

The KOMA-Script package

tocstyle*

Markus Kohm

2008/10/20

While the main classes of the KOMA-Script bundle were made, there were several ideas for formatting the table of contents and lists of floats, but almost none of them were implemented. One reason was, that the KOMA-Script author didn't like to change the L^AT_EX kernel at a class, because this may result in several 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

1. How It Works	2
2. Optional Features	3
3. Using TOC Styles	5
4. Setting-up Single Features	7
5. Defining New TOC Styles	9
6. Implementation	9
6.1. Option	9
6.2. Body	11
6.2.1. Redefining L ^A T _E X Kernel Macros	11
6.2.2. Redefining Class Macros	16
6.2.3. New Macros	18
6.2.4. Defining Some TOC Styles	26

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

6.2.5. Defining Some TOC Styles 28

A. Examples for the Different TOC Styles 29

- A.1. Graduated Versions 29
 - A.1.1. `standard` with Option `tocgraduated` 29
 - A.1.2. KOMAlike with Option `tocgraduated` 30
 - A.1.3. `classic` with Option `tocgraduated` 31
 - A.1.4. `allwithdot` with Option `tocgraduated` 32
 - A.1.5. `noonewithdot` with Option `tocgraduated` 33
 - A.1.6. `nopagecolumn` with Option `tocgraduated` 34
- A.2. Flat Versions 35
 - A.2.1. `standard` with Option `tocflat` 35
 - A.2.2. KOMAlike with Option `tocflat` 36
 - A.2.3. `classic` with Option `tocflat` 37
 - A.2.4. `allwithdot` with Option `tocflat` 38
 - A.2.5. `noonewithdot` with Option `tocflat` 39
 - A.2.6. `nopagecolumn` with Option `tocflat` 40
- A.3. Fullflat Versions 41
 - A.3.1. `standard` with Option `tocfullflat` 41
 - A.3.2. KOMAlike with Option `tocfullflat` 42
 - A.3.3. `classic` with Option `tocfullflat` 43
 - A.3.4. `allwithdot` with Option `tocfullflat` 44
 - A.3.5. `noonewithdot` with Option `tocfullflat` 45
 - A.3.6. `nopagecolumn` with Option `tocfullflat` 46

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 whether 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 generell 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 L^AT_EX run with all TOC entries. So you need at least three L^AT_EX 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:

1. How It Works	2
2. Optional Features	3
3. Using TOC Styles	5
4. Setting-up Single Features	7

5.	Defining New TOC Styles	9
6.	Implementation	9
6.1.	Option	9
6.2.	Body	11
6.2.1.	Redefining L ^A T _E X Kernel Macros	11
6.2.2.	Redefining Class Macros	16
6.2.3.	New Macros	18
6.2.4.	Defining Some TOC Styles	26
6.2.5.	Defining Some TOC Styles	28
A.	Examples for the Different TOC Styles	29
A.1.	Graduated Versions	29
A.2.	Flat Versions	35
A.3.	Fullflat Versions	41

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:

1.	How It Works	2
2.	Optional Features	3
3.	Using TOC Styles	5
4.	Setting-up Single Features	7
5.	Defining New TOC Styles	9
6.	Implementation	9
6.1.	Option	9
6.2.	Body	11
6.2.1.	Redefining L ^A T _E X Kernel Macros	11
6.2.2.	Redefining Class Macros	16
6.2.3.	New Macros	18
6.2.4.	Defining Some TOC Styles	26
6.2.5.	Defining Some TOC Styles	28
A.	Examples for the Different TOC Styles	29
A.1.	Graduated Versions	29
A.2.	Flat Versions	35
A.3.	Fullflat Versions	41

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:

1. How It Works	2
2. Optional Features	3
3. Using TOC Styles	5
4. Setting-up Single Features	7
5. Defining New TOC Styles	9
6. Implementation	9
6.1. Option	9
6.2. Body	11
6.2.1. Redefining L ^A T _E X Kernel Macros	11
6.2.2. Redefining Class Macros	16
6.2.3. New Macros	18
6.2.4. Defining Some TOC Styles	26
6.2.5. Defining Some TOC Styles	28
A. Examples for the Different TOC Styles	29
A.1. Graduated Versions	29
A.2. Flat Versions	35
A.3. Fullflat Versions	41

Default is option `tocgraduated`.

`tocbreaksstrict` Default option `tocbreaksstrict` sets a lot of penalties before and after
`tocbreakscareless` 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` With default option `toctextentriesleft` unnumbered TOC entries, e.g.
`toctextentriesleft` 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 previous 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.

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 (<code>\bfseries</code>). 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 <code>\large</code> . 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 <code>\usekomafont{disposition}</code> will be used if <code>\usekomafont</code> was defined and sans serif, bold face (<code>\sffamily\bfseries</code>) 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 omitted. This means, that the page numbers are set 1 em after the text.

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` Both commands have one optional argument `⟨TOC⟩`. You may deactivate 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 be reactivated only using `\reactivatetocstyle` without the optional argument or empty optional argument too.

Table 2: Known TOC Shortcuts

<code>toc</code>	Table of contents of almost all known classes.
<code>lof</code>	List of figures of almost all known classes.
<code>lot</code>	List of tables of almost all known classes.
<code>lol</code>	List of listings of package listings. Currently the usability of listings with <code>tocstyle</code> is not recommended. Maybe it works, maybe not. Maybe you should try <code>\deactivatetocstyle[lol]</code> .

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

`\settocfeature`
`\settocstylefeature`

These commands are used to set a single feature either 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 shown at table 3.

The order of used commands for a feature is

1. commands for all depths of all TOCs,
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` and `\settocstylefeature`.

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 <code>\@pnumwidth</code> 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 <code>\parfillskip</code> after setting up all lengths
raggedhook	the only allowed values here are <code>\raggedright</code> or nothing
spaceafternumber	amount of minimum space after the entry number, if the needed width will be calculated automaticly

5. Defining New TOC Styles

Now you know how to select a predefined 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 too much mistakes.

The second optional argument is a comma separated 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.

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 useful to define your own toc styles.

6. 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 }
```

6.1. Option

Options change general behaviour of TOCs.

```
\selecttocstyleoption
9 \newif\iftocstyle@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 }

```

tocbreaksstrict Switch on extended pernalities.

```

tocbreakscareless 60 \DeclareOption{tocbreaksstrict}{\selecttocstyleoption\CurrentOption}
61 \DeclareOption{tocbreakscareless}{\selecttocstyleoption\CurrentOption}

```

tocindentauto

```

tocindentmanual 62 \DeclareOption{tocindentauto}{\selecttocstyleoption\CurrentOption}
63 \DeclareOption{tocindentmanual}{\selecttocstyleoption\CurrentOption}

```

toctextentriesindented

toctextentriesleft

```

64 \DeclareOption{toctextentriesindented}{\selecttocstyleoption\CurrentOption}
65 \DeclareOption{toctextentriesleft}{\selecttocstyleoption\CurrentOption}

tocgraduated
  tocflat 66 \DeclareOption{tocgraduated}{\selecttocstyleoption\CurrentOption}
  tocfullflat 67 \DeclareOption{tocflat}{\selecttocstyleoption\CurrentOption}
68 \DeclareOption{tocfullflat}{\selecttocstyleoption\CurrentOption}

```

Defaults and others:

```

69 \ExecuteOptions{tocbreaksstrict,tocindentauto,tocgraduated,%
70   toctextentriesleft}
71 \ProcessOptions\relax

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

```

6.2. Body

There are two parts at `tocstyle`:

- redefining internal L^AT_EX kernel macros,
- defining new macros and redefining class macros.

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

6.2.1. Redefining L^AT_EX Kernel Macros

Some L^AT_EX 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.

87 \newcommand*\tocstyle@saved@starttoc{}
88 \let\tocstyle@saved@starttoc\@starttoc

```

```

89 \renewcommand*{\@starttoc}[1]{%
90   \tocstyle@pre@starttoc{#1}%
91   \tocstyle@saved@starttoc{#1}%
92   \tocstyle@post@starttoc{#1}%
93 }

\tocstyle@saved@dottedtocline For saving the unchanged definition (at \begindocument):
94 \newcommand*{\tocstyle@saved@dottedtocline}{}

\tocstyle@dottedtocline Implement new definition and redefine:
95 \newcommand*{\tocstyle@dottedtocline}[5]{%
96   \let\numberline\tocstyle@numberline
97   \ifnum #1>\c@tocdepth \else

Penalty feature: no page break between higher and lower depths.
98     \if\tocstyle@penalties
99       \begingroup
100         \@tempcnta 20010
101         \advance \@tempcnta by -#1
102         \ifnum \@tempcnta>\lastpenalty
103           \aftergroup\penalty\aftergroup\@lowpenalty
104         \fi
105       \endgroup
106     \fi

Activation of all features for this TOC and depth:
107     \edef\tocstyledepth{#1}%
108     \tocstyle@activate@features

Similar to kernel command but if feature entryvskip was set use \addvspace:
109     \ifx\tocstyle@feature@entryvskip\relax
110       \vskip \z@ \@plus.2\p@
111     \else
112       \addvspace{\tocstyle@feature@entryvskip}%
113     \fi
114     {%

Preinitialization of lengths and skips and then call a hook
115     \parskip \z@ \parindent \z@ \leftskip \z@ \rightskip \z@
116     \tocstyle@feature@raggedhook
117 %     \end{macrocode}
118 % Set number indent to \cs{@tempdimb} and text indent to \cs{@tempdima}.
119 %     \begin{macrocode}
120     \@tempdima #3\relax
121     \@tempdimb #2\relax
122 <trace>     \typeout{m (\tocstyleTOC, \tocstyledepth): \the\@tempdima}%
123 %     \end{macrocode}
124 % Calc auto lengths
125 %     \begin{macrocode}
126     \ifnum #1>\z@\relax
127       \@tempcnta #1\relax \advance\@tempcnta \m@ne

```

```

128         \ifcsname tocstyle@skipwidth@\tocstyleTOC @\the\@tempcnta\endcsname
129         \ifcsname tocstyle@numwidth@\tocstyleTOC @\the\@tempcnta\endcsname
130         \@tempdimb
131         \csname tocstyle@skipwidth@\tocstyleTOC @\the\@tempcnta\endcsname
132         \advance\@tempdimb
133         \csname tocstyle@numwidth@\tocstyleTOC @\the\@tempcnta\endcsname
134     \fi
135 \fi
136 \fi
137 <trace>     \typeout{C (\tocstyleTOC, \tocstyledepth): \the\@tempdimb}%
138 \ifcsname tocstyle@skipwidth@\tocstyleTOC @#1\endcsname
139     \ifdim \@tempdimb>
140         \csname tocstyle@skipwidth@\tocstyleTOC @#1\endcsname\relax
141         \expandafter\xdef\csname tocstyle@skipwidth@\tocstyleTOC
142             @#1\endcsname{\the\@tempdimb}%
143     \fi
144 \else
145     \expandafter\xdef\csname tocstyle@skipwidth@\tocstyleTOC
146         @#1\endcsname{\the\@tempdimb}%
147 \fi
148 \iftocstyle@autolength
149     \ifcsname tocstyle@maxskipwidth@\tocstyleTOC @#1\endcsname
150         \@tempdimb \csname tocstyle@maxskipwidth@\tocstyleTOC @#1\endcsname
151         \relax
152     \fi
153     \ifcsname tocstyle@maxnumwidth@\tocstyleTOC @#1\endcsname
154         \@tempdima \csname tocstyle@maxnumwidth@\tocstyleTOC @#1\endcsname
155         \relax
156     \fi
157 <trace>     \typeout{a (\tocstyleTOC, \tocstyledepth): \the\@tempdima}%
158 <trace>     \typeout{A (\tocstyleTOC, \tocstyledepth): \the\@tempdimb}%
159 \else
160     \@tempdimb #2\relax
161 <trace>     \typeout{M (\tocstyleTOC, \tocstyledepth): \the\@tempdimb}%
162 \fi
163 \ifcsname tocstyle@unumwidth@\tocstyleTOC @\endcsname
164     \ifdim \@tempdima>
165         \csname tocstyle@unumwidth@\tocstyleTOC @\endcsname\relax
166         \expandafter\xdef\csname tocstyle@unumwidth@\tocstyleTOC
167             @\endcsname{\the\@tempdima}%
168     \fi
169 \else
170     \expandafter\xdef\csname tocstyle@unumwidth@\tocstyleTOC
171         @\endcsname{\the\@tempdima}%
172 \fi
173 \ifcase\tocstyle@indentstyle\relax\else
174     \@tempdimb \z@
175     \ifcsname tocstyle@maxunumwidth@\tocstyleTOC @\endcsname
176         \@tempdima \csname tocstyle@maxunumwidth@\tocstyleTOC @\endcsname

```

```

177         \relax
178     \fi
179 <trace>         \typeout{s (\tocstyleT0C, \tocstyledepth): \the\@tempdima}%
180 <trace>         \typeout{S (\tocstyleT0C, \tocstyledepth): \the\@tempdimb}%
181     \fi
182 %     \end{macrocode}
183 % Advance instead of set, because of the hook above:
184 %     \begin{macrocode}
185     \advance\parindent \@tempdimb\@afterindenttrue
186     \advance\leftskip \parindent
187     \advance\rightskip \@tocrmarg
188     \parfillskip -\rightskip
189     \ifx\tocstyle@feature@parfillskip\relax\else
190         \advance\parfillskip \tocstyle@feature@parfillskip\relax
191     \fi
192     \interlinepenalty\@M
193     \leavevmode
194     \advance\leftskip \@tempdima
195     \null\nobreak

\hskip\leftskip optional moved to \numberline
196     \iftocstyle@indentnotnumbered\else
197         \hskip -\leftskip
198     \fi

Change at start of the entry
199     \tocstyle@feature@entryhook

Similar to kernel command but if feature leaders was set use this in-
stead of the default leaders. And if feature dothook was set (default is
\normalfont) use this at the default leaders.
200     {#4}\nobreak
201     \ifx\tocstyle@feature@leaders\relax
202         \leaders\hbox{$\m@th
203             \mkern \@dotsep mu\hbox{\tocstyle@feature@dothook .}%
204             \mkern \@dotsep mu$}\hfill
205     \else
206         \tocstyle@feature@leaders
207     \fi
208     \nobreak
209     \ifx\tocstyle@feature@pagenumberbox\relax
210         \hb@xt@\@pnumwidth{\hfil\tocstyle@feature@pagenumberhook #5}%
211     \else
212         \tocstyle@feature@pagenumberbox{\tocstyle@feature@pagenumberhook #5}%
213     \fi
214     \par
215 }%

Last change is, another penalty change:
216     \iftocstyle@penalties
217     \bgroup

```

```

218         \@tempcnta 20009
219         \advance\@tempcnta by -#1
220         \edef\reserved@a{\egroup\penalty\the\@tempcnta\relax}%
221     \reserved@a
222 \fi
223 \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.

```

224 \newcommand*{\tocstyle@saved@numberline}{%
225 \newcommand*{\tocstyle@numberline}[1]{%
226 \begingroup
227 \ifx\tocstyle@feature@spaceafternumber\relax
228 \settowidth\@tempdima{\tocstyle@@numberline{#1}\enskip}%
229 \else
230 \settowidth\@tempdima{\tocstyle@@numberline{#1}}%
231 \advance \@tempdima \tocstyle@feature@spaceafternumber\relax
232 \fi
233 \ifcsname tocstyle@numwidth@\tocstyleTOC @\tocstyledepth\endcsname
234 \ifdim \@tempdima >
235 \csname tocstyle@numwidth@\tocstyleTOC @\tocstyledepth\endcsname\relax
236 \expandafter\xdef\csname tocstyle@numwidth@\tocstyleTOC
237 @\tocstyledepth\endcsname{\the\@tempdima}%
238 \fi
239 \else
240 \expandafter\xdef\csname tocstyle@numwidth@\tocstyleTOC
241 @\tocstyledepth\endcsname{\the\@tempdima}%
242 \fi
243 \endgroup
244 \iftocstyle@indentnotnumbered
245 \hskip -\leftskip
246 \fi
247 \ifcase \tocstyle@indentstyle
248 \hb@xt@\@tempdima{\tocstyle@@numberline{#1}\hfil}%
249 \or
250 \hb@xt@\@tempdima{\tocstyle@@numberline{#1}\hfil}%
251 \else
252 \ifx\tocstyle@feature@spaceafternumber\relax
253 \hbox{\tocstyle@@numberline{#1}\enskip}%
254 \else
255 \hbox{\tocstyle@@numberline{#1}\hskip
256 \tocstyle@feature@spaceafternumber\relax}%
257 \fi
258 \fi
259 }

```

`\tocstyle@@numberline` Do the main work!

```

260 \newcommand*{\tocstyle@@numberline}[1]{%
261 #1\csname autodot\endcsname

```

6.2.2. Redefining Class Macros

```

\l@part    Try to redefine the toc commands at startup.
\l@chapter 263 \AtBeginDocument{%
\l@section 264   \ifcsname l@part\endcsname
\l@subsection 265   \ifcsname l@chapter\endcsname
\l@subsubsection 266   \setbox\@tempboxa\vbox{\hsize\maxdimen
\l@paragraph 267   \l@part{\tocstyle@l@define{part}{-1}}{}}}%
268   \else
\l@subparagraph 269   \setbox\@tempboxa\vbox{\hsize\maxdimen
\l@table 270   \l@part{\tocstyle@l@define{part}{0}}{}}}%
\l@figure 271   \fi
272   \fi
273   \ifcsname l@chapter\endcsname
274   \setbox\@tempboxa\vbox{\hsize\maxdimen
275   \l@chapter{\tocstyle@l@define{chapter}{0}}{}}}%
276   \fi
277   \ifcsname l@section\endcsname
278   \setbox\@tempboxa\vbox{\hsize\maxdimen
279   \l@section{\tocstyle@l@define{section}{1}}{}}}%
280   \fi
281   \ifcsname l@subsection\endcsname
282   \setbox\@tempboxa\vbox{\hsize\maxdimen
283   \l@subsection{\tocstyle@l@define{subsection}{2}}{}}}%
284   \fi
285   \ifcsname l@subsubsection\endcsname
286   \setbox\@tempboxa\vbox{\hsize\maxdimen
287   \l@subsubsection{\tocstyle@l@define{subsubsection}{3}}{}}}%
288   \fi
289   \ifcsname l@paragraph\endcsname
290   \setbox\@tempboxa\vbox{\hsize\maxdimen
291   \l@paragraph{\tocstyle@l@define{paragraph}{4}}{}}}%
292   \fi
293   \ifcsname l@subparagraph\endcsname
294   \setbox\@tempboxa\vbox{\hsize\maxdimen
295   \l@subparagraph{\tocstyle@l@define{subparagraph}{5}}{}}}%
296   \fi
297   \ifcsname l@table\endcsname
298   \setbox\@tempboxa\vbox{\hsize\maxdimen
299   \l@table{\tocstyle@l@define{table}{1}}{}}}%
300   \fi
301   \ifcsname l@figure\endcsname
302   \setbox\@tempboxa\vbox{\hsize\maxdimen
303   \l@figure{\tocstyle@l@define{figure}{1}}{}}}%
304   \fi

```

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

then redefine.

```

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

```

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

```

326 \def\@tempa#1{\hb@xt@\@tempdima{#1\autodot\hfil}}%
327 \ifx\numberline\@tempa\else
328   \def\@tempa#1{\hb@xt@\@tempdima{#1\hfil}}%
329   \ifx\numberline\@tempa\else
330     \tocstyle@macrochangewarning\numberline
331   \fi
332 \fi
333 \let\tocstyle@saved@numberline\numberline
334 }

```

`\tocstyle@macrochangewarning`

```

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

```

347 }

```
\tocstyle@l@define
\tocstyle@activate@all@l 348 \newcommand*{\tocstyle@activate@all@l}{%
349 \newcommand*{\tocstyle@l@define}[2]{%
350   \advance\leftskip-\@tempdima
351   \edef\@tempa{%
352     \noexpand\global\noexpand\let
353     \expandafter\noexpand\csname tocstyle@sav@l@#1\endcsname
354     \expandafter\noexpand\csname l@#1\endcsname
355     \noexpand\gdef
356     \expandafter\noexpand\csname tocstyle@l@#1\endcsname{%
357       \noexpand\@dottedtocline{#2}{\the\leftskip}{\the\@tempdima}}%
358     \noexpand\g@addto@macro\noexpand\tocstyle@activate@all@l{%
359       \noexpand\let\expandafter\noexpand\csname l@#1\endcsname
360       \expandafter\noexpand\csname tocstyle@l@#1\endcsname
361     }%
362   }%
363   \PackageInfo{tocstyle}{prepare \expandafter\string
364     \csname l@#1\endcsname\space for redefinition}%
365   \@tempa
366 }
```

6.2.3. New Macros

\showtoc

```
367 \newcommand*{\showtoc}[2][\aliastoc\tocstyleTOC\tocstyleAliasTOC]{%
368   \ifcsname tocstyle@copyn@me@#2\endcsname
369     \@tempcnta \csname tocstyle@copyn@me@#2\endcsname\relax
370     \advance\@tempcnta \@ne
371     \expandafter\xdef\csname tocstyle@copyn@me@#2\endcsname{\the\@tempcnta}%
372   \else
373     \expandafter\xdef\csname tocstyle@copyn@me@#2\endcsname{1}%
374   \fi
375   \ifx\@do@filelist\relax\let\@do@filelist\@empty\fi
376   \edef\@tempa{\noexpand\g@addto@macro\noexpand\@do@filelist{%
377     \noexpand\tocstyle@copy@toc{#2}{\csname
378       tocstyle@copyn@me@#2\endcsname}}}%
379   }\@tempa%
380   \begin@group
381     \edef\tocstyleAliasTOC{#2}%
382     \edef\tocstyleTOC{#2\csname tocstyle@copyn@me@#2\endcsname}%
383     #1
384     \tocstyle@pre@starttoc{#2\csname tocstyle@copyn@me@#2\endcsname}%
385     \makeatletter
386     \@input{\jobname.#2\csname tocstyle@copyn@me@#2\endcsname}%
387     \@nobreakfalse
388     \tocstyle@post@starttoc{#2\csname tocstyle@copyn@me@#2\endcsname}%
389   \end@group
```

```

390 }

\tocstyle@copy@toc
391 \newcommand*{\tocstyle@copy@toc}[2]{%
392   \if@filesw
393     \begingroup
394     \endlinechar=\m@ne
395   % While \LaTeX{} does not close the files, we have to do it know.
396     \immediate\closeout\csname tf@#1\endcsname
397     \immediate\openin\@inputcheck \jobname.#1
398     \immediate\openout\@partaux \jobname.#1#2
399     \loop\unless\ifeof\@inputcheck
400       \immediate\readline\@inputcheck to \@tempa
401       \immediate\write\@partaux{\@tempa}%
402     \repeat
403     \immediate\closeout\@partaux
404     \immediate\closein\@inputcheck
405   \endgroup
406 \fi
407 }

\aliastoc Internal use not the real TOC shortcut but another one.
408 \newcommand*{\aliastoc}[2]{%
409   \expandafter\edef\csname tocstyle@alias@TOC@#1\endcsname{#2}%
410 }

\tocstyle@pre@starttoc Commands before and after the original \@starttoc.
\tocstyle@post@starttoc 411 \newcommand*{\tocstyle@pre@starttoc}[1]{%
412   \begingroup
413     \expandafter\ifx\csname tocstyle@deactivated@\endcsname\relax
414     \expandafter\ifx\csname tocstyle@deactivated@#1\endcsname\relax\relax
415       \tocstyle@activetrue
416     \else
417       \tocstyle@activefalse
418     \fi
419   \else
420     \tocstyle@activefalse
421   \fi
422   \iftocstyle@active
423     \let\@dottedtocline\tocstyle@dottedtocline
424     \parskip \z@
425     \parindent \z@
426     \parfillskip \z@\@plus 1fil
427     \ifcsname tocstyle@alias@TOC@#1\endcsname
428       \edef\tocstyleAliasTOC{\csname tocstyle@alias@TOC@#1\endcsname}%
429     \else
430       \edef\tocstyleAliasTOC{#1}%
431     \fi
432     \edef\tocstyleTOC{#1}%

```

```

433     \tocstyle@activate@all@l
434   \fi
435 }
436 \newcommand*{\tocstyle@post@starttoc}[1]{%
437   \iftocstyle@active
438     \if@filesw
439       \ifcsname tocstyle@unumwidth@#1\endcsname
440         \protected@write\@auxout{}{%
441           \protect\tocstyle@set@width{unum}{#1}{}%
442           \csname tocstyle@unumwidth@#1\endcsname}%
443       }%
444     \fi
445     \expandafter\let\expandafter\@tempa
446     \csname tocstyle@depthlist@#1\endcsname
447     \ifx\@tempa\relax\else
448       \expandafter\@for \expandafter\@tempa\expandafter:\expandafter=\@tempa
449       \do {%
450         \ifcsname tocstyle@numwidth@#1\@tempa\endcsname
451           \protected@write\@auxout{}{%
452             \protect\tocstyle@set@width{num}{#1}{\@tempa}{%
453               \csname tocstyle@numwidth@#1\@tempa\endcsname}%
454           }%
455         \fi
456         \ifcsname tocstyle@skipwidth@#1\@tempa\endcsname
457           \protected@write\@auxout{}{%
458             \protect\tocstyle@set@width{skip}{#1}{\@tempa}{%
459               \csname tocstyle@skipwidth@#1\@tempa\endcsname}%
460           }%
461         \fi
462       }%
463     \fi
464   \fi
465 \endgroup
466 }

```

`\tocstyle@set@width`

```

468 \newcommand*{\tocstyle@set@width}[4]{%
469   \expandafter\gdef\csname tocstyle@max#1width@#2@#3\endcsname{#4}%
470 }

```

`\tocstyleTOC` Shortcut of the current processed TOC. Empty outside of TOCs.

```

\tocstyleAliasTOC 471 \newcommand*{\tocstyleTOC}{}
472 \newcommand*{\tocstyleAliasTOC}{}

```

`\tocstyledepth` Current depth of the current processed TOC entry.

```

473 \newcommand*{\tocstyledepth}{}

```

`\deactivatetocstyle` You may (de)activate all influence of `\tocstyle` either for one or all TOCs.

`\reactivatetocstyle`

```

474 \newif\iftocstyle@active
475 \newcommand*{\deactivatetocstyle}[1] [] {%
476   \expandafter\let\csname tocstyle@deactivated@#1\endcsname\@empty}
477 \newcommand*{\reactivatetocstyle}[1] [] {%
478   \expandafter\let\csname tocstyle@deactivated@#1\endcsname\relax}

\settocfeature The primary command to set the features of a depth of a TOC.
\@settocfeature 479 \newcommand*{\@settocfeature}[1] [] {%
\@@settocfeature 480   \@ifnextchar[ {\@settocfeature[{#1}]}{\@settocfeature[{#1}] []}
481 }
482 \def\@@settocfeature[#1][#2]#3#4{%
483   \trace \typeout{exclude: \tocstyle@feature@excludelist}%
484   \@expandtwoargs\in@{,#3,}{,\tocstyle@feature@excludelist,}%
485   \ifin@ \else
486     \expandafter\ifcsname tocstyle@feature@#3\endcsname
487     \@namedef{\tocstyle@feature@#3@#1@#2}{#4}%
488     \begingroup
489     \expandafter\let\expandafter\@tempa
490     \csname tocstyle@commandlist@#1\endcsname
491     \@expandtwoargs\in@{,\tocstyle@feature@#3@#1@#2,}{,\@tempa,}%
492     \ifin@ \let\@tempa\endgroup \else
493     \edef\@tempa{\endgroup
494       \noexpand\expandafter\noexpand\ifx
495       \noexpand\csname tocstyle@commandlist@#1\endcsname\relax
496       \noexpand\expandafter\noexpand\expandafter\noexpand\expandafter
497       \noexpand\def
498       \noexpand\else
499       \noexpand\expandafter\noexpand\expandafter\noexpand\expandafter
500       \noexpand\l@addto@macro
501       \noexpand\fi
502       \noexpand\csname tocstyle@commandlist@#1\endcsname%
503       {\tocstyle@feature@#3@#1@#2,}}%
504     \fi
505     \@tempa
506   \else
507     \PackageError{tocstyle}{unkown feature ‘#3’}{%
508       You’ve told me to set up toc style feature ‘#3’,\MessageBreak
509       but I don’t know this feature.\MessageBreak
510       See the tocstyle manual for all known feature.\MessageBreak
511     }%
512   \fi
513 \fi
514 }
515 \newcommand*{\settocfeature}{}
516 \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.
517 \providecommand{\l@addto@macro}[2] {%

```

```

518 \edef#1{\unexpanded\expandafter{#1#2}}%
519 }%

```

`\settocstylefeature` Same as above without TOC argument.

```

\@settocstylefeature 520 \newcommand*{\@settocstylefeature}{%
521 \ifnextchar[ {\@settocfeature[]}{\@settocfeature[] []}%
522 }
523 \newcommand*{\settocstylefeature}{}
524 \let\settocstylefeature\@settocstylefeature

```

Different commands will be defined:

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

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

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

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

`\tocstyle@activate@features` Activates the features

```

525 \newcommand*{\tocstyle@activate@features}{%
526 \expandafter\ifx\csname tocstyle@depthlist@\tocstyleTOC\endcsname\relax
527 \expandafter\xdef\csname tocstyle@depthlist@\tocstyleTOC\endcsname{%
528 \tocstyledepth}%
529 \else
530 \expandafter\let\expandafter\@tempa
531 \csname tocstyle@depthlist@\tocstyleTOC\endcsname
532 \@expandtwoargs\in@{,\tocstyledepth,}{,\@tempa,}%
533 \ifin@\else
534 \expandafter\xdef\csname tocstyle@depthlist@\tocstyleTOC\endcsname{%
535 \csname tocstyle@depthlist@\tocstyleTOC\endcsname,\tocstyledepth}%
536 \fi
537 \fi
538 \expandafter\@for \expandafter\@tempa
539 \expandafter:\expandafter=\tocstyle@featurelist \do
540 {%
541 \@ifundefined{tocstyle@feature@\@tempa @\tocstyleAliasTOC @\tocstyledepth}{%
542 \@ifundefined{tocstyle@feature@\@tempa @@\tocstyledepth}{%
543 \@ifundefined{tocstyle@feature@\@tempa @\tocstyleAliasTOC @}{%
544 \@ifundefined{tocstyle@feature@\@tempa @@}{%
545 \expandafter\let\csname tocstyle@feature@\@tempa\endcsname\relax
546 }{%
547 \expandafter\let\csname tocstyle@feature@\@tempa
548 \expandafter\endcsname
549 \csname tocstyle@feature@\@tempa @@\endcsname
550 }%
551 }{%
552 \expandafter\let\csname tocstyle@feature@\@tempa
553 \expandafter\endcsname

```

```

554         \csname tocstyle@feature@\@tempa @\tocstyleAliasTOC @\endcsname
555     }%
556 }{%
557     \expandafter\let\csname tocstyle@feature@\@tempa
558     \expandafter\endcsname
559     \csname tocstyle@feature@\@tempa @@\tocstyledepth\endcsname
560 }%
561 }{%
562     \expandafter\let\csname tocstyle@feature@\@tempa
563     \expandafter\endcsname
564     \csname tocstyle@feature@\@tempa @\tocstyleAliasTOC @\tocstyledepth\endcsname
565 }%
566 }%
567 }

```

`\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 overwrite the parent's definitions. So a new TOC style, that defines all features doesn't need a parent.

```

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

```

```

598 }%
599 }
600 \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 in-
stead and the user will be told about this change.

601 \newcommand*{\usetocstyle}[2][]{%
602   \@ifundefined{tocstyle@deprecated@style@#2}{%
603     \@ifundefined{tocstyle@style@#2}{%
604       \PackageError{tocstyle}{unknown TOC style ‘#2’}{%
605         You’ve told me to use TOC style ‘#2’,\MessageBreak
606         but there’s no TOC style ‘#2’ defined.}%
607     }{%
608       \def\settocfeature{%
609         \@ifnextchar[ {\@settocfeature[{#1}]}{\@settocfeature[{#1}] []}%
610       }%
611       \let\settocstylefeature\settocfeature

Deactivate all known features for this TOC

612   \expandafter\ifx\csname tocstyle@commandlist@#1\endcsname\relax
613   \else
614     \expandafter\expandafter\expandafter\@for
615     \expandafter\expandafter\expandafter\@tempa
616     \expandafter\expandafter\expandafter:%
617     \expandafter\expandafter\expandafter=%
618     \csname tocstyle@commandlist@#1\endcsname
619     \do{%
620       \expandafter\let\csname \@tempa\endcsname\relax
621     }%

So there are no more known features for this TOC.

622   \expandafter\let\csname tocstyle@commandlist@#1\endcsname\relax
623   \fi

Activate all known features for this style and TOC

624   \@usetocstyle{#2}%
625   \let\settocfeature\@settocfeature
626   \let\settocstylefeature\@settocstylefeature
627 }%
628 }{%
629   \expandafter\ifx\csname tocstyle@deprecated@style@#2\endcsname\empty
630   \PackageWarning{tocstyle}{%
631     deprecated TOC style ‘#2’!\MessageBreak
632     You should not longer use this style,\MessageBreak
633     because it will be removed soon.\MessageBreak
634     You should select another TOC style}%
635   \usetocstyle[{#1}]{deprecated@#2}%

```



```

636     \else
637         \PackageWarning{tocstyle}{%
638             deprecated TOC style '#2'\MessageBreak
639             You should use TOC style '\csname
640             tocstyle@deprecated@style@#2\endcsname'\MessageBreak
641             instead of '#2'}%
642     \fi
643 }%
644 }
645 \newcommand*{\@usetocstyle}[1]{%
646     \csname tocstyle@style@#1\endcsname
647 }

\tocstyle@featurelist    Comma separated list of all known features
648 \newcommand*{\tocstyle@featurelist}{%
649     pagenumberhook,entryhook,dothook,entryvskip,leaders,raggedhook,%
650     spaceafternumber,parfillskip,pagumberbox,%
651 }

\tocstyle@feature@pagenumberhook
\tocstyle@feature@pagenumberhook 652 \newcommand*{\tocstyle@feature@pagenumberhook}{}
\tocstyle@feature@entryhook       653 \let\tocstyle@feature@pagenumberhook\relax
\tocstyle@feature@dothook         654 \newcommand*{\tocstyle@feature@pagumberbox}{}
\tocstyle@feature@entryvskip      655 \let\tocstyle@feature@pagumberbox\relax
\tocstyle@feature@leaders         656 \newcommand*{\tocstyle@feature@entryhook}{}
\tocstyle@feature@parfillskip     657 \let\tocstyle@feature@entryhook\relax
\tocstyle@feature@raggedhook      658 \newcommand*{\tocstyle@feature@dothook}{}
\tocstyle@feature@spaceafternumber 659 \let\tocstyle@feature@dothook\relax
\tocstyle@feature@spaceafternumber 660 \newcommand*{\tocstyle@feature@entryvskip}{}
\tocstyle@feature@spaceafternumber 661 \let\tocstyle@feature@entryvskip\relax
\tocstyle@feature@spaceafternumber 662 \newcommand*{\tocstyle@feature@leaders}{}
\tocstyle@feature@spaceafternumber 663 \let\tocstyle@feature@leaders\relax
\tocstyle@feature@spaceafternumber 664 \newcommand*{\tocstyle@feature@parfillskip}{}
\tocstyle@feature@spaceafternumber 665 \let\tocstyle@feature@parfillskip\relax
\tocstyle@feature@spaceafternumber 666 \newcommand*{\tocstyle@feature@raggedhook}{}
\tocstyle@feature@spaceafternumber 667 \let\tocstyle@feature@raggedhook\relax
\tocstyle@feature@spaceafternumber 668 \newcommand*{\tocstyle@feature@spaceafternumber}{}
\tocstyle@feature@spaceafternumber 669 \let\tocstyle@feature@spaceafternumber\relax

\iftochasdepth    Uses \tocstyle@depthlist@<TOC> to test, if the TOC has the depth
                  already.
670 \newcommand*{\iftochasdepth}[2]{%
671     \begingroup
672         \expandafter\let\expandafter\@tempa\csname tocstyle@depthlist@#1\endcsname
673         \ifx\@tempa\relax
674             \aftergroup\@secondoftwo
675         \else
676             \@expandtwoargs\in@{,#2,}{,\@tempa}%
677             \expandafter\aftergroup\ifin@

```

```

678         \@firstoftwo
679     \else
680         \@secondoftwo
681     \fi
682 \fi
683 \endgroup
684 }

```

6.2.4. Defining Some TOC Styles

```

685 \newtocstyle{standard}{%
686     \settocfeature{dothook}{\normalfont}%
687     \settocfeature[-1]{entryhook}{\bfseries}%
688     \settocfeature[-1]{entryvskip}{2.25em\@plus\p@}%
689     \settocfeature[-1]{leaders}{\hfill}%
690     \settocfeature[0]{entryvskip}{1em\@plus\p@}%
691     \settocfeature[0]{leaders}{\hfill}%
692     \settocfeature[0]{entryhook}{%
693         \begingroup
694             \edef\@tempa{toc}%
695             \ifx\tocstyleAliasTOC\@tempa\aftergroup\bfseries\fi
696         \endgroup
697     }%
698     \begingroup\expandafter\expandafter\expandafter\endgroup
699     \expandafter\ifx\csname l@chapter\endcsname\relax
700         \settocfeature[1]{entryvskip}{1em\@plus\p@}%
701         \settocfeature[1]{leaders}{\hfill}%
702         \settocfeature[1]{entryhook}{%
703             \begingroup
704                 \edef\@tempa{toc}%
705                 \ifx\tocstyleAliasTOC\@tempa\aftergroup\bfseries\fi
706             \endgroup
707         }%
708     \fi
709 }
710 \begingroup\expandafter\expandafter\expandafter\endgroup
711 \expandafter\ifx\csname sectfont\endcsname\relax
712     \newtocstyle{KOMAlike}{%
713         \settocfeature{dothook}{\normalfont}%
714         \settocfeature[-1]{entryhook}{\sffamily\bfseries}%
715         \settocfeature[-1]{entryvskip}{2.25em\@plus\p@}%
716         \settocfeature[-1]{leaders}{\hfill}%
717         \settocfeature[0]{entryvskip}{1em\@plus\p@}%
718         \settocfeature[0]{leaders}{\hfill}%
719         \settocfeature[0]{entryhook}{%
720             \begingroup
721                 \edef\@tempa{toc}%
722                 \ifx\tocstyleAliasTOC\@tempa\aftergroup\sffamily\bfseries\fi
723             \endgroup

```

```

724 }%
725 \begingroup\expandafter\expandafter\expandafter\endgroup
726 \expandafter\ifx\csname l@chapter\endcsname\relax
727   \settocfeature[1]{entryvskip}{1em\@plus\p@}%
728   \settocfeature[1]{leaders}{\hfill}%
729   \settocfeature[1]{entryhook}{%
730     \begingroup
731       \edef\@tempa{toc}%
732       \ifx\tocstyleAliasTOC\@tempa\aftergroup\sffamily\bfseries\fi
733     \endgroup
734   }%
735 \fi
736 }
737 \else
738 \newtocstyle{KOMAlike}{%
739   \settocfeature{dothook}{\normalfont}%
740   \settocfeature[-1]{entryhook}{\sectfont}%
741   \settocfeature[-1]{entryvskip}{2.25em\@plus\p@}%
742   \settocfeature[-1]{leaders}{\hfill}%
743   \settocfeature[0]{entryvskip}{1em\@plus\p@}%
744   \settocfeature[0]{leaders}{\hfill}%
745   \settocfeature[0]{entryhook}{%
746     \begingroup
747       \edef\@tempa{toc}%
748       \ifx\tocstyleAliasTOC\@tempa\aftergroup\sectfont\fi
749     \endgroup
750   }%
751 \begingroup\expandafter\expandafter\expandafter\endgroup
752 \expandafter\ifx\csname l@chapter\endcsname\relax
753   \settocfeature[1]{entryvskip}{1em\@plus\p@}%
754   \settocfeature[1]{leaders}{\hfill}%
755   \settocfeature[1]{entryhook}{%
756     \begingroup
757       \edef\@tempa{toc}%
758       \ifx\tocstyleAliasTOC\@tempa\aftergroup\sectfont\fi
759     \endgroup
760   }%
761 \fi
762 }
763 \fi
764 \newcommand*{\tocstyle@deprecated@style@KOMAScript}{KOMAlike}%
765 \newtocstyle[KOMAlike]{classic}{%
766   \settocfeature{pagenumberhook}{\normalfont\normalcolor}%
767   \settocfeature{raggedhook}{\raggedright}%
768 }
769 \newtocstyle[classic][leaders]{allwithdot}{%
770 \newtocstyle[allwithdot]{noonewithdot}{%
771   \settocfeature{leaders}{\hfill}%
772 }

```

```

773 \newtocstyle[classic][leaders]{nopagecolumn}{%
774   \settocfeature{leaders}{\quad}%
775   \settocfeature{parfillskip}{\z@ plus 1fil}%
776   \settocfeature{pagenumberbox}{\hbox}%
777 }

```

6.2.5. Defining Some TOC Styles

Loading a optional configuration file.

```

778 \InputIfFileExists{tocstyle.cfg}{%
779   \PackageInfo{tocstyle}{using tocstyle.cfg}%
780 }{%
781   \PackageInfo{tocstyle}{no tocstyle.cfg found}%
782 }

```

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`

1. How It Works	2
2. Optional Features	3
3. Using TOC Styles	5
4. Setting-up Single Features	7
5. Defining New TOC Styles	9
6. Implementation	9
6.1. Option	9
6.2. Body	11
6.2.1. Redefining L ^A T _E X Kernel Macros	11
6.2.2. Redefining Class Macros	16
6.2.3. New Macros	18
6.2.4. Defining Some TOC Styles	26
6.2.5. Defining Some TOC Styles	28
A. Examples for the Different TOC Styles	29
A.1. Graduated Versions	29
A.2. Flat Versions	35
A.3. Fullflat Versions	41

A.1.2. KOMAlike with Option `tocgraduated`

1. How It Works	2
2. Optional Features	3
3. Using TOC Styles	5
4. Setting-up Single Features	7
5. Defining New TOC Styles	9
6. Implementation	9
6.1. Option	9
6.2. Body	11
6.2.1. Redefining L ^A T _E X Kernel Macros	11
6.2.2. Redefining Class Macros	16
6.2.3. New Macros	18
6.2.4. Defining Some TOC Styles	26
6.2.5. Defining Some TOC Styles	28
A. Examples for the Different TOC Styles	29
A.1. Graduated Versions	29
A.2. Flat Versions	35
A.3. Fullflat Versions	41

A.1.3. classic with Option tocgraduated

1. How It Works	2
2. Optional Features	3
3. Using TOC Styles	5
4. Setting-up Single Features	7
5. Defining New TOC Styles	9
6. Implementation	9
6.1. Option	9
6.2. Body	11
6.2.1. Redefining L ^A T _E X Kernel Macros	11
6.2.2. Redefining Class Macros	16
6.2.3. New Macros	18
6.2.4. Defining Some TOC Styles	26
6.2.5. Defining Some TOC Styles	28
A. Examples for the Different TOC Styles	29
A.1. Graduated Versions	29
A.2. Flat Versions	35
A.3. Fullflat Versions	41

A.1.4. allwithdot with Option tocgraduated

1. How It Works	2
2. Optional Features	3
3. Using TOC Styles	5
4. Setting-up Single Features	7
5. Defining New TOC Styles	9
6. Implementation	9
6.1. Option	9
6.2. Body	11
6.2.1. Redefining L ^A T _E X Kernel Macros	11
6.2.2. Redefining Class Macros	16
6.2.3. New Macros	18
6.2.4. Defining Some TOC Styles	26
6.2.5. Defining Some TOC Styles	28
A. Examples for the Different TOC Styles	29
A.1. Graduated Versions	29
A.2. Flat Versions	35
A.3. Fullflat Versions	41

A.1.5. noonewithdot with Option tocgraduated	
1. How It Works	2
2. Optional Features	3
3. Using TOC Styles	5
4. Setting-up Single Features	7
5. Defining New TOC Styles	9
6. Implementation	9
6.1. Option	9
6.2. Body	11
6.2.1. Redefining L ^A T _E X Kernel Macros	11
6.2.2. Redefining Class Macros	16
6.2.3. New Macros	18
6.2.4. Defining Some TOC Styles	26
6.2.5. Defining Some TOC Styles	28
A. Examples for the Different TOC Styles	29
A.1. Graduated Versions	29
A.2. Flat Versions	35
A.3. Fullflat Versions	41

A.1.6. nopagecolumn with Option tocgraduated

1. How It Works	2
2. Optional Features	3
3. Using TOC Styles	5
4. Setting-up Single Features	7
5. Defining New TOC Styles	9
6. Implementation	9
6.1. Option	9
6.2. Body	11
6.2.1. Redefining L ^A T _E X Kernel Macros	11
6.2.2. Redefining Class Macros	16
6.2.3. New Macros	18
6.2.4. Defining Some TOC Styles	26
6.2.5. Defining Some TOC Styles	28
A. Examples for the Different TOC Styles	29
A.1. Graduated Versions	29
A.2. Flat Versions	35
A.3. Fullflat Versions	41

A.2. Flat Versions

Now, all flat versions of the table of contents

A.2.1. standard with Option `tocflat`

1.	How It Works	2
2.	Optional Features	3
3.	Using TOC Styles	5
4.	Setting-up Single Features	7
5.	Defining New TOC Styles	9
6.	Implementation	9
6.1.	Option	9
6.2.	Body	11
6.2.1.	Redefining \LaTeX Kernel Macros	11
6.2.2.	Redefining Class Macros	16
6.2.3.	New Macros	18
6.2.4.	Defining Some TOC Styles	26
6.2.5.	Defining Some TOC Styles	28
A.	Examples for the Different TOC Styles	29
A.1.	Graduated Versions	29
A.2.	Flat Versions	35
A.3.	Fullflat Versions	41

A.2.2. KOMALike with Option `tocflat`

1.	How It Works	2
2.	Optional Features	3
3.	Using TOC Styles	5
4.	Setting-up Single Features	7
5.	Defining New TOC Styles	9
6.	Implementation	9
6.1.	Option	9
6.2.	Body	11
6.2.1.	Redefining \LaTeX Kernel Macros	11
6.2.2.	Redefining Class Macros	16
6.2.3.	New Macros	18
6.2.4.	Defining Some TOC Styles	26
6.2.5.	Defining Some TOC Styles	28
A.	Examples for the Different TOC Styles	29
A.1.	Graduated Versions	29
A.2.	Flat Versions	35
A.3.	Fullflat Versions	41

A.2.3. classic with Option tocflat

1.	How It Works	2
2.	Optional Features	3
3.	Using TOC Styles	5
4.	Setting-up Single Features	7
5.	Defining New TOC Styles	9
6.	Implementation	9
6.1.	Option	9
6.2.	Body	11
6.2.1.	Redefining L ^A T _E X Kernel Macros	11
6.2.2.	Redefining Class Macros	16
6.2.3.	New Macros	18
6.2.4.	Defining Some TOC Styles	26
6.2.5.	Defining Some TOC Styles	28
A.	Examples for the Different TOC Styles	29
A.1.	Graduated Versions	29
A.2.	Flat Versions	35
A.3.	Fullflat Versions	41

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

1.	How It Works	2
2.	Optional Features	3
3.	Using TOC Styles	5
4.	Setting-up Single Features	7
5.	Defining New TOC Styles	9
6.	Implementation	9
6.1.	Option	9
6.2.	Body	11
6.2.1.	Redefining \LaTeX Kernel Macros	11
6.2.2.	Redefining Class Macros	16
6.2.3.	New Macros	18
6.2.4.	Defining Some TOC Styles	26
6.2.5.	Defining Some TOC Styles	28
A.	Examples for the Different TOC Styles	29
A.1.	Graduated Versions	29
A.2.	Flat Versions	35
A.3.	Fullflat Versions	41

A.2.5. noonewithdot with Option tocflat

1.	How It Works	2
2.	Optional Features	3
3.	Using TOC Styles	5
4.	Setting-up Single Features	7
5.	Defining New TOC Styles	9
6.	Implementation	9
6.1.	Option	9
6.2.	Body	11
6.2.1.	Redefining \LaTeX Kernel Macros	11
6.2.2.	Redefining Class Macros	16
6.2.3.	New Macros	18
6.2.4.	Defining Some TOC Styles	26
6.2.5.	Defining Some TOC Styles	28
A.	Examples for the Different TOC Styles	29
A.1.	Graduated Versions	29
A.2.	Flat Versions	35
A.3.	Fullflat Versions	41

A.2.6. nopagecolumn with Option tocflat

- 1. How It Works** 2
- 2. Optional Features** 3
- 3. Using TOC Styles** 5
- 4. Setting-up Single Features** 7
- 5. Defining New TOC Styles** 9
- 6. Implementation** 9
 - 6.1. Option 9
 - 6.2. Body 11
 - 6.2.1. Redefining L^AT_EX Kernel Macros 11
 - 6.2.2. Redefining Class Macros 16
 - 6.2.3. New Macros 18
 - 6.2.4. Defining Some TOC Styles 26
 - 6.2.5. Defining Some TOC Styles 28
- A. Examples for the Different TOC Styles** 29
 - A.1. Graduated Versions 29
 - A.2. Flat Versions 35
 - A.3. Fullflat Versions 41

A.3. Fullflat Versions

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

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

1. How It Works	2
2. Optional Features	3
3. Using TOC Styles	5
4. Setting-up Single Features	7
5. Defining New TOC Styles	9
6. Implementation	9
6.1. Option	9
6.2. Body	11
6.2.1. Redefining L ^A T _E X Kernel Macros	11
6.2.2. Redefining Class Macros	16
6.2.3. New Macros	18
6.2.4. Defining Some TOC Styles	26
6.2.5. Defining Some TOC Styles	28
A. Examples for the Different TOC Styles	29
A.1. Graduated Versions	29
A.2. Flat Versions	35
A.3. Fullflat Versions	41

A.3.2. KOMAlike with Option `tocfullflat`

1. How It Works	2
2. Optional Features	3
3. Using TOC Styles	5
4. Setting-up Single Features	7
5. Defining New TOC Styles	9
6. Implementation	9
6.1. Option	9
6.2. Body	11
6.2.1. Redefining L ^A T _E X Kernel Macros	11
6.2.2. Redefining Class Macros	16
6.2.3. New Macros	18
6.2.4. Defining Some TOC Styles	26
6.2.5. Defining Some TOC Styles	28
A. Examples for the Different TOC Styles	29
A.1. Graduated Versions	29
A.2. Flat Versions	35
A.3. Fullflat Versions	41

A.3.3. classic with Option tocfullflat

1. How It Works	2
2. Optional Features	3
3. Using TOC Styles	5
4. Setting-up Single Features	7
5. Defining New TOC Styles	9
6. Implementation	9
6.1. Option	9
6.2. Body	11
6.2.1. Redefining L ^A T _E X Kernel Macros	11
6.2.2. Redefining Class Macros	16
6.2.3. New Macros	18
6.2.4. Defining Some TOC Styles	26
6.2.5. Defining Some TOC Styles	28
A. Examples for the Different TOC Styles	29
A.1. Graduated Versions	29
A.2. Flat Versions	35
A.3. Fullflat Versions	41

A.3.4. allwithdot with Option tocfullflat

1. How It Works	2
2. Optional Features	3
3. Using TOC Styles	5
4. Setting-up Single Features	7
5. Defining New TOC Styles	9
6. Implementation	9
6.1. Option	9
6.2. Body	11
6.2.1. Redefining L ^A T _E X Kernel Macros	11
6.2.2. Redefining Class Macros	16
6.2.3. New Macros	18
6.2.4. Defining Some TOC Styles	26
6.2.5. Defining Some TOC Styles	28
A. Examples for the Different TOC Styles	29
A.1. Graduated Versions	29
A.2. Flat Versions	35
A.3. Fullflat Versions	41

A.3.5. noonewithdot with Option tocfullflat

1. How It Works	2
2. Optional Features	3
3. Using TOC Styles	5
4. Setting-up Single Features	7
5. Defining New TOC Styles	9
6. Implementation	9
6.1. Option	9
6.2. Body	11
6.2.1. Redefining L ^A T _E X Kernel Macros	11
6.2.2. Redefining Class Macros	16
6.2.3. New Macros	18
6.2.4. Defining Some TOC Styles	26
6.2.5. Defining Some TOC Styles	28
A. Examples for the Different TOC Styles	29
A.1. Graduated Versions	29
A.2. Flat Versions	35
A.3. Fullflat Versions	41

A.3.6. nopagecolumn with Option tocfullflat

1. How It Works	2
2. Optional Features	3
3. Using TOC Styles	5
4. Setting-up Single Features	7
5. Defining New TOC Styles	9
6. Implementation	9
6.1. Option	9
6.2. Body	11
6.2.1. Redefining L ^A T _E X Kernel Macros	11
6.2.2. Redefining Class Macros	16
6.2.3. New Macros	18
6.2.4. Defining Some TOC Styles	26
6.2.5. Defining Some TOC Styles	28
A. Examples for the Different TOC Styles	29
A.1. Graduated Versions	29
A.2. Flat Versions	35
A.3. Fullflat Versions	41

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.

Symbols		R	
<code>\@@@settocfeature</code>	479	<code>\reactivatetocstyle</code>	6, 474
<code>\@dottedtocline</code>	305	S	
<code>\@settocfeature</code>	479	<code>\selecttocstyleoption</code>	9
<code>\@settocstylefeature</code>	520	<code>\settocfeature</code>	7, 479
<code>\@starttoc</code>	87	<code>\settocstylefeature</code>	7, 520
<code>\@usetocstyle</code>	601	<code>\showtoc</code>	367
A		T	
<code>\aliastoc</code>	408	<code>\tocbreakscareless</code>	5
D		<code>tocbreakscareless (Option)</code>	60
<code>\deactivatetocstyle</code>	6, 474	<code>\tocbreaksstrict</code>	5
I		<code>tocbreaksstrict (Option)</code>	60
<code>\iftochasdepth</code>	670	<code>\tocflat</code>	3
L		<code>tocflat (Option)</code>	66
<code>\l@addto@macro</code>	517	<code>\tocfullflat</code>	3
<code>\l@chapter</code>	263	<code>tocfullflat (Option)</code>	66
<code>\l@figure</code>	263	<code>\tocgraduated</code>	3
<code>\l@paragraph</code>	263	<code>tocgraduated (Option)</code>	66
<code>\l@part</code>	263	<code>\tocindentauto</code>	3
<code>\l@section</code>	263	<code>tocindentauto (Option)</code>	62
<code>\l@subparagraph</code>	263	<code>\tocindentmanual</code>	3
<code>\l@subsection</code>	263	<code>tocindentmanual (Option)</code>	62
<code>\l@subsubsection</code>	263	<code>\tocstyle@@numberline</code>	260
<code>\l@table</code>	263	<code>\tocstyle@activate@all@l</code>	348
N		<code>\tocstyle@activate@features</code>	525
<code>\newtocstyle</code>	9, 568	<code>\tocstyle@copy@toc</code>	391
<code>\numberline</code>	326	<code>\tocstyle@dottedtocline</code>	95
O		<code>\tocstyle@feature@<feature>@@</code>	525
Optionen:		<code>\tocstyle@feature@@<feature>@<TOC>@</code>	525
<code>tocbreakscareless</code>	60	<code>\tocstyle@feature@@<feature>@<TOC>@<depth></code>	525
<code>tocbreaksstrict</code>	60	<code>\tocstyle@feature@@<feature>@@<depth></code>	525
<code>tocflat</code>	66	<code>\tocstyle@feature@dothook</code>	652
<code>tocfullflat</code>	66	<code>\tocstyle@feature@entryhook</code>	652
<code>tocgraduated</code>	66	<code>\tocstyle@feature@entryvskip</code>	652
<code>tocindentauto</code>	62	<code>\tocstyle@feature@leaders</code>	652
<code>tocindentmanual</code>	62	<code>\tocstyle@feature@pagenumberhook</code>	652
<code>toctextentriesindented</code>	64	<code>\tocstyle@feature@parfillskip</code>	652
<code>toctextentriesleft</code>	64		

<code>\tocstyle@feature@raggedhook</code>	652	<code>\tocstyle@set@width</code>	468
<code>\tocstyle@feature@spaceafternumber</code>	652	<code>\tocstyleAliasTOC</code>	471
<code>\tocstyle@featurelist</code>	648	<code>\tocstyledepth</code>	473
<code>\tocstyle@l@define</code>	348	<code>\tocstyleTOC</code>	471
<code>\tocstyle@macrochangewarning</code>	335	<code>\toctextentriesindented</code>	5
<code>\tocstyle@numberline</code>	224	<code>toctextentriesindented</code> (Op- tion)	64
<code>\tocstyle@post@starttoc</code>	411	<code>\toctextentriesleft</code>	5
<code>\tocstyle@pre@starttoc</code>	411	<code>toctextentriesleft</code> (Option)	64
<code>\tocstyle@saved@@starttoc</code>	87		
<code>\tocstyle@saved@dottedtocline</code>	94		
<code>\tocstyle@saved@numberline</code>	224		
		U	
		<code>\usetocstyle</code>	5, 601

Change History

v0.1

General: start of new package . . . 1

v0.2a

`\@usetocstyle`: extended for
deprecated TOC styles . . . 24