The KOMA-Script package `tocstyle`*

Markus Kohm

2017/02/23

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.

Since KOMA-Script 3.20 most of the features of tocstyle are also provided by other KOMA-Script packages like tocbasic.

Contents

---

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

1. Package Status

Package tocstyle has alpha status for a very long time (since 2007), despite the fact that it was very stable all over the years. The main reason for this is that it was an experiment and I always intended to make a better successor. Another reason was, that there are known issues, which are hardly to fix. So tocstyle never became an official part of KOMA-Script. Nevertheless it has been published with KOMA-Script, because I thought it would be an useful addition. I never thought that the interim status of the package would be for more than eight years.

    With KOMA-Script 3.20 several features of tocstyle can be found also in tocbasic. I recommend to use that official and essential part of KOMA-Script instead of tocstyle whenever possible. Nevertheless, I plan to ensure

the compatibility of tocstyle with new releases of KOMA-Script for at least already existing, unchanged documents as long as it is possible. But expect that tocstyle will move from KOMA-Script bundle to another package, e. g., KOMA-Script-obsolete or a stand-alone package.

2. 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.

3. 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   With option `tocindentauto` all widths at the TOCs are calculated by
tocindentmanual

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   The option `tocgraduated` selects the graduated version of all TOCs.
tocflat   You know the graduated version from the standard classes. Entries of
tocfullflat   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`   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.

4. 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.

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 3). 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.

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.

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]`.

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

5. 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 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,

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`.

8

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

---

6. 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.

7. 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 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 3 was made using

```
\showtoc[{%
```

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

If you want to show a copy of 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 \ \ \tocstyleAliasTOC   These are read-only macros. While processing a TOC using `\@starttoc` 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.

8. 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.

9. 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 }
```

9.1. 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   Do we have `\chapter` and `\l@chapter`?

nochapter

`\iftochaschapter`

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

tocbreaksstrict   Switch on extended pernalties.

tocbreakscareless

```
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
```

9.2. 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.

9.2.1. 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
\tocstyle@saved@@starttoc
The original definition will be extended by defaults for \parskip, \parindent 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
```

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
```

14

```
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 }
```

9.2.2. Redefining Class Macros

\l@part   Try to redefine the toc commands at startup.
\l@chapter
\l@section
\l@subsection
\l@subsubsection
\l@paragraph
\l@subparagraph
\l@table
\l@figure

```
261 \AtBeginDocument{%
262   \@ifpackageloaded{tocbasic}{%
263     \@ifpackagelater{tocbasic}{2016/03/01}{%
264       \PackageWarningNoLine{tocstyle}{%
265         Usage of `tocstyle' with new `tocbasic' detected.\MessageBreak
266         This is not an error! You can do this.\MessageBreak
267         Nevertheless, you should note, combining `tocstyle'\MessageBreak
268         with this version of `tocbasic' will break several\MessageBreak
269         features of `tocbasic'. You should use `tocstyle'\MessageBreak
270         features instead of `tocbasic' attributes for all\MessageBreak
271         entry changes you want.\MessageBreak
272         You may also get additional warnings because of\MessageBreak
273         redefined `\string\numberline'. Ignore them}%
274     }{}%
275   }{}%
276   \@ifpackageloaded{etoc}{%
277     \PackageWarningNoLine{tocstyle}{%
278       Usage of `tocstyle' with `etoc' detected.\MessageBreak
279       I suggest to use either `tocstyle' or `etoc'\MessageBreak
280       but not both of them together}%
```

18

```
281  }{}%
282  \@ifpackageloaded{titletoc}{%
283    \PackageWarningNoLine{tocstyle}{%
284      Usage of 'tocstyle' with 'titletoc' detected.\MessageBreak
285      I suggest to use either 'tocstyle' or 'titletoc'\MessageBreak
286      but not both of them together. I expect even\MessageBreak
287      error messages because of this combination}%
288  }{}%
289  \def\tocstyle@test@level{-1}%
290  \def\tocstyle@test@levelname{part}%
291  \@whilesw\ifcsname l@\tocstyle@test@levelname\endcsname\fi{%
292    \setbox\@tempboxa\vbox{\hsize\maxdimen
293      \let\normalcolor\relax
294      \let\color\@gobble
295      \edef\reserved@a{%
296        \expandafter\noexpand\csname l@\tocstyle@test@levelname\endcsname{%
297          \noexpand\tocstyle@l@define{\tocstyle@test@levelname}%
298                                     {\tocstyle@test@level}%
299        }{}%
300      }%
301      \reserved@a
302    }%
303    \edef\tocstyle@test@levelname{sub\tocstyle@test@levelname}%
304    \edef\tocstyle@test@level{\the\numexpr\tocstyle@test@level+1}%
305  }%
306  \ifnum\tocstyle@test@level<\z@
307    \def\tocstyle@test@level{0}%
308  \fi
309  \def\tocstyle@test@levelname{chapter}%
310  \@whilesw\ifcsname l@\tocstyle@test@levelname\endcsname\fi{%
311    \setbox\@tempboxa\vbox{\hsize\maxdimen
312      \let\normalcolor\relax
313      \let\color\@gobble
314      \edef\reserved@a{%
315        \expandafter\noexpand\csname l@\tocstyle@test@levelname\endcsname{%
316          \noexpand\tocstyle@l@define{\tocstyle@test@levelname}%
317                                     {\tocstyle@test@level}%
318        }{}%
319      }%
320      \reserved@a
321    }%
322    \edef\tocstyle@test@levelname{sub\tocstyle@test@levelname}%
323    \edef\tocstyle@test@level{\the\numexpr\tocstyle@test@level+1}%
324  }%
325  \ifnum\tocstyle@test@level<\@ne
326    \def\tocstyle@test@level{1}%
327  \fi
328  \def\tocstyle@test@levelname{section}%
329  \@whilesw\ifcsname l@\tocstyle@test@levelname\endcsname\fi{%
```

```
330    \setbox\@tempboxa\vbox{\hsize\maxdimen
331      \let\normalcolor\relax
332      \let\color\@gobble
333      \edef\reserved@a{%
334        \expandafter\noexpand\csname l@\tocstyle@test@levelname\endcsname{%
335          \noexpand\tocstyle@l@define{\tocstyle@test@levelname}%
336                                     {\tocstyle@test@level}%
337        }{}%
338      }%
339      \reserved@a
340    }%
341    \edef\tocstyle@test@levelname{sub\tocstyle@test@levelname}%
342    \edef\tocstyle@test@level{\the\numexpr\tocstyle@test@level+1}%
343  }%
344  \def\tocstyle@test@levelname{paragraph}%
345  \@whilesw\ifcsname l@\tocstyle@test@levelname\endcsname\fi{%
346    \setbox\@tempboxa\vbox{\hsize\maxdimen
347      \let\normalcolor\relax
348      \let\color\@gobble
349      \edef\reserved@a{%
350        \expandafter\noexpand\csname l@\tocstyle@test@levelname\endcsname{%
351          \noexpand\tocstyle@l@define{\tocstyle@test@levelname}%
352                                     {\tocstyle@test@level}%
353        }{}%
354      }%
355      \reserved@a
356    }%
357    \edef\tocstyle@test@levelname{sub\tocstyle@test@levelname}%
358    \edef\tocstyle@test@level{\the\numexpr\tocstyle@test@level+1}%
359  }%
360  \def\tocstyle@test@level{1}%
361  \def\tocstyle@test@levelname{table}%
362  \@whilesw\ifcsname l@\tocstyle@test@levelname\endcsname\fi{%
363    \setbox\@tempboxa\vbox{\hsize\maxdimen
364      \let\normalcolor\relax
365      \let\color\@gobble
366      \edef\reserved@a{%
367        \expandafter\noexpand\csname l@\tocstyle@test@levelname\endcsname{%
368          \noexpand\tocstyle@l@define{\tocstyle@test@levelname}%
369                                     {\tocstyle@test@level}%
370        }{}%
371      }%
372      \reserved@a
373    }%
374    \edef\tocstyle@test@levelname{sub\tocstyle@test@levelname}%
375    \edef\tocstyle@test@level{\the\numexpr\tocstyle@test@level+1}%
376  }%
377  \def\tocstyle@test@level{1}%
378  \def\tocstyle@test@levelname{figure}%
```

```
379    \@whilesw\ifcsname l@\tocstyle@test@levelname\endcsname\fi{%
380      \setbox\@tempboxa\vbox{\hsize\maxdimen
381        \let\normalcolor\relax
382        \let\color\@gobble
383        \edef\reserved@a{%
384          \expandafter\noexpand\csname l@\tocstyle@test@levelname\endcsname{%
385            \noexpand\tocstyle@l@define{\tocstyle@test@levelname}%
386                                       {\tocstyle@test@level}%
387          }{}%
388        }%
389        \reserved@a
390      }%
391      \edef\tocstyle@test@levelname{sub\tocstyle@test@levelname}%
392      \edef\tocstyle@test@level{\the\numexpr\tocstyle@test@level+1}%
393    }%
```

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

```
394    \def\@tempa#1#2#3#4#5{%
395      \ifnum #1>\c@tocdepth \else
396        \vskip \z@ \@plus.2\p@
397        {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
398         \parindent #2\relax\@afterindenttrue
399         \interlinepenalty\@M
400         \leavevmode
401         \@tempdima #3\relax
402         \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
403         {#4}\nobreak
404         \leaders\hbox{$\m@th
405            \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
406            mu$}\hfill
407         \nobreak
408         \hb@xt@\@pnumwidth{\hfil \normalfont \normalcolor #5}%
409         \par}%
410      \fi}%
411    \ifx\@dottedtocline\@tempa\else
412      \tocstyle@macrochangewarning\@dottedtocline
413    \fi
414    \let\tocstyle@saved@dottedtocline\@dottedtocline
```

\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:

```
415    \def\@tempa#1{\hb@xt@\@tempdima{#1\autodot\hfil}}%
416    \ifx\numberline\@tempa\else
417      \def\@tempa#1{\hb@xt@\@tempdima{#1\hfil}}%
418      \ifx\numberline@tempa\else
419        \tocstyle@macrochangewarning\numberline
420      \fi
```

```
421    \fi
422    \let\tocstyle@saved@numberline\numberline
423 }
```

\tocstyle@macrochangewarning

```
424 \newcommand*{\tocstyle@macrochangewarning}[1]{%
425    \PackageWarningNoLine{tocstyle}{%
426      unexpected \string#1\space definition!\MessageBreak
427      You are either using an unknown LaTeX kernel\MessageBreak
428      version, an unknown class or package, that redefines\MessageBreak
429      \string#1, or a \string#1\space
430      redefinition\MessageBreak
431      at the document preamble.\MessageBreak
432      Because of this you may get unexpected results!\MessageBreak
433      Maybe it would be better not to use package tocstyle}%
434    \PackageInfo{tocstyle}{Unexpected definition is:\MessageBreak
435      \meaning#1}%
436 }
```

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

```
437 \newcommand*{\tocstyle@activate@all@l}{}
438 \newcommand*{\tocstyle@l@define}[2]{%
439    \advance\leftskip-\@tempdima
440    \edef\@tempa{%
441      \noexpand\global\noexpand\let
442      \expandafter\noexpand\csname tocstyle@saved@l@#1\endcsname
443      \expandafter\noexpand\csname l@#1\endcsname
444      \noexpand\gdef
445      \expandafter\noexpand\csname tocstyle@l@#1\endcsname{%
446        \noexpand\@dottedtocline{#2}{\the\leftskip}{\the\@tempdima}}%
447      \noexpand\g@addto@macro\noexpand\tocstyle@activate@all@l{%
448        \noexpand\let\expandafter\noexpand\csname l@#1\endcsname
449        \expandafter\noexpand\csname tocstyle@l@#1\endcsname
450      }%
451    }%
452    \PackageInfo{tocstyle}{prepare \expandafter\string
453      \csname l@#1\endcsname\space for redefinition}%
454    \@tempa
455 }
```

9.2.3. New Macros

\showtoc

```
456 \newcommand*{\showtoc}[2][\aliastoc\tocstyleTOC\tocstyleAliasTOC]{%
457    \ifcsname tocstyle@copyname@#2\endcsname
458      \@tempcnta \csname tocstyle@copyname@#2\endcsname\relax
459      \advance\@tempcnta \@ne
460      \expandafter\xdef\csname tocstyle@copyname@#2\endcsname{\the\@tempcnta}%
461    \else
```

```
462    \expandafter\xdef\csname tocstyle@copyname@#2\endcsname{1}%
463  \fi
464  \ifx\@dofilelist\relax\let\@dofilelist\@empty\fi
465  \edef\@tempa{\noexpand\g@addto@macro\noexpand\@dofilelist{%
466      \noexpand\tocstyle@copy@toc{#2}{\csname
467        tocstyle@copyname@#2\endcsname}}%
468  }\@tempa%
469  \begingroup
470    \edef\tocstyleAliasTOC{#2}%
471    \edef\tocstyleTOC{#2\csname tocstyle@copyname@#2\endcsname}%
472    #1
473    \tocstyle@pre@starttoc{#2\csname tocstyle@copyname@#2\endcsname}%
474    \makeatletter
475    \@input{\jobname.#2\csname tocstyle@copyname@#2\endcsname}%
476    \@nobreakfalse
477    \tocstyle@post@starttoc{#2\csname tocstyle@copyname@#2\endcsname}%
478  \endgroup
479 }
```

\tocstyle@copy@toc

```
480 \newcommand*{\tocstyle@copy@toc}[2]{%
481  \if@filesw
482    \begingroup
483      \endlinechar=\m@ne
```

While LaTeX does not close the files, we have to do it know.

```
484      \immediate\closeout\csname tf@#1\endcsname
485      \immediate\openin\@inputcheck \jobname.#1
486      \immediate\openout\@partaux \jobname.#1#2
487      \loop\unless\ifeof\@inputcheck
488        \immediate\readline\@inputcheck to \@tempa
489        \immediate\write\@partaux{\@tempa}%
490      \repeat
491      \immediate\closeout\@partaux
492      \immediate\closein\@inputcheck
493    \endgroup
494  \fi
495 }
```

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

```
496 \newcommand*{\aliastoc}[2]{%
497  \expandafter\edef\csname tocstyle@alias@TOC@#1\endcsname{#2}%
498 }
```

\tocstyle@pre@starttoc   Commands before and after the original \@starttoc.
\tocstyle@post@starttoc
```
499 \newcommand*{\tocstyle@pre@starttoc}[1]{%
500  \begingroup
501    \expandafter\ifx\csname tocstyle@deactivated@\endcsname\relax
502      \expandafter\ifx\csname tocstyle@deactivated@#1\endcsname\relax\relax
```

```
503        \tocstyle@activetrue
504      \else
505        \tocstyle@activefalse
506      \fi
507    \else
508      \tocstyle@activefalse
509    \fi
510    \iftocstyle@active
511      \let\@dottedtocline\tocstyle@dottedtocline
512      \parskip \z@
513      \parindent \z@
514      \parfillskip \z@\@plus 1fil
515      \ifcsname tocstyle@alias@TOC@#1\endcsname
516        \edef\tocstyleAliasTOC{\csname tocstyle@alias@TOC@#1\endcsname}%
517      \else
518        \edef\tocstyleAliasTOC{#1}%
519      \fi
520      \edef\tocstyleTOC{#1}%
521      \tocstyle@activate@all@l
522    \fi
523 }
524 \newcommand*{\tocstyle@post@starttoc}[1]{%
525    \iftocstyle@active
526      \if@filesw
527        \ifcsname tocstyle@unumwidth@#1@\endcsname
528          \protected@write\@auxout{}{%
529            \protect\tocstyle@set@width{unum}{#1}{}{%
530              \csname tocstyle@unumwidth@#1@\endcsname}%
531          }%
532        \fi
533        \expandafter\let\expandafter\@tempa
534          \csname tocstyle@depthlist@#1\endcsname
535        \ifx\@tempa\relax\else
536          \expandafter\@for \expandafter\@tempa\expandafter:\expandafter=\@tempa
537          \do {%
538            \ifcsname tocstyle@numwidth@#1@\@tempa\endcsname
539              \protected@write\@auxout{}{%
540                \protect\tocstyle@set@width{num}{#1}{\@tempa}{%
541                  \csname tocstyle@numwidth@#1@\@tempa\endcsname}%
542              }%
543            \fi
544            \ifcsname tocstyle@skipwidth@#1@\@tempa\endcsname
545              \protected@write\@auxout{}{%
546                \protect\tocstyle@set@width{skip}{#1}{\@tempa}{%
547                  \csname tocstyle@skipwidth@#1@\@tempa\endcsname}%
548              }%
549            \fi
550          }%
551        \fi
```

```
552        \fi
553      \fi
554    \endgroup
555 }
```

\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!

```
556 \newcommand*{\tocstyle@set@width}[4]{%
557    \iftocstyle@indentnotnumbered
558      \ifdim #4<\z@
559        \expandafter\gdef\csname tocstyle@max#1width@#2@#3\endcsname{%
560          \dimexpr #4/2\relax}%
561      \else
562        \expandafter\gdef\csname tocstyle@max#1width@#2@#3\endcsname{#4}%
563      \fi
564    \else
565      \ifdim #4<\z@
566        \expandafter\gdef\csname tocstyle@max#1width@#2@#3\endcsname{\z@}%
567      \else
568        \expandafter\gdef\csname tocstyle@max#1width@#2@#3\endcsname{#4}%
569      \fi
570    \fi
571 }
572 \AtBeginDocument{%
573    \if@filesw
574      \immediate\write\@auxout{%
575        \string\providecommand*\string\tocstyle@set@width[4]{}%
576      }%
577    \fi
578 }
```

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

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

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

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

```
587 \newcommand*{\@settocfeature}[1][]{%
588   \kernel@ifnextchar[ {\@@settocfeature[{#1}]}{\@@settocfeature[{#1}][]}
589 }
590 \def\@@settocfeature[#1][#2]#3#4{%
591 ⟨trace⟩   \typeout{exclude: \tocstyle@feature@excludelist}%
592   \@expandtwoargs\in@{,#3,}{,\tocstyle@feature@excludelist,}%
593   \ifin@\else
594     \expandafter\ifcsname tocstyle@feature@#3\endcsname
595       \@namedef{tocstyle@feature@#3@#1@#2}{#4}%
596       \begingroup
597         \expandafter\let\expandafter\@tempa
598         \csname tocstyle@commandlist@#1\endcsname
599         \@expandtwoargs\in@{,tocstyle@feature@#3@#1@#2,}{,\@tempa,}%
600         \ifin@\let\@tempa\endgroup\else
601           \edef\@tempa{\endgroup
602             \noexpand\expandafter\noexpand\ifx
603             \noexpand\csname tocstyle@commandlist@#1\noexpand\endcsname\relax
604               \noexpand\expandafter\noexpand\expandafter\noexpand\expandafter
605               \noexpand\def
606             \noexpand\else
607               \noexpand\expandafter\noexpand\expandafter\noexpand\expandafter
608               \noexpand\l@addto@macro
609             \noexpand\fi
610             \noexpand\csname tocstyle@commandlist@#1\noexpand\endcsname%
611             {tocstyle@feature@#3@#1@#2,}}%
612         \fi
613       \@tempa
614     \else
615       \PackageError{tocstyle}{unkown feature '#3'}{%
616         You've told me to set up toc style feature '#3',\MessageBreak
617         but I don't know this feature.\MessageBreak
618         See the tocstyle manual for all known feature.\MessageBreak
619       }%
620     \fi
621   \fi
622 }
623 \newcommand*{\settocfeature}{}
624 \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.

```
625 \providecommand{\l@addto@macro}[2]{%
626   \edef#1{\unexpanded\expandafter{#1#2}}%
627 }%
```

\settocstylefeature   Same as above without TOC argument.
\@settocstylefeature
```
628 \newcommand*{\@settocstylefeature}{%
629   \kernel@ifnextchar[ {\@settocfeature[]}{\@settocfeature[][]}}%
630 }
```

26

```
631 \newcommand*{\settocstylefeature}{}
632 \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

```
633 \newcommand*{\tocstyle@activate@features}{%
634   \expandafter\ifx\csname tocstyle@depthlist@\tocstyleTOC\endcsname\relax
635     \expandafter\xdef\csname tocstyle@depthlist@\tocstyleTOC\endcsname{%
636       \tocstyledepth}%
637   \else
638     \expandafter\let\expandafter\@tempa
639     \csname tocstyle@depthlist@\tocstyleTOC\endcsname
640     \@expandtwoargs\in@{,\tocstyledepth,}{,\@tempa,}%
641     \ifin@\else
642       \expandafter\xdef\csname tocstyle@depthlist@\tocstyleTOC\endcsname{%
643         \csname tocstyle@depthlist@\tocstyleTOC\endcsname,\tocstyledepth}%
644     \fi
645   \fi
646   \expandafter\@for \expandafter\@tempa
647   \expandafter:\expandafter=\tocstyle@featurelist \do
648   {%
649     \@ifundefined{tocstyle@feature@\@tempa @\tocstyleAliasTOC @\tocstyledepth}{%
650       \@ifundefined{tocstyle@feature@\@tempa @@\tocstyledepth}{%
651         \@ifundefined{tocstyle@feature@\@tempa @\tocstyleAliasTOC @}{%
652           \@ifundefined{tocstyle@feature@\@tempa @@}{%
653             \expandafter\let\csname tocstyle@feature@\@tempa\endcsname\relax
654           }{%
655             \expandafter\let\csname tocstyle@feature@\@tempa
656             \expandafter\endcsname
657             \csname tocstyle@feature@\@tempa @@\endcsname
658           }%
659         }{%
660           \expandafter\let\csname tocstyle@feature@\@tempa
661           \expandafter\endcsname
662           \csname tocstyle@feature@\@tempa @\tocstyleAliasTOC @\endcsname
663         }%
664       }{%
665         \expandafter\let\csname tocstyle@feature@\@tempa
666         \expandafter\endcsname
667         \csname tocstyle@feature@\@tempa @@\tocstyledepth\endcsname
668       }%
```

```
669     }{%
670       \expandafter\let\csname tocstyle@feature@\@tempa
671       \expandafter\endcsname
672       \csname tocstyle@feature@\@tempa @\tocstyleAliasTOC @\tocstyledepth\endcsna
673     }%
674   }%
675 }
```

\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.

```
676 \newcommand*{\newtocstyle}{%
677   \kernel@ifnextchar [{\@newtocstyle}{\@newtocstyle[]}}
678 \newcommand*{\@newtocstyle}{}
679 \def\@newtocstyle[#1]{%
680   \kernel@ifnextchar [{\@@newtocstyle[{#1}]}{\@newtocstyle[{#1}][]}}
681 \newcommand*{\@@newtocstyle}{}
682 \def\@@newtocstyle[#1][#2]#3#4{%
683   \@ifundefined{tocstyle@style@#3}{%
684     \@ifundefined{tocstyle@style@#1}{%
685       \ifx \relax#1\relax\else
686         \PackageError{tocstyle}{unknown parent TOC style '#1'}{%
687           You've told me to inheritate parent TOC style '#1',\MessageBreak
688           but there's no TOC style '#1' defined.}%
689       \fi
690       \expandafter\def\csname tocstyle@style@#3\endcsname{#4}%
691     }{%
692       \expandafter\def\csname tocstyle@style@#3\endcsname{%
693         \edef\reserved@a{%
694           \noexpand\l@addto@macro\noexpand\tocstyle@feature@excludelist{#2}%
695           \noexpand\@usetocstyle{#1}%
696           \noexpand\def\noexpand\tocstyle@feature@excludelist{%
697             \tocstyle@feature@excludelist}%
698         }\reserved@a
699         #4%
700       }%
701     }%
702   }{%
703     \PackageError{tocstyle}{TOC style '#3' already defined}{%
704       You've tried to define a new TOC style '#3',\MessageBreak
705       but there's already a TOC style named '#3'.}%
706   }%
707 }
708 \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.

```
709 \newcommand*{\usetocstyle}[2][]{%
710   \@ifundefined{tocstyle@deprecated@style@#2}{%
711     \@ifundefined{tocstyle@style@#2}{%
712       \PackageError{tocstyle}{unknown TOC style '#2'}{%
713         You've told me to use TOC style '#2',\MessageBreak
714         but there's no TOC style '#2' defined.}%
715     }{%
716       \def\settocfeature{%
717         \kernel@ifnextchar[ %]
718         {\@@settocfeature[{#1}]}{\@@settocfeature[{#1}][]}%
719       }%
720       \let\settocstylefeature\settocfeature
```

Deactivate all known features for this TOC

```
721       \expandafter\ifx\csname tocstyle@commandlist@#1\endcsname\relax
722       \else
723         \expandafter\expandafter\expandafter\@for
724         \expandafter\expandafter\expandafter\@tempa
725         \expandafter\expandafter\expandafter:%
726         \expandafter\expandafter\expandafter=%
727         \csname tocstyle@commandlist@#1\endcsname
728         \do{%
729           \expandafter\let\csname \@tempa\endcsname\relax
730         }%
```

So there are no more known features for this TOC.

```
731         \expandafter\let\csname tocstyle@commandlist@#1\endcsname\relax
732       \fi
```

Activate all known features for this style and TOC

```
733       \@usetocstyle{#2}%
734       \let\settocfeature\@settocfeature
735       \let\settocstylefeature\@settocstylefeature
736     }%
737   }{%
738     \expandafter\ifx\csname tocstyle@deprecated@style@#2\endcsname\@empty
739       \PackageWarning{tocstyle}{%
740         deprecated TOC style '#2'!\MessageBreak
741         You should not longer use this style,\MessageBreak
742         because it will be removed soon.\MessageBreak
743         You should select another TOC style}%
744       \usetocstyle[{#1}]{deprecated@#2}%
745     \else
746       \PackageWarning{tocstyle}{%
747         deprecated TOC style '#2'!\MessageBreak
748         You should use TOC style '\csname
749         tocstyle@deprecated@style@#2\endcsname'\MessageBreak
750         instead of '#2'}%
```

```
751     \fi
752   }%
753 }
754 \newcommand*{\@usetocstyle}[1]{%
755   \csname tocstyle@style@#1\endcsname
756 }
```

\tocstyle@featurelist   Comma seperated list of all known features

```
757 \newcommand*{\tocstyle@featurelist}{%
758   pagenumberhook,entryhook,dothook,entryvskip,leaders,raggedhook,%
759   spaceafternumber,parfillskip,pagenumberbox,%
760 }
```

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

```
761 \newcommand*{\tocstyle@feature@pagenumberhook}{}
762 \let\tocstyle@feature@pagenumberhook\relax
763 \newcommand*{\tocstyle@feature@pagenumberbox}{}
764 \let\tocstyle@feature@pagenumberbox\relax
765 \newcommand*{\tocstye@feature@entryhook}{}
766 \let\tocstyle@feature@entryhook\relax
767 \newcommand*{\tocstye@feature@dothook}{}
768 \let\tocstyle@feature@dothook\relax
769 \newcommand*{\tocstye@feature@entryvskip}{}
770 \let\tocstyle@feature@entryvskip\relax
771 \newcommand*{\tocstye@feature@leaders}{}
772 \let\tocstyle@feature@leaders\relax
773 \newcommand*{\tocstye@feature@parfillskip}{}
774 \let\tocstyle@feature@parfillskip\relax
775 \newcommand*{\tocstye@feature@raggedhook}{}
776 \let\tocstyle@feature@raggedhook\relax
777 \newcommand*{\tocstye@feature@spaceafternumber}{}
778 \let\tocstyle@feature@spaceafternumber\relax
```

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

```
779 \newcommand*{\iftochasdepth}[2]{%
780   \begingroup
781     \expandafter\let\expandafter\@tempa\csname tocstyle@depthlist@#1\endcsname
782     \ifx\@tempa\relax
783       \aftergroup\@secondoftwo
784     \else
785       \@expandtwoargs\in@{,#2,}{,\@tempa}%
786       \expandafter\aftergroup\ifin@
787         \@firstoftwo
788       \else
789         \@secondoftwo
790       \fi
791     \fi
792   \endgroup
```

30

```
793 }
```

9.2.4. Defining Some TOC Styles

\ext@toc From version 0.2i the indirect extension for the auxiliary file of the table
of contents is used like the KOMA-Script classes do.

```
794 \providecommand*{\ext@toc}{toc}

795 \newtocstyle{standard}{%
796   \settocfeature{dothook}{\normalfont}%
797   \settocfeature[-1]{entryhook}{\bfseries}%
798   \settocfeature[-1]{entryvskip}{2.25em\@plus\p@}%
799   \settocfeature[-1]{leaders}{\hfill}%
800   \settocfeature[0]{entryvskip}{1em\@plus\p@}%
801   \settocfeature[0]{leaders}{\hfill}%
802   \settocfeature[0]{entryhook}{\bfseries}
803   \iftochaschapter\else
804     \settocfeature[1]{entryvskip}{1em\@plus\p@}%
805     \settocfeature[1]{leaders}{\hfill}%
806     \settocfeature[1]{entryhook}{%
807       \ifx\tocstyleAliasTOC\ext@toc\bfseries\fi
808     }%
809   \fi
810 }
811 \begingroup\expandafter\expandafter\expandafter\endgroup
812 \expandafter\ifx\csname KOMAClassName\endcsname\relax
813   \newtocstyle{KOMAlike}{%
814     \settocfeature{dothook}{\normalfont}%
815     \settocfeature[-1]{entryhook}{\sffamily\bfseries}%
816     \settocfeature[-1]{entryvskip}{2.25em\@plus\p@}%
817     \settocfeature[-1]{leaders}{\hfill}%
818     \settocfeature[-1]{pagenumberhook}{\sffamily\bfseries}%
819     \settocfeature[0]{entryvskip}{1em\@plus\p@}%
820     \settocfeature[0]{leaders}{\hfill}%
821     \settocfeature[0]{entryhook}{\sffamily\bfseries}
822     \settocfeature[0]{pagenumberhook}{\sffamily\bfseries}%
823     \iftochaschapter\else
824       \settocfeature[1]{entryvskip}{1em\@plus\p@}%
825       \settocfeature[1]{leaders}{\hfill}%
826       \settocfeature[1]{entryhook}{%
827         \ifx\tocstyleAliasTOC\ext@toc\sffamily\bfseries\fi
828       }%
829       \settocfeature[1]{pagenumberhook}{%
830         \ifx\tocstyleAliasTOC\ext@toc\sffamily\bfseries\fi
831       }%
832     \fi
833   }
834 \else
835   \newtocstyle{KOMAlike}{%
```

31

```
836    \settocfeature{dothook}{\normalfont}%
837    \settocfeature[-1]{entryhook}{\usekomafont{partentry}}%
838    \settocfeature[-1]{entryvskip}{2.25em\@plus\p@}%
839    \settocfeature[-1]{leaders}{\hfill}%
840    \settocfeature[-1]{pagenumberhook}{\usekomafont{partentrypagenumber}}%
841    \settocfeature[0]{entryvskip}{1em\@plus\p@}%
842    \settocfeature[0]{leaders}{\hfill}%
843    \settocfeature[0]{entryhook}{\usekomafont{chapterentry}}%
844    \settocfeature[0]{pagenumberhook}{\usekomafont{chapterentrypagenumber}}%
845    \iftochaschapter\else
846      \settocfeature[1]{entryvskip}{1em\@plus\p@}%
847      \settocfeature[1]{leaders}{\hfill}%
848      \settocfeature[1]{entryhook}{%
849        \begingroup
850          \ifx\tocstyleAliasTOC\ext@toc
851            \def\@tempa{\endgroup\usekomafont{sectionentry}}%
852          \else
853            \let\@tempa\endgroup
854          \fi
855        \@tempa
856      }%
857      \settocfeature[1]{pagenumberhook}{%
858        \begingroup
859          \ifx\tocstyleAliasTOC\ext@toc
860            \def\@tempa{\endgroup\usekomafont{sectionentrypagenumber}}%
861          \else
862            \let\@tempa\endgroup
863          \fi
864        \@tempa
865      }%
866    \fi
867  }
868 \fi
869 \newcommand*{\tocstyle@deprecated@style@KOMAScript}{KOMAlike}%
870 \newtocstyle[KOMAlike]{classic}{%
871   \settocfeature[-1]{pagenumberhook}{\normalfont\normalcolor}%
872   \settocfeature[0]{pagenumberhook}{\normalfont\normalcolor}%
873   \iftochaschapter\else
874     \settocfeature[1]{pagenumberhook}{\normalfont\normalcolor}%
875   \fi
876   \settocfeature{pagenumberhook}{\normalfont\normalcolor}%
877   \settocfeature{raggedhook}{\raggedright}%
878 }
879 \newtocstyle[classic][leaders]{allwithdot}{}
880 \newtocstyle[allwithdot]{noonewithdot}{%
881   \settocfeature{leaders}{\hfill}%
882 }
883 \newtocstyle[classic][leaders]{nopagecolumn}{%
884   \settocfeature{leaders}{\quad}%
```

```
885  \settocfeature{parfillskip}{\z@ plus 1fil}%
886  \settocfeature{pagenumberbox}{\hbox}%
887 }
```

9.2.5. Defining Some TOC Styles

Loading a optional configuration file.

```
888 \InputIfFileExists{tocstyle.cfg}{%
889   \PackageInfo{tocstyle}{using tocstyle.cfg}%
890 }{%
891   \PackageInfo{tocstyle}{no tocstyle.cfg found}%
892 }
```

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`

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