional features.

2. Optional Features

Optional features will be selected using a package option while loading the package or using the package option as a global option loading the class using `\documentclass`. Optional features change generall behaviour of all TOCs.

`tocindentauto`
`tocindentmanual`

With option `tocindentauto` all widths at the TOCs are calculated by tocstyle. The calculation of the width needs at least one LaTeX run with all TOC entries. So you need at least three LaTeX runs:

- one to write all the TOC entries to the TOC file

- one with the known TOC entries from the TOC file but unknown widths

- one with the known TOC entries from the TOC file and known widths

If the TOC entries changed between the second and the third run — e.g. because of page numbers changed — you'll need one more run (and so on).

Note: The widths of all entries of same depth and same TOC are same. Don't ask for less width of page numbers at the first than the last TOC page!

`tocgraduated`
`tocflat`
`tocfullflat`

The option `tocgraduated` selects the graduated version of all TOCs. You know the graduated version from the standard classes. Entries of lower depth are indented against entries of higher depth. This may e.g. look like:

The option `tocflat` selects the flat — aka left aligned — version of all TOCs. You know the flat version from the KOMA-Script classes using option `tocleft`. This may e.g. look like:

The option `tocfullflat` is similar to flat version of all TOCs, but there is even no box of same width for the numbers of all entries. This may e.g. look like:

Default is option `tocgraduated`.

`tocbreaksstrict`
`tocbreakscareless`

Default option `tocbreaksstrict` sets a lot of penalties before and after TOC entries to avoid page break between a TOC entry and it's parent. But sometimes you may like to allow more page breaks. You may use option `tocbreakscareless` for this.

`toctextentriesindented`
`toctextentriesleft`

With default option `toctextentriesleft` unnumbered TOC entries, e.g. from KOMA-Script command `\addchap`, are indented only as wide as the number of numbered TOC entries of the same level are. But with option `toctextentriesindented` these are indented as if they have an empty number.

3. Using TOC Styles

Package tocstyle hat several predefined toc styles. Most users will never need to define their own toc style but only select one of the predefined and maybe configure it by one of the options described at the previsous section.

`\usetocstyle`

You may set the style of one or all TOCs. If you want to set the style of all TOCs, you'd simply say `\usetocstyle{⟨style⟩}`. This will set all settings of the given style to all TOCs. Individual settings will overwrite this generall setting.

If you use `\usetocstyle[⟨TOC⟩]{⟨style⟩}`, only the style of the given TOC will be set. This will be done *after* the general setting. Only individual settings of single features may overwrite the setting of the style.

The table 1 shows the predefined styles, that may be used as mandatory argument of `\tocstyle`. The optional argument ⟨TOC⟩ is the shortcut (file extension) of the TOC. Examples of known shortcuts are shown at table 2.

Note: Before you're setting a style the style of the TOCs are unspecified. This means that some entries may be set using tocstyle others may not.

`\deactivatetocstyle`
`\reactivatetocstyle`

Both commands have one optional argument ⟨TOC⟩. You may deactive the influence of tocstyle for a TOC and reactivate it. If you use `\deactivatetocstyle` without the optional argument or empty optional argument, the influence of tocstyle for all TOCs will be deactivated and may reactivated only using `\reactivatetocstyle` without the optional argument or empty optional argument too.

After deactivation of tocstyle for one TOC or all TOCs you may continue configuring TOCs. All these changes will be used after reactivation.

4. Setting-up Single Features

At the previous section you've learned how to select a predefined TOC style. You were also told, that you may change one ore more features against the

Table 1: Predefined TOC Styles

`standard`
> A style similar to the standard classes. All width are predefined to the width of the standard classes, but may be overwritten by the general options (see section 2). The depth -1 (part) and 0 (chapter) are set in bold face (`\bfseries`). If no depth 0 was found at the TOC, depth 1 (section) will be set in bold face, too. All other depth will be set in normal font. Depth -1 (part) will be set using `\large`. The font changes are valid for the page numbers, too.

`KOMAlike`
> A style similar to the KOMA-Script classes. This is almost the same like `standard`, but instead of bold face `\usekomafont{disposition}` will be used if `\usekomafont` was defined and sans serif, bold face (`\sffamily\bfseries`) if not.

`classic`
> Like `KOMAlike` but all page numbers are set using normal font.

`allwithdot`
> Like `classic` but dots between entry text and page numbers are used at all depths.

`noonewithdot`
> Like `classic` but not dots between entry text and page numbers are used.

`nopagecolumn`
> Like `noonewithdot` but also the gap between text and page numbers is omited. This means, that the page numbers are set 1 em after the text.

Table 2: Known TOC Shortcuts

`toc`
> Table of contents of almost all known classes.

`lof`
> List of figures of almost all known classes.

`lot`
> List of tables of almost all known classes.

`lol`
> List of listings of package listings. Currently the usability of listings with tocstyle is not recomended. Maybe it works, maybe not. Maybe you should try `\deactivatetocstyle[lol]`.

Table 3: Features that May Be Set

`dothook`
  will be executed before any dot of the dot line
`entryhook`
  hook before the entry will be set
`entryvskip`
  initial vertical skip amount (if not set 0pt plus .2pt will be used)
`leaders`
  commands for fillin the gap between entry text and page number (if not set the default leaders command with dots will be used)
`pagenumberbox`
  the box command for setting the page number (if not set the default box of with `\@pnumwidth` will be used); note, that this has to be a command with exactly one argument
`pagenumberhook`
  hook before the page number will be set at the page number box
`parfillskip`
  add this amount to the default value of `\parfillskip` after setting up all lengths
`raggedhook`
  the only allowed values here are `\raggedright` or nothing
`spaceafternumber`
  amount of minimum space after the entry number, if the needed width will be calculated automaticly

---

used predefined TOC style for one or all depth of one ore all TOCs. Now you will learn how to do this.

`\settocfeature`
`\settocstylefeature`
These commands are used to set a single feature eiher of all depth of all TOCs (`\settocfeature {⟨feature⟩} {⟨command⟩}` or `\settocstylefeature {⟨feature⟩} {⟨commands⟩}`), or of all depth of a single TOC (`\settocfeature [⟨TOC⟩] {⟨feature⟩} {⟨commands⟩}`), or of a single depth of all TOCs (`\settocstylefeature [⟨depth⟩] {⟨feature⟩} {⟨commands⟩}`), or of a single depth of a single TOC (`\settocfeature [⟨TOC⟩] [⟨depth⟩] {⟨feature⟩} {⟨commands⟩}`).

Parameter ⟨commands⟩ is a list of commands. In most cases these must not be commands, that need an argument. So you should e.g. not use `\textbf` but `\bfseries` to switch to bold face. Parameter ⟨feature⟩ is the feature, that may be configured with parameter ⟨commands⟩. All known features are show at table 3.

The order of used commands for a feature is

1. commands for all depths of all TOCs,

8

2. commands for all depth of a single TOC,

3. commands for a single depth of all TOCs,

4. commands for a single depth of a single TOC,

and settings of \usetocstyle may be overwritten by \settocfeature und \settocstylefeature.

5. Defining New TOC Styles

Now you know how to select a prefedined TOC style and how to change single features. But wouldn't it be nice to define your own TOC style?

\newtocstyle   You may do this using \newtocstyle[⟨*parent style*⟩] [⟨*exclude features*⟩] {⟨*style name*⟩} {⟨*\settocstylefeature-commands*⟩}. If you used the optional argument ⟨*parent style*⟩ all features of the parent style will be part of the new style, before overwriting them with the features of the ⟨*\settocstylefeature-commands*⟩. You should not use any other commands at the last argument. But at \newtocstyle the command \settocfeature becomes an alias for \settocstylefeature to avoid to much mistakes.

The second optional argument is a comma seperated list of feature names. If it is used, these features of the parent style (and all ancestors of the parent) will not be part of the new style.

\aliastoc   Using \aliastoc{⟨*original-TOC*⟩}{⟨*alias-TOC*⟩} you may define an alias for a TOC. The first argument is the original TOC for that the second argument should be the alias. An alias-TOC will be processed with all settings, that were done for the original-TOC. Internally this command is used as default for the optional, first argument of \showtoc.

6. Processing a TOC

While LaTeX inputs a toc file it processes the commands of the TOC. These commands mainly produce the entries of the toc. Some commands are only available or valid while a TOC is processed. But be carefull: Some of these are read-only commands. Changing such a read-only command may result in various errors!

\@starttoc   The internal comand \@starttoc is defined by the LaTeX kernel. It is used
\showtoc   by package and class authors to build commands like \tableofcontents or \listoffigures. Without using it you will not get a toc file. tocstyle redefines it, to add pre- and post-processing commands. The original definition found by tocstyle will be used inside the redefinition.

\showtoc [⟨*preprocession*⟩]{⟨*TOC*⟩} is an addition of tocstyle. Using it will procude a copy of TOC and process this copy. The copy

9

will be done just after creating the original TOC. The copy will be an alias for the original file. The extension of the copy is the generated alias if ⟨*TOC*⟩. You may generate the alias using `\aliastoc` at the optional argument of `\showtoc`. The default for this optional argument will be `\aliastoc\tocstyleTOC\tocstyleAliasTOC` and the default alias `\tocstyleAliasTOC` will be `\tocstyleTOC` extended by a number. The first TOC example at section 2 was made using

```
\showtoc[{%
  \aliastoc{\tocstyleTOC}{toc}%
  \usetocstyle[toc]{standard}%
  \settocfeature[toc]{raggedhook}{\raggedright}%
  \selecttocstyleoption{tocgraduated}%
}]{toc}
```

If you want to show a copy the table of contents, that shows only depth 1 of the headlines you may simply use:

```
\showtoc[{%
  \expandafter\value{tocdepth}=1\relax
  \aliastoc{\tocstyleTOC}{toc}%
}]{toc}
```

or

```
\newcounter{savedtocdepth}
\setcounter{savedtocdepth}{\value{tocdepth}}
\setcounter{tocdepth}{1}
\showtoc{toc}
\setcounter{tocdepth}{\value{savedtocdepth}}
```

`\tocstyleTOC` These are read-only macros. While processing a TOC using `\@starttoc`
`\tocstyleAliasTOC` or `\showtoc`, `\tocstyleAliasTOC` is the shortcut, that is valid for the features and `\tocstyleTOC` is valid for the file extension to be used.

`\tocstyledepth` This is a read-only macro. While processing a single toc entry with `\@dottedtocline` this is the depth (first argument of `\@dottedtocline`) of this entry. Most users will never need this, but it is often used internaly. Because of this *you should never change it!*

`\iftochasdepth` Using `\iftochasdepth{`⟨*TOC*⟩`}{`⟨*depth*⟩`}{`⟨*true*⟩`}` `{`⟨*false*⟩`}` you may test, if an entry of a given depth was already output to a TOC. If so the commands of argument ⟨*true*⟩ will be processed. If not so the commands of argument ⟨*false*⟩ will be processed.

7. Configuration file

There's another feature for new toc styles. If there's a file `tocstyle.cfg` it will be loaded at the end of the package. This is usefull to define your own toc styles.

8. Implementation

```
1 \PackageWarningNoLine{tocstyle}{%
2   THIS IS AN ALPHA VERSION!\MessageBreak
3   USAGE OF THIS VERSION IS ON YOUR OWN RISK!\MessageBreak
4   EVERYTHING MAY HAPPEN!\MessageBreak
5   EVERYTHING MAY CHANGE IN FUTURE!\MessageBreak
6   THERE IS NO SUPPORT, IF YOU USE THIS PACKAGE!\MessageBreak
7   Maybe it would be better, not to load this package%
8 }
```

8.1. Option

Options change general behaviour of TOCs.

\selecttocstyleoption

```
9  \newif\if@tocstyle@penalties
10 \newif\iftocstyle@autolength
11 \newif\iftocstyle@indentnotnumbered
12 \newcount\tocstyle@indentstyle\tocstyle@indentstyle=\z@
13 \newcommand*{\selecttocstyleoption}[1]{%
14   \begingroup
15     \edef\@tempa{#1}%
16     \edef\@tempb{tocbreaksstrict}%
17     \ifx\@tempa\@tempb\aftergroup\@tocstyle@penaltiestrue\else
18       \edef\@tempb{tocbreakscareless}%
19       \ifx\@tempa\@tempb\aftergroup\@tocstyle@penaltiesfalse\else
20         \edef\@tempb{tocindentauto}
21         \ifx\@tempa\@tempb\aftergroup\tocstyle@autolengthtrue\else
22           \edef\@tempb{tocindentmanual}%
23           \ifx\@tempa\@tempb\aftergroup\tocstyle@autolengthfalse\else
24             \edef\@tempb{tocgraduated}%
25             \ifx\@tempa\@tempb
26               \aftergroup\tocstyle@indentstyle\aftergroup\z@
27             \else
28               \edef\@tempb{tocflat}%
29               \ifx\@tempa\@tempb
30                 \aftergroup\tocstyle@indentstyle\aftergroup\@ne
31                 \aftergroup\relax
32               \else
33                 \edef\@tempb{tocfullflat}%
34                 \ifx\@tempa\@tempb
35                   \aftergroup\tocstyle@indentstyle\aftergroup\tw@
36                   \aftergroup\relax
37                 \else
38                   \edef\@tempb{toctextentriesindented}%
39                   \ifx\@tempa\@tempb\aftergroup\tocstyle@indentnotnumberedtrue
40                   \else
41                     \edef\@tempb{toctextentriesleft}%
42                     \ifx\@tempa\@tempb
```

```
43                        \aftergroup\tocstyle@indentnotnumberedfalse
44                    \else
45                    \PackageError{tocstyle}{unknown option '#1'}{%
46                        You've told me to select toc style option
47                        '#1',\MessageBreak
48                        but tocstyle doesn't know an option named '#1'}%
49                \fi
50               \fi
51             \fi
52           \fi
53          \fi
54         \fi
55        \fi
56      \fi
57     \fi
58   \endgroup
59 }
```

chapter
nochapter
\iftochaschapter

Do we have \chapter and \l@chapter?

```
60 \newif\iftochaschapter\tochaschapterfalse
61 \ifcsname l@chapter\endcsname
62   \ifcsname chapter\endcsname
63     \tochaschaptertrue
64   \fi
65 \fi
```

tocbreaksstrict
tocbreakscareless

Switch on extended pernalties.

```
66 \DeclareOption{tocbreaksstrict}{\selecttocstyleoption\CurrentOption}
67 \DeclareOption{tocbreakscareless}{\selecttocstyleoption\CurrentOption}
```

tocindentauto
tocindentmanual

```
68 \DeclareOption{tocindentauto}{\selecttocstyleoption\CurrentOption}
69 \DeclareOption{tocindentmanual}{\selecttocstyleoption\CurrentOption}
```

toctextentriesindented
toctextentriesleft

```
70 \DeclareOption{toctextentriesindented}{\selecttocstyleoption\CurrentOption}
71 \DeclareOption{toctextentriesleft}{\selecttocstyleoption\CurrentOption}
```

tocgraduated
tocflat
tocfullflat

```
72 \DeclareOption{tocgraduated}{\selecttocstyleoption\CurrentOption}
73 \DeclareOption{tocflat}{\selecttocstyleoption\CurrentOption}
74 \DeclareOption{tocfullflat}{\selecttocstyleoption\CurrentOption}
```

Defaults and others:

```
75 \ExecuteOptions{tocbreaksstrict,tocindentauto,tocgraduated,%
76   toctextentriesleft}
77 \ProcessOptions\relax
```

```
78 \ifcsname if@tocleft\endcsname
79   \expandafter\let\csname if@tempswa\expandafter\endcsname
80   \csname if@tocleft\endcsname
81 \else
82   \@tempswafalse
83 \fi
84 \if@tempswa
85   \PackageWarningNoLine{tocstyle}{%
86     You should not use class option 'toc=flat'!\MessageBreak
87     This may result in errors or unexpected results.\MessageBreak
88     I'll try to deactivate 'toc=flat', now.\MessageBreak
89     You may use package options 'tocflat' and\MessageBreak
90     'tocindentauto' instead of 'toc=flat'}%
91   \csname @tocleftfalse\endcsname
92 \fi
```

8.2. Body

There are two parts at tocstyle:

- redefining internal LaTeX kernel macros,

- defining new macros and redefining class macros.

Redefining LaTeX kernel macros may not be switched of. But redefining class macros will only be on demand.

8.2.1. Redefining LaTeX Kernel Macros

Some LaTeX kernel macros must be redefined to add the new functionality. Before redefining them, we test against the definition at kernel 2005/12/01

\@starttoc The original definition will be extended by defaults for \parskip, \parindent
\tocstyle@saved@@starttoc and \parfillskip and storage of the shortcut of the current TOC.

```
93 \newcommand*\tocstyle@saved@starttoc{}
94 \let\tocstyle@saved@starttoc\@starttoc
95 \renewcommand*{\@starttoc}[1]{%
96   \tocstyle@pre@starttoc{#1}%
97   \tocstyle@saved@starttoc{#1}%
98   \tocstyle@post@starttoc{#1}%
99 }
```

\tocstyle@saved@dottedtocline For saving the unchanged definition (at \begindocument):

```
100 \newcommand*{\tocstyle@saved@dottedtocline}{}
```

\tocstyle@dottedtocline Implement new definition and redefine:

```
101 \newcommand*{\tocstyle@dottedtocline}[5]{%
102   \let\numberline\tocstyle@numberline
103   \ifnum #1>\c@tocdepth \else
```

13

Penalty feature: no page break between higher and lower depths.

```
104        \if@tocstyle@penalties
105          \begingroup
106            \@tempcnta 20010
107            \advance \@tempcnta by -#1
108            \ifnum \@tempcnta>\lastpenalty
109              \aftergroup\penalty\aftergroup\@lowpenalty
110            \fi
111          \endgroup
112        \fi
```

Activation of all features for this TOC and depth:

```
113        \edef\tocstyledepth{#1}%
114        \tocstyle@activate@features
```

Similar to kernel command but if feature `entryvskip` was set use `\addvspace`:

```
115        \ifx\tocstyle@feature@entryvskip\relax
116          \vskip \z@ \@plus.2\p@
117        \else
118          \addvspace{\tocstyle@feature@entryvskip}%
119        \fi
120        {%
```

Preinitialization of lengths and skips and then call a hook

```
121          \parskip \z@ \parindent \z@ \leftskip \z@ \rightskip \z@
122          \tocstyle@feature@raggedhook
```

Set number indent to `\@tempdimb` and text indent to `\@tempdima`.

```
123          \@tempdima #3\relax
124          \@tempdimb #2\relax
125 ⟨trace⟩        \typeout{number indent by \string\l@... (\tocstyleTOC, \tocstyledepth
126 ⟨trace⟩        \typeout{text indent by \string\l@... (\tocstyleTOC, \tocstyledepth):
```

Calc auto lengths. Use max. of last run of parents if available.

```
127          \ifnum #1>\z@\relax
128            \@tempcnta #1\relax \advance\@tempcnta \m@ne
129            \ifcsname tocstyle@maxskipwidth@\tocstyleTOC @\the\@tempcnta\endcsname
130              \ifcsname tocstyle@maxnumwidth@\tocstyleTOC @\the\@tempcnta\endcsname
131                \@tempdimb
132                \csname tocstyle@maxskipwidth@\tocstyleTOC @\the\@tempcnta\endcsname
133                \advance\@tempdimb
134                \csname tocstyle@maxnumwidth@\tocstyleTOC @\the\@tempcnta\endcsname
135              \fi
136            \fi
137          \fi
138 ⟨trace⟩        \typeout{number indent by parent (\tocstyleTOC, \tocstyledepth): \spa
139          \ifcsname tocstyle@skipwidth@\tocstyleTOC @#1\endcsname
140            \ifdim \@tempdimb>
141            \csname tocstyle@skipwidth@\tocstyleTOC @#1\endcsname\relax
142              \expandafter\xdef\csname tocstyle@skipwidth@\tocstyleTOC
143              @#1\endcsname{\the\@tempdimb}%
```

```
144            \fi
145          \else
146            \expandafter\xdef\csname tocstyle@skipwidth@\tocstyleTOC
147            @#1\endcsname{\the\@tempdimb}%
148          \fi
149          \iftocstyle@autolength
150            \ifcsname tocstyle@maxskipwidth@\tocstyleTOC @#1\endcsname
151              \@tempdimb \csname tocstyle@maxskipwidth@\tocstyleTOC @#1\endcsname
152              \relax
153            \fi
154            \ifcsname tocstyle@maxnumwidth@\tocstyleTOC @#1\endcsname
155              \@tempdima \csname tocstyle@maxnumwidth@\tocstyleTOC @#1\endcsname
156              \relax
157            \fi
158 ⟨trace⟩         \typeout{text indent calculated (\tocstyleTOC, \tocstyledepth): \th
159 ⟨trace⟩         \typeout{number indent calculated (\tocstyleTOC, \tocstyledepth): \
160          \else
161            \@tempdimb #2\relax
162 ⟨trace⟩         \typeout{number indent explicite (\tocstyleTOC, \tocstyledepth): \t
163          \fi
164          \ifcsname tocstyle@unumwidth@\tocstyleTOC @\endcsname
165            \ifdim \@tempdima>
166            \csname tocstyle@unumwidth@\tocstyleTOC @\endcsname\relax
167              \expandafter\xdef\csname tocstyle@unumwidth@\tocstyleTOC
168              @\endcsname{\the\@tempdima}%
169            \fi
170          \else
171            \expandafter\xdef\csname tocstyle@unumwidth@\tocstyleTOC
172            @\endcsname{\the\@tempdima}%
173          \fi
174          \ifcase\tocstyle@indentstyle\relax\else
175            \@tempdimb \z@
176            \ifcsname tocstyle@maxunumwidth@\tocstyleTOC @\endcsname
177              \@tempdima \csname tocstyle@maxunumwidth@\tocstyleTOC @\endcsname
178              \relax
179            \fi
180 ⟨trace⟩         \typeout{text noindent (\tocstyleTOC, \tocstyledepth): \the\@tempdi
181 ⟨trace⟩         \typeout{number noindent (\tocstyleTOC, \tocstyledepth): \the\@temp
182          \fi
```

Advance instead of set, because of the hook above:

```
183          \advance\parindent \@tempdimb\@afterindenttrue
184          \advance\leftskip \parindent
185          \advance\rightskip \@tocrmarg
186          \parfillskip -\rightskip
187          \ifx\tocstyle@feature@parfillskip\relax\else
188            \advance\parfillskip \tocstyle@feature@parfillskip\relax
189          \fi
190          \interlinepenalty\@M
191          \leavevmode
```

```
192        \advance\leftskip \@tempdima
193        \null\nobreak
```

\hskip\-leftskip optional moved to \numberline

```
194        \iftocstyle@indentnotnumbered\else
195          \hskip -\leftskip
196        \fi
```

Change at start of the entry

```
197        \tocstyle@feature@entryhook
```

Similar to kernel command but if feature leaders was set use this instead of the default leaders. And if feature dothook was set (default is \normalfont) use this at the default leaders.

```
198        {#4}\nobreak
199        \ifx\tocstyle@feature@leaders\relax
200          \leaders\hbox{$\m@th
201            \mkern \@dotsep mu\hbox{\tocstyle@feature@dothook .}%
202            \mkern \@dotsep mu$}\hfill
203        \else
204          \tocstyle@feature@leaders
205        \fi
206        \nobreak
207        \ifx\tocstyle@feature@pagenumberbox\relax
208          \hb@xt@\@pnumwidth{\hfil\tocstyle@feature@pagenumberhook #5}%
209        \else
210          \tocstyle@feature@pagenumberbox{\tocstyle@feature@pagenumberhook #5}%
211        \fi
212        \par
213      }%
```

Last change is, another penalty change:

```
214      \if@tocstyle@penalties
215        \bgroup
216          \@tempcnta 20009
217          \advance\@tempcnta by -#1
218          \edef\reserved@a{\egroup\penalty\the\@tempcnta\relax}%
219        \reserved@a
220      \fi
221    \fi}
```

\tocstyle@saved@numberline Define a new \numberline, that will do all the job after \begindocument
\tocstyle@numberline and one to save the original definition.

```
222 \newcommand*{\tocstyle@saved@numberline}{}
223 \newcommand*{\tocstyle@numberline}[1]{%
224   \begingroup
225     \ifx\tocstyle@feature@spaceafternumber\relax
226       \settowidth\@tempdima{\tocstyle@@numberline{#1}\enskip}%
227     \else
228       \settowidth\@tempdima{\tocstyle@@numberline{#1}}%
```

```
229        \advance \@tempdima \tocstyle@feature@spaceafternumber\relax
230      \fi
231      \ifcsname tocstyle@numwidth@\tocstyleTOC @\tocstyledepth\endcsname
232        \ifdim \@tempdima >
233        \csname tocstyle@numwidth@\tocstyleTOC @\tocstyledepth\endcsname\relax
234          \expandafter\xdef\csname tocstyle@numwidth@\tocstyleTOC
235          @\tocstyledepth\endcsname{\the\@tempdima}%
236        \fi
237      \else
238        \expandafter\xdef\csname tocstyle@numwidth@\tocstyleTOC
239        @\tocstyledepth\endcsname{\the\@tempdima}%
240      \fi
241    \endgroup
242    \iftocstyle@indentnotnumbered
243      \hskip -\leftskip
244    \fi
245    \ifcase \tocstyle@indentstyle
246      \hb@xt@\@tempdima{\tocstyle@@numberline{#1}\hfil}%
247    \or
248      \hb@xt@\@tempdima{\tocstyle@@numberline{#1}\hfil}%
249    \else
250      \ifx\tocstyle@feature@spaceafternumber\relax
251        \hbox{\tocstyle@@numberline{#1}\enskip}%
252      \else
253        \hbox{\tocstyle@@numberline{#1}\hskip
254          \tocstyle@feature@spaceafternumber\relax}%
255      \fi
256    \fi
257 }
```

\tocstyle@@numberline Do the main work!

```
258 \newcommand*{\tocstyle@@numberline}[1]{%
259   #1\csname autodot\endcsname
260 }
```

8.2.2. Redefining Class Macros

\l@part
\l@chapter
\l@section
\l@subsection
\l@subsubsection
\l@paragraph
\l@subparagraph
\l@table
\l@figure

Try to redefine the toc commands at startup.

```
261 \AtBeginDocument{%
262   \ifcsname l@part\endcsname
263     \setbox\@tempboxa\vbox{\hsize\maxdimen
264       \l@part{\tocstyle@l@define{part}{-1}}{}}%
265   \fi
266   \ifcsname l@chapter\endcsname
267     \setbox\@tempboxa\vbox{\hsize\maxdimen
268       \l@chapter{\tocstyle@l@define{chapter}{0}}{}}%
269   \fi
270   \ifcsname l@section\endcsname
271     \setbox\@tempboxa\vbox{\hsize\maxdimen
```

```
272        \l@section{\tocstyle@l@define{section}{1}}{}}%
273     \fi
274     \ifcsname l@subsection\endcsname
275        \setbox\@tempboxa\vbox{\hsize\maxdimen
276          \l@subsection{\tocstyle@l@define{subsection}{2}}{}}%
277     \fi
278     \ifcsname l@subsubsection\endcsname
279        \setbox\@tempboxa\vbox{\hsize\maxdimen
280          \l@subsubsection{\tocstyle@l@define{subsubsection}{3}}{}}%
281     \fi
282     \ifcsname l@paragraph\endcsname
283        \setbox\@tempboxa\vbox{\hsize\maxdimen
284          \l@paragraph{\tocstyle@l@define{paragraph}{4}}{}}%
285     \fi
286     \ifcsname l@subparagraph\endcsname
287        \setbox\@tempboxa\vbox{\hsize\maxdimen
288          \l@subparagraph{\tocstyle@l@define{subparagraph}{5}}{}}%
289     \fi
290     \ifcsname l@table\endcsname
291        \setbox\@tempboxa\vbox{\hsize\maxdimen
292          \l@table{\tocstyle@l@define{table}{1}}{}}%
293     \fi
294     \ifcsname l@figure\endcsname
295        \setbox\@tempboxa\vbox{\hsize\maxdimen
296          \l@figure{\tocstyle@l@define{figure}{1}}{}}%
297     \fi
```

\@dottedtocline   This will be used even for undotted toc lines. First check the definition,
then redefine.

```
298     \def\@tempa#1#2#3#4#5{%
299        \ifnum #1>\c@tocdepth \else
300          \vskip \z@ \@plus.2\p@
301          {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
302           \parindent #2\relax\@afterindenttrue
303           \interlinepenalty\@M
304           \leavevmode
305           \@tempdima #3\relax
306           \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
307           {#4}\nobreak
308           \leaders\hbox{$\m@th
309              \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
310              mu$}\hfill
311           \nobreak
312           \hb@xt@\@pnumwidth{\hfil \normalfont \normalcolor #5}%
313           \par}%
314        \fi}%
315     \ifx\@dottedtocline\@tempa\else
316        \tocstyle@macrochangewarning\@dottedtocline
317     \fi
```

\numberline This macro needed to be redefined to calculate the with of the numbers. First of all: check the definition. This is a bit more difficult, because of respecting KOMA-Script:

```
319    \def\@tempa#1{\hb@xt@\@tempdima{#1\autodot\hfil}}%
320    \ifx\numberline\@tempa\else
321      \def\@tempa#1{\hb@xt@\@tempdima{#1\hfil}}%
322      \ifx\numberline@tempa\else
323        \tocstyle@macrochangewarning\numberline
324      \fi
325    \fi
326    \let\tocstyle@saved@numberline\numberline
327 }
```

\tocstyle@macrochangewarning

```
328 \newcommand*{\tocstyle@macrochangewarning}[1]{%
329    \PackageWarningNoLine{tocstyle}{%
330      unexpected \string#1\space definition!\MessageBreak
331      You are either using an unknown LaTeX kernel\MessageBreak
332      version, an unknown class or package, that redefines\MessageBreak
333      \string#1, or a \string#1\space
334      redefinition\MessageBreak
335      at the document preamble.\MessageBreak
336      Because of this you may get unexpected results!\MessageBreak
337      Maybe it would be better not to use package tocstyle}%
338    \PackageInfo{tocstyle}{Unexpected definition is:\MessageBreak
339      \meaning#1}%
340 }
```

\tocstyle@l@define
\tocstyle@activate@all@l

```
341 \newcommand*{\tocstyle@activate@all@l}{}
342 \newcommand*{\tocstyle@l@define}[2]{%
343    \advance\leftskip-\@tempdima
344    \edef\@tempa{%
345      \noexpand\global\noexpand\let
346      \expandafter\noexpand\csname tocstyle@saved@l@#1\endcsname
347      \expandafter\noexpand\csname l@#1\endcsname
348      \noexpand\gdef
349      \expandafter\noexpand\csname tocstyle@l@#1\endcsname{%
350        \noexpand\@dottedtocline{#2}{\the\leftskip}{\the\@tempdima}}%
351      \noexpand\g@addto@macro\noexpand\tocstyle@activate@all@l{%
352        \noexpand\let\expandafter\noexpand\csname l@#1\endcsname
353        \expandafter\noexpand\csname tocstyle@l@#1\endcsname
354      }%
355    }%
356    \PackageInfo{tocstyle}{prepare \expandafter\string
357      \csname l@#1\endcsname\space for redefinition}%
358    \@tempa
```

```
359 }
```

8.2.3. New Macros

`\showtoc`

```
360 \newcommand*{\showtoc}[2][\aliastoc\tocstyleTOC\tocstyleAliasTOC]{%
361   \ifcsname tocstyle@copyname@#2\endcsname
362     \@tempcnta \csname tocstyle@copyname@#2\endcsname\relax
363     \advance\@tempcnta \@ne
364     \expandafter\xdef\csname tocstyle@copyname@#2\endcsname{\the\@tempcnta}%
365   \else
366     \expandafter\xdef\csname tocstyle@copyname@#2\endcsname{1}%
367   \fi
368   \ifx\@dofilelist\relax\let\@dofilelist\@empty\fi
369   \edef\@tempa{\noexpand\g@addto@macro\noexpand\@dofilelist{%
370       \noexpand\tocstyle@copy@toc{#2}{\csname
371         tocstyle@copyname@#2\endcsname}}%
372   }\@tempa%
373   \begingroup
374     \edef\tocstyleAliasTOC{#2}%
375     \edef\tocstyleTOC{#2\csname tocstyle@copyname@#2\endcsname}%
376     #1
377     \tocstyle@pre@starttoc{#2\csname tocstyle@copyname@#2\endcsname}%
378     \makeatletter
379     \@input{\jobname.#2\csname tocstyle@copyname@#2\endcsname}%
380     \@nobreakfalse
381     \tocstyle@post@starttoc{#2\csname tocstyle@copyname@#2\endcsname}%
382   \endgroup
383 }
```

`\tocstyle@copy@toc`

```
384 \newcommand*{\tocstyle@copy@toc}[2]{%
385   \if@filesw
386     \begingroup
387       \endlinechar=\m@ne
388 % While \LaTeX{} does not close the files, we have to do it know.
389       \immediate\closeout\csname tf@#1\endcsname
390       \immediate\openin\@inputcheck \jobname.#1
391       \immediate\openout\@partaux \jobname.#1#2
392       \loop\unless\ifeof\@inputcheck
393         \immediate\readline\@inputcheck to \@tempa
394         \immediate\write\@partaux{\@tempa}%
395       \repeat
396       \immediate\closeout\@partaux
397       \immediate\closein\@inputcheck
398     \endgroup
399   \fi
400 }
```

`\aliastoc` Internal use not the real TOC shortcut but another one.

```
401 \newcommand*{\aliastoc}[2]{%
402   \expandafter\edef\csname tocstyle@alias@TOC@#1\endcsname{#2}%
403 }
```

`\tocstyle@pre@starttoc` Commands before and after the original `\@starttoc`.
`\tocstyle@post@starttoc`
```
404 \newcommand*{\tocstyle@pre@starttoc}[1]{%
405   \begingroup
406     \expandafter\ifx\csname tocstyle@deactivated@\endcsname\relax
407       \expandafter\ifx\csname tocstyle@deactivated@#1\endcsname\relax\relax
408         \tocstyle@activetrue
409       \else
410         \tocstyle@activefalse
411       \fi
412     \else
413       \tocstyle@activefalse
414     \fi
415     \iftocstyle@active
416       \let\@dottedtocline\tocstyle@dottedtocline
417       \parskip \z@
418       \parindent \z@
419       \parfillskip \z@\@plus 1fil
420       \ifcsname tocstyle@alias@TOC@#1\endcsname
421         \edef\tocstyleAliasTOC{\csname tocstyle@alias@TOC@#1\endcsname}%
422       \else
423         \edef\tocstyleAliasTOC{#1}%
424       \fi
425       \edef\tocstyleTOC{#1}%
426       \tocstyle@activate@all@l
427     \fi
428 }
429 \newcommand*{\tocstyle@post@starttoc}[1]{%
430   \iftocstyle@active
431     \if@filesw
432       \ifcsname tocstyle@unumwidth@#1@\endcsname
433         \protected@write\@auxout{}{%
434           \protect\tocstyle@set@width{unum}{#1}{}{%
435             \csname tocstyle@unumwidth@#1@\endcsname}%
436         }%
437       \fi
438       \expandafter\let\expandafter\@tempa
439         \csname tocstyle@depthlist@#1\endcsname
440       \ifx\@tempa\relax\else
441         \expandafter\@for \expandafter\@tempa\expandafter:\expandafter=\@tempa
442         \do {%
443           \ifcsname tocstyle@numwidth@#1@\@tempa\endcsname
444             \protected@write\@auxout{}{%
445               \protect\tocstyle@set@width{num}{#1}{\@tempa}{%
446                 \csname tocstyle@numwidth@#1@\@tempa\endcsname}%
```

```
447                    }%
448                  \fi
449              \ifcsname tocstyle@skipwidth@#1@\@tempa\endcsname
450                  \protected@write\@auxout{}{%
451                    \protect\tocstyle@set@width{skip}{#1}{\@tempa}{%
452                      \csname tocstyle@skipwidth@#1@\@tempa\endcsname}%
453                  }%
454              \fi
455            }%
456          \fi
457        \fi
458      \fi
459    \endgroup
460 }
```

\tocstyle@set@width Some classes do not use \numberline. This may result in negativ widths (esp. negativ skips). Following special handling of negativ values improves the toc handling of the standard classes. Nevertheless indentation of not numbered entries does not work with such classes!

```
461 \newcommand*{\tocstyle@set@width}[4]{%
462   \iftocstyle@indentnotnumbered
463     \ifdim #4<\z@
464       \expandafter\gdef\csname tocstyle@max#1width@#2@#3\endcsname{%
465         \dimexpr #4/2\relax}%
466     \else
467       \expandafter\gdef\csname tocstyle@max#1width@#2@#3\endcsname{#4}%
468     \fi
469   \else
470     \ifdim #4<\z@
471       \expandafter\gdef\csname tocstyle@max#1width@#2@#3\endcsname{\z@}%
472     \else
473       \expandafter\gdef\csname tocstyle@max#1width@#2@#3\endcsname{#4}%
474     \fi
475   \fi
476 }
```

\tocstyleTOC   Shortcut of the current processed TOC. Empty outside of TOCs.
\tocstyleAliasTOC
```
477 \newcommand*{\tocstyleTOC}{}
478 \newcommand*{\tocstyleAliasTOC}{}
```

\tocstyledepth   Current depth of the current processed TOC entry.
```
479 \newcommand*{\tocstyledepth}{}
```

\deactivatetocstyle You may (de)activate all influence of tocstyle either for one or all TOCs.
\reactivatetocstyle
```
480 \newif\iftocstyle@active
481 \newcommand*{\deactivatetocstyle}[1][]{%
482   \expandafter\let\csname tocstyle@deactivated@#1\endcsname\@empty}
483 \newcommand*{\reactivatetocstyle}[1][]{%
484   \expandafter\let\csname tocstyle@deactivated@#1\endcsname\relax}
```

22

`\settocfeature`	The primary command to set the features of a depth of a TOC.

```
\settocfeature
\@settocfeature    485 \newcommand*{\@settocfeature}[1][]{%
\@@@settocfeature  486   \@ifnextchar[ {\@@settocfeature[{#1}]}{\@@settocfeature[{#1}][]}
                   487 }
                   488 \def\@@settocfeature[#1][#2]#3#4{%
                   489 ⟨trace⟩   \typeout{exclude: \tocstyle@feature@excludelist}%
                   490   \@expandtwoargs\in@{,#3,}{,\tocstyle@feature@excludelist,}%
                   491   \ifin@\else
                   492     \expandafter\ifcsname tocstyle@feature@#3\endcsname
                   493       \@namedef{tocstyle@feature@#3@#1@#2}{#4}%
                   494       \begingroup
                   495         \expandafter\let\expandafter\@tempa
                   496         \csname tocstyle@commandlist@#1\endcsname
                   497         \@expandtwoargs\in@{,tocstyle@feature@#3@#1@#2,}{,\@tempa,}%
                   498         \ifin@\let\@tempa\endgroup\else
                   499           \edef\@tempa{\endgroup
                   500             \noexpand\expandafter\noexpand\ifx
                   501             \noexpand\csname tocstyle@commandlist@#1\noexpand\endcsname\relax
                   502               \noexpand\expandafter\noexpand\expandafter\noexpand\expandafter
                   503               \noexpand\def
                   504             \noexpand\else
                   505               \noexpand\expandafter\noexpand\expandafter\noexpand\expandafter
                   506               \noexpand\l@addto@macro
                   507             \noexpand\fi
                   508             \noexpand\csname tocstyle@commandlist@#1\noexpand\endcsname%
                   509             {tocstyle@feature@#3@#1@#2,}}%
                   510         \fi
                   511       \@tempa
                   512     \else
                   513       \PackageError{tocstyle}{unkown feature '#3'}{%
                   514         You've told me to set up toc style feature '#3',\MessageBreak
                   515         but I don't know this feature.\MessageBreak
                   516         See the tocstyle manual for all known feature.\MessageBreak
                   517       }%
                   518     \fi
                   519   \fi
                   520 }
                   521 \newcommand*{\settocfeature}{}
                   522 \let\settocfeature\@settocfeature
```

`\l@addto@macro`	Something like `\g@addto@macro` but only with local effect. While other packages or classes may also define this, `\providecommand` will be used.

```
                   523 \providecommand{\l@addto@macro}[2]{%
                   524   \edef#1{\unexpanded\expandafter{#1#2}}%
                   525 }%
```

`\settocstylefeature`	Same as above without TOC argument.

```
\@settocstylefeature  526 \newcommand*{\@settocstylefeature}{%
                      527   \@ifnextchar[ {\@settocfeature[]}{\@settocfeature[][]}}%
```

23

```
528 }
529 \newcommand*{\settocstylefeature}{}
530 \let\settocstylefeature\@settocstylefeature
```

Different commands will be defined:

`ocstyle@feature@!<feature!>@@` Global feature (all TOCs all depths).

`@feature@@!<feature!>@!<TOC!>@` All depth feature for one TOC.

`ature@@!<feature!>@@!<depth!>` All TOCs feature for one depth.

`!<feature!>@!<TOC!>@!<depth!>` One depth of one TOC feature.

`\tocstyle@activate@features` Activates the features

```
531 \newcommand*{\tocstyle@activate@features}{%
532   \expandafter\ifx\csname tocstyle@depthlist@\tocstyleTOC\endcsname\relax
533     \expandafter\xdef\csname tocstyle@depthlist@\tocstyleTOC\endcsname{%
534       \tocstyledepth}%
535   \else
536     \expandafter\let\expandafter\@tempa
537     \csname tocstyle@depthlist@\tocstyleTOC\endcsname
538     \@expandtwoargs\in@{,\tocstyledepth,}{,\@tempa,}%
539     \ifin@\else
540       \expandafter\xdef\csname tocstyle@depthlist@\tocstyleTOC\endcsname{%
541         \csname tocstyle@depthlist@\tocstyleTOC\endcsname,\tocstyledepth}%
542     \fi
543   \fi
544   \expandafter\@for \expandafter\@tempa
545   \expandafter:\expandafter=\tocstyle@featurelist \do
546   {%
547     \@ifundefined{tocstyle@feature@\@tempa @\tocstyleAliasTOC @\tocstyledepth}{%
548       \@ifundefined{tocstyle@feature@\@tempa @@\tocstyledepth}{%
549         \@ifundefined{tocstyle@feature@\@tempa @\tocstyleAliasTOC @}{%
550           \@ifundefined{tocstyle@feature@\@tempa @@}{%
551             \expandafter\let\csname tocstyle@feature@\@tempa\endcsname\relax
552           }{%
553             \expandafter\let\csname tocstyle@feature@\@tempa
554             \expandafter\endcsname
555             \csname tocstyle@feature@\@tempa @@\endcsname
556           }%
557         }{%
558           \expandafter\let\csname tocstyle@feature@\@tempa
559           \expandafter\endcsname
560           \csname tocstyle@feature@\@tempa @\tocstyleAliasTOC @\endcsname
561         }%
562       }{%
563         \expandafter\let\csname tocstyle@feature@\@tempa
564         \expandafter\endcsname
565         \csname tocstyle@feature@\@tempa @@\tocstyledepth\endcsname
```

```
566        }%
567      }{%
568        \expandafter\let\csname tocstyle@feature@\@tempa
569        \expandafter\endcsname
570        \csname tocstyle@feature@\@tempa @\tocstyleAliasTOC @\tocstyledepth\endcsna
571      }%
572    }%
573 }
```

\newtocstyle   Defining a new TOC style. First optional argument is a TOC style, that
                will be activated before the new definitions. Note that all new definitions
                will overwirte the parent's definitions. So a new TOC style, that defines
                all features doesn't need a parent.

```
574 \newcommand*{\newtocstyle}{%
575    \@ifnextchar [{\@newtocstyle}{\@newtocstyle[]}}
576 \newcommand*{\@newtocstyle}{}
577 \def\@newtocstyle[#1]{%
578    \@ifnextchar [{\@@newtocstyle[{#1}]}{\@newtocstyle[{#1}][]}}
579 \newcommand*{\@@newtocstyle}{}
580 \def\@@newtocstyle[#1][#2]#3#4{%
581    \@ifundefined{tocstyle@style@#3}{%
582      \@ifundefined{tocstyle@style@#1}{%
583        \ifx \relax#1\relax\else
584          \PackageError{tocstyle}{unknown parent TOC style '#1'}{%
585            You've told me to inheritate parent TOC style '#1',\MessageBreak
586            but there's no TOC style '#1' defined.}%
587        \fi
588        \expandafter\def\csname tocstyle@style@#3\endcsname{#4}%
589      }{%
590        \expandafter\def\csname tocstyle@style@#3\endcsname{%
591          \edef\reserved@a{%
592            \noexpand\l@addto@macro\noexpand\tocstyle@feature@excludelist{#2}%
593            \noexpand\@usetocstyle{#1}%
594            \noexpand\def\noexpand\tocstyle@feature@excludelist{%
595              \tocstyle@feature@excludelist}%
596          }\reserved@a
597          #4%
598        }%
599      }%
600    }{%
601      \PackageError{tocstyle}{TOC style '#3' already defined}{%
602        You've tried to define a new TOC style '#3',\MessageBreak
603        but there's already a TOC style named '#3'.}%
604    }%
605 }
606 \newcommand*{\tocstyle@feature@excludelist}{}
```

\usetocstyle   Use the predefined TOC style. You may define \tocstyle@deprecated@style@foo
\@usetocstyle   to mark TOC style foo to be deprecated. If \tocstyle@deprecated@style@foo

is `\@empty` TOC style `deprecated@foo` will be used instead almost silently. Otherwise TOC style `\tocstyle@deprecated@style@foo` will be used instead and the user will be told about this change.

```
607 \newcommand*{\usetocstyle}[2][]{%
608   \@ifundefined{tocstyle@deprecated@style@#2}{%
609     \@ifundefined{tocstyle@style@#2}{%
610       \PackageError{tocstyle}{unknown TOC style '#2'}{%
611         You've told me to use TOC style '#2',\MessageBreak
612         but there's no TOC style '#2' defined.}%
613     }{%
614       \def\settocfeature{%
615         \@ifnextchar[ %]
616         {\@@settocfeature[{#1}]}{\@@settocfeature[{#1}][]}%
617       }%
618       \let\settocstylefeature\settocfeature
```

Deactivate all known features for this TOC

```
619       \expandafter\ifx\csname tocstyle@commandlist@#1\endcsname\relax
620       \else
621         \expandafter\expandafter\expandafter\@for
622         \expandafter\expandafter\expandafter\@tempa
623         \expandafter\expandafter\expandafter:%
624         \expandafter\expandafter\expandafter=%
625         \csname tocstyle@commandlist@#1\endcsname
626         \do{%
627           \expandafter\let\csname \@tempa\endcsname\relax
628         }%
```

So there are no more known features for this TOC.

```
629         \expandafter\let\csname tocstyle@commandlist@#1\endcsname\relax
630       \fi
```

Activate all known features for this style and TOC

```
631       \@usetocstyle{#2}%
632       \let\settocfeature\@settocfeature
633       \let\settocstylefeature\@settocstylefeature
634     }%
635   }{%
636     \expandafter\ifx\csname tocstyle@deprecated@style@#2\endcsname\@empty
637       \PackageWarning{tocstyle}{%
638         deprecated TOC style '#2'!\MessageBreak
639         You should not longer use this style,\MessageBreak
640         because it will be removed soon.\MessageBreak
641         You should select another TOC style}%
642       \usetocstyle[{#1}]{deprecated@#2}%
643     \else
644       \PackageWarning{tocstyle}{%
645         deprecated TOC style '#2'!\MessageBreak
646         You should use TOC style '\csname
647         tocstyle@deprecated@style@#2\endcsname'\MessageBreak
```

```
648         instead of '#2'}%
649     \fi
650   }%
651 }
652 \newcommand*{\@usetocstyle}[1]{%
653   \csname tocstyle@style@#1\endcsname
654 }
```

\tocstyle@featurelist Comma seperated list of all known features

```
655 \newcommand*{\tocstyle@featurelist}{%
656   pagenumberhook,entryhook,dothook,entryvskip,leaders,raggedhook,%
657   spaceafternumber,parfillskip,pagenumberbox,%
658 }
```

\ocstyle@feature@pagenumberhook
\ocstyle@feature@pagenumberhook
\tocstyle@feature@entryhook
\tocstyle@feature@dothook
\tocstyle@feature@entryvskip
\tocstyle@feature@leaders
\tocstyle@feature@parfillskip
\tocstyle@feature@raggedhook
style@feature@spaceafternumber

```
659 \newcommand*{\tocstyle@feature@pagenumberhook}{}
660 \let\tocstyle@feature@pagenumberhook\relax
661 \newcommand*{\tocstyle@feature@pagenumberbox}{}
662 \let\tocstyle@feature@pagenumberbox\relax
663 \newcommand*{\tocstye@feature@entryhook}{}
664 \let\tocstyle@feature@entryhook\relax
665 \newcommand*{\tocstye@feature@dothook}{}
666 \let\tocstyle@feature@dothook\relax
667 \newcommand*{\tocstyle@feature@entryvskip}{}
668 \let\tocstyle@feature@entryvskip\relax
669 \newcommand*{\tocstye@feature@leaders}{}
670 \let\tocstyle@feature@leaders\relax
671 \newcommand*{\tocstye@feature@parfillskip}{}
672 \let\tocstyle@feature@parfillskip\relax
673 \newcommand*{\tocstye@feature@raggedhook}{}
674 \let\tocstyle@feature@raggedhook\relax
675 \newcommand*{\tocstye@feature@spaceafternumber}{}
676 \let\tocstyle@feature@spaceafternumber\relax
```

\iftochasdepth Uses \tocstyle@depthlist@⟨TOC⟩ to test, if the TOC has the depth already.

```
677 \newcommand*{\iftochasdepth}[2]{%
678   \begingroup
679     \expandafter\let\expandafter\@tempa\csname tocstyle@depthlist@#1\endcsname
680     \ifx\@tempa\relax
681       \aftergroup\@secondoftwo
682     \else
683       \@expandtwoargs\in@{,#2,}{,\@tempa}%
684       \expandafter\aftergroup\ifin@
685         \@firstoftwo
686       \else
687         \@secondoftwo
688       \fi
689     \fi
```

```
690    \endgroup
691 }
```

8.2.4. Defining Some TOC Styles

```
692 \newtocstyle{standard}{%
693    \settocfeature{dothook}{\normalfont}%
694    \settocfeature[-1]{entryhook}{\bfseries}%
695    \settocfeature[-1]{entryvskip}{2.25em\@plus\p@}%
696    \settocfeature[-1]{leaders}{\hfill}%
697    \settocfeature[0]{entryvskip}{1em\@plus\p@}%
698    \settocfeature[0]{leaders}{\hfill}%
699    \settocfeature[0]{entryhook}{\bfseries}
700    \iftochaschapter\else
701      \settocfeature[1]{entryvskip}{1em\@plus\p@}%
702      \settocfeature[1]{leaders}{\hfill}%
703      \settocfeature[1]{entryhook}{%
704        \begingroup
705          \edef\@tempa{toc}%
706          \ifx\tocstyleAliasTOC\@tempa\aftergroup\bfseries\fi
707        \endgroup
708      }%
709    \fi
710 }
711 \begingroup\expandafter\expandafter\expandafter\endgroup
712 \expandafter\ifx\csname KOMAClassName\endcsname\relax
713    \newtocstyle{KOMAlike}{%
714      \settocfeature{dothook}{\normalfont}%
715      \settocfeature[-1]{entryhook}{\sffamily\bfseries}%
716      \settocfeature[-1]{entryvskip}{2.25em\@plus\p@}%
717      \settocfeature[-1]{leaders}{\hfill}%
718      \settocfeature[-1]{pagenumberhook}{\sffamily\bfseries}%
719      \settocfeature[0]{entryvskip}{1em\@plus\p@}%
720      \settocfeature[0]{leaders}{\hfill}%
721      \settocfeature[0]{entryhook}{\sffamily\bfseries}
722      \settocfeature[0]{pagenumberhook}{\sffamily\bfseries}%
723      \iftochaschapter\else
724        \settocfeature[1]{entryvskip}{1em\@plus\p@}%
725        \settocfeature[1]{leaders}{\hfill}%
726        \settocfeature[1]{entryhook}{%
727          \begingroup
728            \edef\@tempa{toc}%
729            \ifx\tocstyleAliasTOC\@tempa
730              \aftergroup\sffamily\aftergroup\bfseries
731            \fi
732          \endgroup
733        }%
734        \settocfeature[1]{pagenumberhook}{%
735          \begingroup
```

```
736        \edef\@tempa{toc}%
737        \ifx\tocstyleAliasTOC\@tempa
738          \aftergroup\sffamily\aftergroup\bfseries
739        \fi
740      \endgroup
741    }%
742  \fi
743  }
744 \else
745  \newtocstyle{KOMAlike}{%
746    \settocfeature{dothook}{\normalfont}%
747    \settocfeature[-1]{entryhook}{\usekomafont{partentry}}%
748    \settocfeature[-1]{entryvskip}{2.25em\@plus\p@}%
749    \settocfeature[-1]{leaders}{\hfill}%
750    \settocfeature[-1]{pagenumberhook}{\usekomafont{partentrypagenumber}}%
751    \settocfeature[0]{entryvskip}{1em\@plus\p@}%
752    \settocfeature[0]{leaders}{\hfill}%
753    \settocfeature[0]{entryhook}{\usekomafont{chapterentry}}%
754    \settocfeature[0]{pagenumberhook}{\usekomafont{chapterentrypagenumber}}%
755    \iftochaschapter\else
756      \settocfeature[1]{entryvskip}{1em\@plus\p@}%
757      \settocfeature[1]{leaders}{\hfill}%
758      \settocfeature[1]{entryhook}{%
759        \begingroup
760          \edef\@tempa{toc}%
761          \ifx\tocstyleAliasTOC\@tempa
762            \def\@tempa{\endgroup\usekomafont{sectionentry}}%
763          \else
764            \let\@tempa\endgroup
765          \fi
766        \@tempa
767      }%
768      \settocfeature[1]{pagenumberhook}{%
769        \begingroup
770          \edef\@tempa{toc}%
771          \ifx\tocstyleAliasTOC\@tempa
772            \def\@tempa{\endgroup\usekomafont{sectionentrypagenumber}}%
773          \else
774            \let\@tempa\endgroup
775          \fi
776        \@tempa
777      }%
778    \fi
779  }
780 \fi
781 \newcommand*{\tocstyle@deprecated@style@KOMAScript}{KOMAlike}%
782 \newtocstyle[KOMAlike]{classic}{%
783   \settocfeature[-1]{pagenumberhook}{\normalfont\normalcolor}%
784   \settocfeature[0]{pagenumberhook}{\normalfont\normalcolor}%
```

```
785  \iftochaschapter\else
786    \settocfeature[1]{pagenumberhook}{\normalfont\normalcolor}%
787  \fi
788  \settocfeature{pagenumberhook}{\normalfont\normalcolor}%
789  \settocfeature{raggedhook}{\raggedright}%
790 }
791 \newtocstyle[classic][leaders]{allwithdot}{}
792 \newtocstyle[allwithdot]{noonewithdot}{%
793   \settocfeature{leaders}{\hfill}%
794 }
795 \newtocstyle[classic][leaders]{nopagecolumn}{%
796   \settocfeature{leaders}{\quad}%
797   \settocfeature{parfillskip}{\z@ plus 1fil}%
798   \settocfeature{pagenumberbox}{\hbox}%
799 }
```

8.2.5. Defining Some TOC Styles

Loading a optional configuration file.

```
800 \InputIfFileExists{tocstyle.cfg}{%
801   \PackageInfo{tocstyle}{using tocstyle.cfg}%
802 }{%
803   \PackageInfo{tocstyle}{no tocstyle.cfg found}%
804 }
```

A. Examples for the Different TOC Styles

Here you will find the table of contents of this document set in the different TOC styles. All are set with option `tocindentauto`.

A.1. Graduated Versions

First of all all graduated versions of the table of contents

A.1.1. `standard` **with Option** `tocgraduated`

A.1.2. `KOMAlike` **with Option** `tocgraduated`

A.1.3. `classic` **with Option** `tocgraduated`

A.1.4. `allwithdot` **with Option** `tocgraduated`

A.1.5. `noonewithdot` **with Option** `tocgraduated`

A.1.6. `nopagecolumn` **with Option** `tocgraduated`

A.2. Flat Versions

Now, all flat versions of the table of contents

A.2.1. `standard` **with Option** `tocflat`

A.2.2. `KOMAlike` with Option `tocflat`

A.2.3. `classic` **with Option** `tocflat`

A.2.4. `allwithdot` with Option `tocflat`

A.2.5. `noonewithdot` **with Option** `tocflat`

A.2.6. `nopagecolumn` **with Option** `tocflat`

A.3. Fullflat Versions

Now, all full-flat versions of the table of contents

A.3.1. `standard` **with Option** `tocfullflat`

A.3.2. `KOMAlike` **with Option** `tocfullflat`

A.3.3. `classic` **with Option** `tocfullflat`

A.3.4. `allwithdot` **with Option** `tocfullflat`

A.3.5. `noonewithdot` **with Option** `tocfullflat`

A.3.6. `nopagecolumn` **with Option** `tocfullflat`

1. How It Works   2

2. Optional Features   3

3. Using TOC Styles   6

4. Setting-up Single Features   6

5. Defining New TOC Styles   9

6. Processing a TOC   9

7. Configuration file   10

8. Implementation   11
8.1. Option   11
8.2. Body   13
8.2.1. Redefining LaTeX Kernel Macros   13
8.2.2. Redefining Class Macros   17
8.2.3. New Macros   20
8.2.4. Defining Some TOC Styles   28
8.2.5. Defining Some TOC Styles   30

A. Examples for the Different TOC Styles   31
A.1. Graduated Versions   31
A.2. Flat Versions   37
A.3. Fullflat Versions   43

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.

Change History