

The obsolete package `tocstyle`*

Markus Kohm

2019/10/09

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.

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

Contents

1. Package Status	2
2. How It Works	3
3. Optional Features	3
4. Using TOC Styles	6
5. Setting-up Single Features	8
6. Defining New TOC Styles	10
7. Processing a TOC	10
8. Configuration file	11
9. Implementation	12
9.1. Option	12

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

9.2.	Body	14
9.2.1.	Redefining L ^A T _E X Kernel Macros	14
9.2.2.	Redefining Class Macros	18
9.2.3.	New Macros	23
9.2.4.	Defining Some TOC Styles	31
9.2.5.	Defining Some TOC Styles	33
A.	Examples for the Different TOC Styles	34
A.1.	Graduated Versions	34
A.1.1.	<code>standard</code> with Option <code>tocgraduated</code>	34
A.1.2.	<code>KOMAlike</code> with Option <code>tocgraduated</code>	35
A.1.3.	<code>classic</code> with Option <code>tocgraduated</code>	36
A.1.4.	<code>allwithdot</code> with Option <code>tocgraduated</code>	37
A.1.5.	<code>noonewithdot</code> with Option <code>tocgraduated</code>	38
A.1.6.	<code>nopagecolumn</code> with Option <code>tocgraduated</code>	39
A.2.	Flat Versions	40
A.2.1.	<code>standard</code> with Option <code>tocflat</code>	40
A.2.2.	<code>KOMAlike</code> with Option <code>tocflat</code>	41
A.2.3.	<code>classic</code> with Option <code>tocflat</code>	42
A.2.4.	<code>allwithdot</code> with Option <code>tocflat</code>	43
A.2.5.	<code>noonewithdot</code> with Option <code>tocflat</code>	44
A.2.6.	<code>nopagecolumn</code> with Option <code>tocflat</code>	45
A.3.	Fullflat Versions	46
A.3.1.	<code>standard</code> with Option <code>tocfullflat</code>	46
A.3.2.	<code>KOMAlike</code> with Option <code>tocfullflat</code>	47
A.3.3.	<code>classic</code> with Option <code>tocfullflat</code>	48
A.3.4.	<code>allwithdot</code> with Option <code>tocfullflat</code>	49
A.3.5.	<code>noonewithdot</code> with Option <code>tocfullflat</code>	50
A.3.6.	<code>nopagecolumn</code> with Option <code>tocfullflat</code>	51

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. Expect that `tocstyle` will

move from KOMA-Script bundle to another package, e. g., KOMA-Script-obsolete or a stand-alone package soon.

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 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.

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 general behaviour of all TOCs.

`tocindentauto`
`tocindentmanual`

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

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

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

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

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

`tocgraduated`
`tocflat`
`tocfullflat`

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

1. Package Status	2
2. How It Works	3
3. Optional Features	3
4. Using TOC Styles	6
5. Setting-up Single Features	8
6. Defining New TOC Styles	10
7. Processing a TOC	10
8. Configuration file	11
9. Implementation	12
9.1. Option	12
9.2. Body	14
9.2.1. Redefining L ^A T _E X Kernel Macros	14
9.2.2. Redefining Class Macros	18
9.2.3. New Macros	23
9.2.4. Defining Some TOC Styles	31
9.2.5. Defining Some TOC Styles	33
A. Examples for the Different TOC Styles	34
A.1. Graduated Versions	34
A.2. Flat Versions	40
A.3. Fullflat Versions	46

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.	Package Status	2
2.	How It Works	3
3.	Optional Features	3
4.	Using TOC Styles	6
5.	Setting-up Single Features	8
6.	Defining New TOC Styles	10
7.	Processing a TOC	10
8.	Configuration file	11
9.	Implementation	12
9.1.	Option	12
9.2.	Body	14
9.2.1.	Redefining L ^A T _E X Kernel Macros	14
9.2.2.	Redefining Class Macros	18
9.2.3.	New Macros	23
9.2.4.	Defining Some TOC Styles	31
9.2.5.	Defining Some TOC Styles	33
A.	Examples for the Different TOC Styles	34
A.1.	Graduated Versions	34
A.2.	Flat Versions	40
A.3.	Fullflat Versions	46

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.	Package Status	2
2.	How It Works	3
3.	Optional Features	3
4.	Using TOC Styles	6
5.	Setting-up Single Features	8
6.	Defining New TOC Styles	10

7. Processing a TOC	10
8. Configuration file	11
9. Implementation	12
9.1. Option	12
9.2. Body	14
9.2.1. Redefining L ^A T _E X Kernel Macros	14
9.2.2. Redefining Class Macros	18
9.2.3. New Macros	23
9.2.4. Defining Some TOC Styles	31
9.2.5. Defining Some TOC Styles	33
A. Examples for the Different TOC Styles	34
A.1. Graduated Versions	34
A.2. Flat Versions	40
A.3. Fullflat Versions	46

Default is option `tocgraduated`.

<code>tocbreaksstrict</code>	Default option <code>tocbreaksstrict</code> sets a lot of penalties before and after TOC entries to avoid page break between a TOC entry and it's parent. But sometimes you may like to allow more page breaks. You may use option <code>tocbreakscareless</code> for this.
<code>tocbreakscareless</code>	
<code>toctextentriesindented</code>	With default option <code>toctextentriesleft</code> unnumbered TOC entries, e.g. from KOMA-Script command <code>\addchap</code> , are indented only as wide as the number of numbered TOC entries of the same level are. But with option <code>toctextentriesindented</code> these are indented as if they have an empty number.
<code>toctextentriesleft</code>	

4. Using TOC Styles

Package `tocstyle` has 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 general setting.

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

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

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 (<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.

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> .

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 $\langle TOC \rangle$. 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.

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 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` $\{\langle feature \rangle\}$ $\{\langle command \rangle\}$ or `\settocstylefeature` $\{\langle feature \rangle\}$ $\{\langle commands \rangle\}$), or of all depth of a single TOC (`\settocfeature` $[\langle TOC \rangle]$ $\{\langle feature \rangle\}$ $\{\langle commands \rangle\}$), or of a single depth of all TOCs (`\settocstylefeature` $[\langle depth \rangle]$ $\{\langle feature \rangle\}$ $\{\langle commands \rangle\}$), or of a single depth of a single TOC (`\settocfeature` $[\langle TOC \rangle]$ $[\langle depth \rangle]$ $\{\langle feature \rangle\}$ $\{\langle commands \rangle\}$).

Parameter $\langle commands \rangle$ 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 $\langle feature \rangle$ is the feature, that may be configured with parameter $\langle commands \rangle$. All known features are shown at table 3.

The order of used commands for a feature is

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

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`.

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 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`

`\showtoc` [*<preprosession>*]{*<TOC>*} is an addition of `tocstyle`. Using it will produce 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 internally. 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 A DEPRECATED ALPHA VERSION!\MessageBreak
3   USAGE OF THIS VERSION IS ON YOUR OWN RISK!\MessageBreak
4   EVERYTHING MAY HAPPEN!\MessageBreak
5   THE PACKAGE IS FROZEN WITH ALL IT'S BUGS!\MessageBreak
6   IT WILL BE REMOVED FROM KOMA-SCRIPT SOON!\MessageBreak
7   THERE IS NO SUPPORT, IF YOU USE THIS PACKAGE!\MessageBreak
8   Maybe it would be better, not to load this package%
9 }
```

9.1. Option

Options change general behaviour of TOCs.

`\selecttocstyleoption`

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

```

40             \ifx\@tempa\@tempb\aftergroup\tocstyle@indentnotnumberedtrue
41         \else
42             \edef\@tempb{toctextentriesleft}%
43             \ifx\@tempa\@tempb
44                 \aftergroup\tocstyle@indentnotnumberedfalse
45             \else
46                 \PackageError{tocstyle}{unknown option ‘#1’}{%
47                     You’ve told me to select toc style option
48                     ‘#1’,\MessageBreak
49                     but tocstyle doesn’t know an option named ‘#1’}%
50             \fi
51         \fi
52     \fi
53 \fi
54 \fi
55 \fi
56 \fi
57 \fi
58 \fi
59 \endgroup
60 }

chapter Do we have \chapter and \l@chapter?
nochapter 61 \newif\iftochaschapter\tochaschapterfalse
\iftochaschapter 62 \ifcsname l@chapter\endcsname
63 \ifcsname chapter\endcsname
64 \tochaschaptertrue
65 \fi
66 \fi

tocbreaksstrict Switch on extended pernalties.
tocbreakscareless 67 \DeclareOption{tocbreaksstrict}{\selecttocstyleoption\CurrentOption}
68 \DeclareOption{tocbreakscareless}{\selecttocstyleoption\CurrentOption}

tocindentauto
tocindentmanual 69 \DeclareOption{tocindentauto}{\selecttocstyleoption\CurrentOption}
70 \DeclareOption{tocindentmanual}{\selecttocstyleoption\CurrentOption}

toctextentriesindented
toctextentriesleft 71 \DeclareOption{toctextentriesindented}{\selecttocstyleoption\CurrentOption}
72 \DeclareOption{toctextentriesleft}{\selecttocstyleoption\CurrentOption}

tocgraduated
tocflat 73 \DeclareOption{tocgraduated}{\selecttocstyleoption\CurrentOption}
tocfullflat 74 \DeclareOption{tocflat}{\selecttocstyleoption\CurrentOption}
75 \DeclareOption{tocfullflat}{\selecttocstyleoption\CurrentOption}

```

Defaults and others:

```

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

79 \ifcsname if@tocleft\endcsname
80  \expandafter\let\csname if@tempswa\expandafter\endcsname
81  \csname if@tocleft\endcsname
82 \else
83  \@tempswafalse
84 \fi
85 \if@tempswa
86  \PackageWarningNoLine{tocstyle}{%
87   You should not use class option ‘toc=flat’!\MessageBreak
88   This may result in errors or unexpected results.\MessageBreak
89   I’ll try to deactivate ‘toc=flat’, now.\MessageBreak
90   You may use package options ‘tocflat’ and\MessageBreak
91   ‘tocindentauto’ instead of ‘toc=flat’}%
92  \csname @tocleftfalse\endcsname
93 \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 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.
94 \newcommand*\tocstyle@saved@starttoc{}
95 \let\tocstyle@saved@starttoc\@starttoc
96 \renewcommand*\@starttoc[1]{%
97  \tocstyle@pre@starttoc{#1}%
98  \tocstyle@saved@starttoc{#1}%
99  \tocstyle@post@starttoc{#1}%
100 }

\tocstyle@saved@dottedtocline For saving the unchanged definition (at \begin{document}):
101 \newcommand*\@starttoc[1]{%

```

```

\tocstyle@dottedtocline Implement new definition and redefine:
102 \newcommand*{\tocstyle@dottedtocline}[5]{%
103   \let\numberline\tocstyle@numberline
104   \ifnum #1>\c@tocdepth \else
      Penalty feature: no page break between higher and lower depths.
105     \if\tocstyle@penalties
106       \begingroup
107         \@tempcnta 20010
108         \advance \@tempcnta by -#1
109         \ifnum \@tempcnta>\lastpenalty
110           \aftergroup\penalty\aftergroup\@lowpenalty
111         \fi
112       \endgroup
113     \fi
      Activation of all features for this TOC and depth:
114     \edef\tocstyledepth{#1}%
115     \tocstyle@activate@features
      Similar to kernel command but if feature entryvskip was set use \addvspace:
116     \ifx\tocstyle@feature@entryvskip\relax
117       \vskip \z@ \@plus.2\p@
118     \else
119       \addvspace{\tocstyle@feature@entryvskip}%
120     \fi
121     {%
      Preinitialization of lengths and skips and then call a hook
122       \parskip \z@ \parindent \z@ \leftskip \z@ \rightskip \z@
123       \tocstyle@feature@raggedhook
      Set number indent to \@tempdimb and text indent to \@tempdima.
124       \@tempdima #3\relax
125       \@tempdimb #2\relax
126 <trace>       \typeout{number indent by \string\l@... (\tocstyleTOC, \tocstyledepth)
127 <trace>       \typeout{text indent by \string\l@... (\tocstyleTOC, \tocstyledepth):
      Calc auto lengths. Use max. of last run of parents if available.
128       \ifnum #1>\z@\relax
129         \@tempcnta #1\relax \advance\@tempcnta \m@ne
130         \ifcsname tocstyle@maxskipwidth\tocstyleTOC @\the\@tempcnta\endcsname
131           \ifcsname tocstyle@maxnumwidth\tocstyleTOC @\the\@tempcnta\endcsname
132             \@tempdimb
133             \csname tocstyle@maxskipwidth\tocstyleTOC @\the\@tempcnta\endcsname
134             \advance\@tempdimb
135             \csname tocstyle@maxnumwidth\tocstyleTOC @\the\@tempcnta\endcsname
136           \fi
137         \fi
138       \fi
139 <trace>       \typeout{number indent by parent (\tocstyleTOC, \tocstyledepth): \spa
140       \ifcsname tocstyle@skipwidth\tocstyleTOC @#1\endcsname

```

```

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

```

Advance instead of set, because of the hook above:

```

184     \advance\parindent \@tempdimb\@afterindenttrue
185     \advance\leftskip \parindent
186     \advance\rightskip \@tocrmarg
187     \parfillskip -\rightskip
188     \ifx\tocstyle@feature@parfillskip\relax\else

```

```

189      \advance\parfillskip \tocstyle@feature@parfillskip\relax
190      \fi
191      \interlinepenalty\@M
192      \leavevmode
193      \advance\leftskip \@tempdima
194      \null\nobreak

```

\hskip\leftskip optional moved to \numberline

```

195      \iftocstyle@indentnotnumbered\else
196      \hskip -\leftskip
197      \fi

```

Change at start of the entry

```

198      \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.

```

199      {#4}\nobreak
200      \ifx\tocstyle@feature@leaders\relax
201      \leaders\hbox{$\m@th
202      \mkern \@dotsep mu\hbox{\tocstyle@feature@dothook .}%
203      \mkern \@dotsep mu$}\hfill
204      \else
205      \tocstyle@feature@leaders
206      \fi
207      \nobreak
208      \ifx\tocstyle@feature@pagenumberbox\relax
209      \hb@xt@\@pnumwidth{\hfil\tocstyle@feature@pagenumberhook #5%
210      \kern-\p@\kern\p@}%
211      \else
212      \tocstyle@feature@pagenumberbox{\tocstyle@feature@pagenumberhook #5%
213      \kern-\p@\kern\p@}%
214      \fi
215      \par
216      }%

```

Last change is, another penalty change:

```

217      \if\tocstyle@penalties
218      \bgroup
219      \@tempcnta 20009
220      \advance\@tempcnta by -#1
221      \edef\reserved@a{\egroup\penalty\the\@tempcnta\relax}%
222      \reserved@a
223      \fi
224      \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.

```

225 \newcommand*{\tocstyle@saved@numberline}{\fi}

```

```

226 \newcommand*{\tocstyle@numberline}[1]{%
227   \begingroup
228     \ifx\tocstyle@feature@spaceafternumber\relax
229       \settowidth\@tempdima{\tocstyle@@numberline{#1}\enskip}%
230     \else
231       \settowidth\@tempdima{\tocstyle@@numberline{#1}}%
232       \advance \@tempdima \tocstyle@feature@spaceafternumber\relax
233     \fi
234     \ifcsname tocstyle@numwidth@\tocstyleTOC @\tocstyledepth\endcsname
235       \ifdim \@tempdima >
236         \csname tocstyle@numwidth@\tocstyleTOC @\tocstyledepth\endcsname\relax
237         \expandafter\xdef\csname tocstyle@numwidth@\tocstyleTOC
238           @\tocstyledepth\endcsname{\the\@tempdima}%
239       \fi
240     \else
241       \expandafter\xdef\csname tocstyle@numwidth@\tocstyleTOC
242         @\tocstyledepth\endcsname{\the\@tempdima}%
243     \fi
244   \endgroup
245   \iftocstyle@indentnotnumbered
246     \hskip -\leftskip
247   \fi
248   \ifcase \tocstyle@indentstyle
249     \hb@xt@\@tempdima{\tocstyle@@numberline{#1}\hfil}%
250   \or
251     \hb@xt@\@tempdima{\tocstyle@@numberline{#1}\hfil}%
252   \else
253     \ifx\tocstyle@feature@spaceafternumber\relax
254       \hbox{\tocstyle@@numberline{#1}\enskip}%
255     \else
256       \hbox{\tocstyle@@numberline{#1}\hskip
257         \tocstyle@feature@spaceafternumber\relax}%
258     \fi
259   \fi
260 }

```

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

```

261 \newcommand*{\tocstyle@@numberline}[1]{%
262   #1\csname autodot\endcsname
263 }

```

9.2.2. Redefining Class Macros

```

\l@part Try to redefine the toc commands at startup.
\l@chapter 264 \AtBeginDocument{%
\l@section 265   \ifpackageloaded{tocbasic}{%
\l@subsection 266     \ifpackagelater{tocbasic}{2016/03/01}{%
\l@subsubsection 267       \PackageWarningNoLine{tocstyle}{%
\l@paragraph 268         Usage of ‘tocstyle’ with new ‘tocbasic’ detected.\MessageBreak
\l@subparagraph
\l@table
\l@figure

```

```

269         This is not an error! You can do this.\MessageBreak
270         Nevertheless, you should note, combining 'tocstyle'\MessageBreak
271         with this version of 'tocbasic' will break several\MessageBreak
272         features of 'tocbasic'. You should use 'tocstyle'\MessageBreak
273         features instead of 'tocbasic' attributes for all\MessageBreak
274         entry changes you want.\MessageBreak
275         You may also get additional warnings because of\MessageBreak
276         redefined '\string\numberline'. Ignore them}%
277     }{}%
278 }{}%
279 \@ifpackageloaded{etoc}{%
280     \PackageWarningNoLine{tocstyle}{%
281         Usage of 'tocstyle' with 'etoc' detected.\MessageBreak
282         I suggest to use either 'tocstyle' or 'etoc'\MessageBreak
283         but not both of them together}%
284 }{}%
285 \@ifpackageloaded{titletoc}{%
286     \PackageWarningNoLine{tocstyle}{%
287         Usage of 'tocstyle' with 'titletoc' detected.\MessageBreak
288         I suggest to use either 'tocstyle' or 'titletoc'\MessageBreak
289         but not both of them together. I expect even\MessageBreak
290         error messages because of this combination}%
291 }{}%
292 \def\tocstyle@test@level{-1}%
293 \def\tocstyle@test@levelname{part}%
294 \@whiles\ifcsname l@\tocstyle@test@levelname\endcsname\fi{%
295     \setbox\@tempboxa\vbox{\hsize\maxdimen
296         \let\normalcolor\relax
297         \let\color\@gobble
298         \edef\reserved@a{%
299             \expandafter\noexpand\csname l@\tocstyle@test@levelname\endcsname{%
300                 \noexpand\tocstyle@l@define{\tocstyle@test@levelname}%
301                 {\tocstyle@test@level}%
302             }{}%
303         }%
304         \reserved@a
305     }%
306     \edef\tocstyle@test@levelname{sub\tocstyle@test@levelname}%
307     \edef\tocstyle@test@level{\the\numexpr\tocstyle@test@level+1}%
308 }%
309 \ifnum\tocstyle@test@level<\z@
310     \def\tocstyle@test@level{0}%
311 \fi
312 \def\tocstyle@test@levelname{chapter}%
313 \@whiles\ifcsname l@\tocstyle@test@levelname\endcsname\fi{%
314     \setbox\@tempboxa\vbox{\hsize\maxdimen
315         \let\normalcolor\relax
316         \let\color\@gobble
317         \edef\reserved@a{%

```

```

318         \expandafter\noexpand\csname l@tocstyle@test@levelname\endcsname{%
319         \noexpand\tocstyle@l@define{\tocstyle@test@levelname}%
320                                     {\tocstyle@test@level}%
321     }{}%
322     }%
323     \reserved@a
324     }%
325     \edef\tocstyle@test@levelname{sub\tocstyle@test@levelname}%
326     \edef\tocstyle@test@level{\the\numexpr\tocstyle@test@level+1}%
327 }%
328 \ifnum\tocstyle@test@level<\@ne
329     \def\tocstyle@test@level{1}%
330 \fi
331 \def\tocstyle@test@levelname{section}%
332 \@whiles\ifcsname l@tocstyle@test@levelname\endcsname\fi{%
333     \setbox\@tempboxa\vbox{\hsize\maxdimen
334         \let\normalcolor\relax
335         \let\color\@gobble
336         \edef\reserved@a{%
337             \expandafter\noexpand\csname l@tocstyle@test@levelname\endcsname{%
338             \noexpand\tocstyle@l@define{\tocstyle@test@levelname}%
339                                     {\tocstyle@test@level}%
340         }{}%
341     }%
342     \reserved@a
343     }%
344     \edef\tocstyle@test@levelname{sub\tocstyle@test@levelname}%
345     \edef\tocstyle@test@level{\the\numexpr\tocstyle@test@level+1}%
346 }%
347 \def\tocstyle@test@levelname{paragraph}%
348 \@whiles\ifcsname l@tocstyle@test@levelname\endcsname\fi{%
349     \setbox\@tempboxa\vbox{\hsize\maxdimen
350         \let\normalcolor\relax
351         \let\color\@gobble
352         \edef\reserved@a{%
353             \expandafter\noexpand\csname l@tocstyle@test@levelname\endcsname{%
354             \noexpand\tocstyle@l@define{\tocstyle@test@levelname}%
355                                     {\tocstyle@test@level}%
356         }{}%
357     }%
358     \reserved@a
359     }%
360     \edef\tocstyle@test@levelname{sub\tocstyle@test@levelname}%
361     \edef\tocstyle@test@level{\the\numexpr\tocstyle@test@level+1}%
362 }%
363 \def\tocstyle@test@level{1}%
364 \def\tocstyle@test@levelname{table}%
365 \@whiles\ifcsname l@tocstyle@test@levelname\endcsname\fi{%
366     \setbox\@tempboxa\vbox{\hsize\maxdimen

```

```

367 \let\normalcolor\relax
368 \let\color\@gobble
369 \edef\reserved@a{%
370 \expandafter\noexpand\csname l@\tocstyle@test@levelname\endcsname{%
371 \noexpand\tocstyle@l@define{\tocstyle@test@levelname}%
372 {\tocstyle@test@level}%
373 }{}%
374 }%
375 \reserved@a
376 }%
377 \edef\tocstyle@test@levelname{sub\tocstyle@test@levelname}%
378 \edef\tocstyle@test@level{\the\numexpr\tocstyle@test@level+1}%
379 }%
380 \def\tocstyle@test@level{1}%
381 \def\tocstyle@test@levelname{figure}%
382 \@whilesw@ifcsname l@\tocstyle@test@levelname\endcsname\fi{%
383 \setbox\@tempboxa\vbox{\hsize\maxdimen
384 \let\normalcolor\relax
385 \let\color\@gobble
386 \edef\reserved@a{%
387 \expandafter\noexpand\csname l@\tocstyle@test@levelname\endcsname{%
388 \noexpand\tocstyle@l@define{\tocstyle@test@levelname}%
389 {\tocstyle@test@level}%
390 }{}%
391 }%
392 \reserved@a
393 }%
394 \edef\tocstyle@test@levelname{sub\tocstyle@test@levelname}%
395 \edef\tocstyle@test@level{\the\numexpr\tocstyle@test@level+1}%
396 }%

```

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

```

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

```

```

413 \fi}%
414 \ifx\@dottedtocline\@tempa\else
415 \def\@tempa#1#2#3#4#5{%
416 \ifnum #1>\c@tocdepth \else
417 \vskip \z@ \@plus.2\p@
418 {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
419 \parindent #2\relax\@afterindenttrue
420 \interlinepenalty\@M
421 \leavevmode
422 \@tempdima #3\relax
423 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
424 {#4}\nobreak
425 \leaders\hbox{$\m@th
426 \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
427 mu$}\hfill
428 \nobreak
429 \hb@xt@\@pnumwidth{\hfil \normalfont \normalcolor #5\kern-\p@\kern\p@}%
430 \par}%
431 \fi}%
432 \ifx\@dottedtocline\@tempa\else
433 \tocstyle@macrochangewarning\@dottedtocline
434 \fi
435 \fi
436 \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:

```

437 \def\@tempa#1{\hb@xt@\@tempdima{#1\autodot\hfil}}%
438 \ifx\numberline\@tempa\else
439 \def\@tempa#1{\hb@xt@\@tempdima{#1\hfil}}%
440 \ifx\numberline\@tempa\else
441 \tocstyle@macrochangewarning\numberline
442 \fi
443 \fi
444 \let\tocstyle@saved@numberline\numberline
445 }

```

`\tocstyle@macrochangewarning`

```

446 \newcommand*{\tocstyle@macrochangewarning}[1]{%
447 \PackageWarningNoLine{tocstyle}{%
448 unexpected \string#1\space definition!\MessageBreak
449 You are either using an unknown LaTeX kernel\MessageBreak
450 version, an unknown class or package, that redefines\MessageBreak
451 \string#1, or a \string#1\space
452 redefinition!\MessageBreak
453 at the document preamble.\MessageBreak
454 Because of this you may get unexpected results!\MessageBreak
455 Maybe it would be better not to use package tocstyle}%

```

```

456 \PackageInfo{tocstyle}{Unexpected definition is:\MessageBreak
457   \meaning#1}%
458 }

```

```

\tocstyle@l@define
\tocstyle@activate@all@l 459 \newcommand*{\tocstyle@activate@all@l}{%
460 \newcommand*{\tocstyle@l@define}[2]{%
461   \advance\leftskip-\@tempdima
462   \edef\@tempa{%
463     \noexpand\global\noexpand\let
464     \expandafter\noexpand\csname tocstyle@sav@l@#1\endcsname
465     \expandafter\noexpand\csname l@#1\endcsname
466     \noexpand\gdef
467     \expandafter\noexpand\csname tocstyle@l@#1\endcsname{%
468       \noexpand\@dottedtocline{#2}{\the\leftskip}{\the\@tempdima}}%
469     \noexpand\g@addto@macro\noexpand\tocstyle@activate@all@l{%
470       \noexpand\let\expandafter\noexpand\csname l@#1\endcsname
471       \expandafter\noexpand\csname tocstyle@l@#1\endcsname
472     }%
473   }%
474   \PackageInfo{tocstyle}{prepare \expandafter\string
475     \csname l@#1\endcsname\space for redefinition}%
476   \@tempa
477 }

```

9.2.3. New Macros

\showtoc

```

478 \newcommand*{\showtoc}[2][\aliastoc\tocstyleTOC\tocstyleAliasTOC]{%
479   \ifcsname tocstyle@copyn@me@#2\endcsname
480     \@tempcnta \csname tocstyle@copyn@me@#2\endcsname\relax
481     \advance\@tempcnta \@ne
482     \expandafter\xdef\csname tocstyle@copyn@me@#2\endcsname{\the\@tempcnta}%
483   \else
484     \expandafter\xdef\csname tocstyle@copyn@me@#2\endcsname{1}%
485   \fi
486   \ifx\@do@filelist\relax\let\@do@filelist\empty\fi
487   \edef\@tempa{\noexpand\g@addto@macro\noexpand\@do@filelist{%
488     \noexpand\tocstyle@copy@toc{#2}{\csname
489       tocstyle@copyn@me@#2\endcsname}}}%
490   }\@tempa%
491   \begin@group
492     \edef\tocstyleAliasTOC{#2}%
493     \edef\tocstyleTOC{#2\csname tocstyle@copyn@me@#2\endcsname}%
494     #1
495     \tocstyle@pre@starttoc{#2\csname tocstyle@copyn@me@#2\endcsname}%
496     \makeatletter
497     \@input{\jobname.#2\csname tocstyle@copyn@me@#2\endcsname}%
498     \@n@breakfalse

```

```

499     \tocstyle@post@starttoc{#2\csname tocstyle@copypname@#2\endcsname}%
500 \endgroup
501 }

```

\tocstyle@copy@toc

```

502 \newcommand*{\tocstyle@copy@toc}[2]{%
503   \if@filesw
504     \begingroup
505       \endlinechar=\m@ne
506       While LATEX does not close the files, we have to do it know.
507       \immediate\closeout\csname tf@#1\endcsname
508       \immediate\openin\@inputcheck \jobname.#1
509       \immediate\openout\@partaux \jobname.#1#2
510       \loop\unless\ifeof\@inputcheck
511         \immediate\readline\@inputcheck to \@tempa
512         \immediate\write\@partaux{\@tempa}%
513       \repeat
514       \immediate\closeout\@partaux
515       \immediate\closein\@inputcheck
516     \endgroup
517   \fi
518 }

```

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

```

518 \newcommand*{\aliastoc}[2]{%
519   \expandafter\edef\csname tocstyle@alias@TOC@#1\endcsname{#2}%
520 }

```

\tocstyle@pre@starttoc Commands before and after the original \@starttoc.

```

\tocstyle@post@starttoc 521 \newcommand*{\tocstyle@pre@starttoc}[1]{%
522   \begingroup
523     \expandafter\ifx\csname tocstyle@deactivated@\endcsname\relax
524       \expandafter\ifx\csname tocstyle@deactivated@#1\endcsname\relax\relax
525         \tocstyle@activetrue
526       \else
527         \tocstyle@activefalse
528       \fi
529     \else
530       \tocstyle@activefalse
531     \fi
532     \iftocstyle@active
533       \let\@dottedtocline\tocstyle@dottedtocline
534       \parskip \z@
535       \parindent \z@
536       \parfillskip \z@\@plus 1fil
537       \ifcsname tocstyle@alias@TOC@#1\endcsname
538         \edef\tocstyleAliasTOC{\csname tocstyle@alias@TOC@#1\endcsname}%
539       \else

```

```

540         \edef\tocstyleAliasTOC{#1}%
541     \fi
542     \edef\tocstyleTOC{#1}%
543     \tocstyle@activate@all@l
544 \fi
545 }
546 \newcommand*{\tocstyle@post@starttoc}[1]{%
547     \iftocstyle@active
548     \if@files
549         \ifcsname tocstyle@unumwidth@#1\endcsname
550             \protected@write\@auxout{}{%
551                 \protect\tocstyle@set@width{unum}{#1}{}%
552                 \csname tocstyle@unumwidth@#1\endcsname}%
553             }%
554         \fi
555         \expandafter\let\expandafter\@tempa
556             \csname tocstyle@depthlist@#1\endcsname
557         \ifx\@tempa\relax\else
558             \expandafter\@for \expandafter\@tempa\expandafter:\expandafter=\@tempa
559             \do {%
560                 \ifcsname tocstyle@numwidth@#1\@tempa\endcsname
561                     \protected@write\@auxout{}{%
562                         \protect\tocstyle@set@width{num}{#1}{\@tempa}%
563                         \csname tocstyle@numwidth@#1\@tempa\endcsname}%
564                     }%
565                 \fi
566                 \ifcsname tocstyle@skipwidth@#1\@tempa\endcsname
567                     \protected@write\@auxout{}{%
568                         \protect\tocstyle@set@width{skip}{#1}{\@tempa}%
569                         \csname tocstyle@skipwidth@#1\@tempa\endcsname}%
570                     }%
571                 \fi
572             }%
573         \fi
574     \fi
575 \endgroup
576 \endgroup
577 }

```

`\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!

```

578 \newcommand*{\tocstyle@set@width}[4]{%
579     \iftocstyle@indentnotnumbered
580         \ifdim #4<\z@
581             \expandafter\gdef\csname tocstyle@max#1width@#2@#3\endcsname{%
582                 \dimexpr #4/2\relax}%
583         \else

```

```

584     \expandafter\gdef\csname tocstyle@max#1width@#2@#3\endcsname{#4}%
585     \fi
586   \else
587     \ifdim #4<\z@
588       \expandafter\gdef\csname tocstyle@max#1width@#2@#3\endcsname{\z@}%
589     \else
590       \expandafter\gdef\csname tocstyle@max#1width@#2@#3\endcsname{#4}%
591     \fi
592   \fi
593 }
594 \AtBeginDocument{%
595   \if@filesw
596     \immediate\write\@auxout{%
597       \string\providecommand*\string\tocstyle@set@width[4]{}%
598     }%
599   \fi
600 }

```

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

```

\tocstyleAliasTOC 601 \newcommand*{\tocstyleTOC}{}
602 \newcommand*{\tocstyleAliasTOC}{}

```

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

```

603 \newcommand*{\tocstyledepth}{}

```

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

```

\reactivatetocstyle 604 \newif\iftocstyle@active
605 \newcommand*{\deactivatetocstyle}[1] []{%
606   \expandafter\let\csname tocstyle@deactivated@#1\endcsname\@empty}
607 \newcommand*{\reactivatetocstyle}[1] []{%
608   \expandafter\let\csname tocstyle@deactivated@#1\endcsname\relax}

```

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

```

\@settocfeature 609 \newcommand*{\@settocfeature}[1] []{%
\@@@settocfeature 610   \kernel@ifnextchar[ {\@@settocfeature[{#1}]}{\@@settocfeature[{#1}] []}
611 }
612 \def\@@settocfeature[#1][#2]#3#4{%
613   \trace \typeout{exclude: \tocstyle@feature@excludelist}%
614   \@expandtwoargs\in@{,#3,}{,\tocstyle@feature@excludelist,}%
615   \ifin@ \else
616     \expandafter\ifcsname tocstyle@feature@#3\endcsname
617     \@namedef{tocstyle@feature@#3@#1@#2}{#4}%
618     \begingroup
619       \expandafter\let\expandafter\@tempa
620       \csname tocstyle@commandlist@#1\endcsname
621       \@expandtwoargs\in@{,\tocstyle@feature@#3@#1@#2,}{,\@tempa,}%
622       \ifin@\let\@tempa\endgroup\else
623         \edef\@tempa{\endgroup
624           \noexpand\expandafter\noexpand\ifx

```

```

625         \noexpand\csname tocstyle@commandlist@#1\noexpand\endcsname\relax
626         \noexpand\expandafter\noexpand\expandafter\noexpand\expandafter
627         \noexpand\def
628         \noexpand\else
629         \noexpand\expandafter\noexpand\expandafter\noexpand\expandafter
630         \noexpand\l@addto@macro
631         \noexpand\fi
632         \noexpand\csname tocstyle@commandlist@#1\noexpand\endcsname%
633         {tocstyle@feature@#3@#1@#2,}%
634     \fi
635     \@tempa
636 \else
637     \PackageError{tocstyle}{unkown feature ‘#3’}{%
638         You’ve told me to set up toc style feature ‘#3’,\MessageBreak
639         but I don’t know this feature.\MessageBreak
640         See the tocstyle manual for all known feature.\MessageBreak
641     }%
642 \fi
643 \fi
644 }
645 \newcommand*{\settocfeature}{}
646 \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.

```

647 \providecommand{\l@addto@macro}[2]{%
648     \edef#1{\unexpanded\expandafter{#1#2}}%
649 }%

```

`\settocstylefeature` Same as above without TOC argument.

```

\settocstylefeature 650 \newcommand*{\@settocstylefeature}{%
651     \kernel@ifnextchar[ {\@settocfeature[]} {\@settocfeature[] []}%
652 }
653 \newcommand*{\settocstylefeature}{}
654 \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.

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

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

`\tocstyle@activate@features` Activates the features

```

655 \newcommand*{\tocstyle@activate@features}{%
656     \expandafter\ifx\csname tocstyle@depthlist@\tocstyleTOC\endcsname\relax

```

```

657 \expandafter\xdef\csname tocstyle@depthlist@\tocstyleTOC\endcsname{%
658 \tocstyledepth}%
659 \else
660 \expandafter\let\expandafter\@tempa
661 \csname tocstyle@depthlist@\tocstyleTOC\endcsname
662 \@expandtwoargs\in@{,\tocstyledepth,}{,\@tempa,}%
663 \ifin@\else
664 \expandafter\xdef\csname tocstyle@depthlist@\tocstyleTOC\endcsname{%
665 \csname tocstyle@depthlist@\tocstyleTOC\endcsname,\tocstyledepth}%
666 \fi
667 \fi
668 \expandafter\@for \expandafter\@tempa
669 \expandafter:\expandafter=\tocstyle@featurelist \do
670 {%
671 \@ifundefined{tocstyle@feature@\@tempa @\tocstyleAliasTOC @\tocstyledepth}{%
672 \@ifundefined{tocstyle@feature@\@tempa @@\tocstyledepth}{%
673 \@ifundefined{tocstyle@feature@\@tempa @\tocstyleAliasTOC @}{%
674 \@ifundefined{tocstyle@feature@\@tempa @@}{%
675 \expandafter\let\csname tocstyle@feature@\@tempa\endcsname\relax
676 }{%
677 \expandafter\let\csname tocstyle@feature@\@tempa
678 \expandafter\endcsname
679 \csname tocstyle@feature@\@tempa @@\endcsname
680 }%
681 }{%
682 \expandafter\let\csname tocstyle@feature@\@tempa
683 \expandafter\endcsname
684 \csname tocstyle@feature@\@tempa @\tocstyleAliasTOC @\endcsname
685 }%
686 }{%
687 \expandafter\let\csname tocstyle@feature@\@tempa
688 \expandafter\endcsname
689 \csname tocstyle@feature@\@tempa @@\tocstyledepth\endcsname
690 }%
691 }{%
692 \expandafter\let\csname tocstyle@feature@\@tempa
693 \expandafter\endcsname
694 \csname tocstyle@feature@\@tempa @\tocstyleAliasTOC @\tocstyledepth\endcsname
695 }%
696 }%
697 }

```

\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.

```

698 \newcommand*{\newtocstyle}{%
699 \kernel@ifnextchar [{\@newtocstyle}{\@newtocstyle[]}]
700 \newcommand*{\@newtocstyle}{%

```

```

701 \def\@newtocstyle[#1]{%
702   \kernel@ifnextchar [{\@newtocstyle[{#1}]}{\@newtocstyle[{#1}] []}]
703   \newcommand*{\@newtocstyle}{%
704   \def\@newtocstyle[#1][#2]#3#4{%
705     \@ifundefined{tocstyle@style@#3}{%
706       \@ifundefined{tocstyle@style@#1}{%
707         \ifx \relax#1\relax\else
708           \PackageError{tocstyle}{unknown parent TOC style ‘#1’}{%
709             You’ve told me to inheritate parent TOC style ‘#1’,\MessageBreak
710             but there’s no TOC style ‘#1’ defined.}%
711         \fi
712         \expandafter\def\csname tocstyle@style@#3\endcsname{#4}%
713       }{%
714         \expandafter\def\csname tocstyle@style@#3\endcsname{%
715           \edef\reserved@a{%
716             \noexpand\l@addto@macro\noexpand\tocstyle@feature@excludelist{#2}%
717             \noexpand\@usetocstyle{#1}%
718             \noexpand\def\noexpand\tocstyle@feature@excludelist{%
719               \tocstyle@feature@excludelist}%
720           }\reserved@a
721           #4%
722         }%
723       }%
724     }{%
725       \PackageError{tocstyle}{TOC style ‘#3’ already defined}{%
726         You’ve tried to define a new TOC style ‘#3’,\MessageBreak
727         but there’s already a TOC style named ‘#3’.}%
728     }%
729 }
730 \newcommand*{\tocstyle@feature@excludelist}{%

```

`\usetocstyle` Use the predefined TOC style. You may define `\tocstyle@deprecated@style@foo` 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.

```

731 \newcommand*{\usetocstyle}[2][]{%
732   \@ifundefined{tocstyle@deprecated@style@#2}{%
733     \@ifundefined{tocstyle@style@#2}{%
734       \PackageError{tocstyle}{unknown TOC style ‘#2’}{%
735         You’ve told me to use TOC style ‘#2’,\MessageBreak
736         but there’s no TOC style ‘#2’ defined.}%
737     }{%
738       \def\settocfeature{%
739         \kernel@ifnextchar[ %]
740         {\@settocfeature[{#1}]}{\@settocfeature[{#1}] []}%
741       }%
742       \let\settocstylefeature\settocfeature

```

Deactivate all known features for this TOC

```

743 \expandafter\ifx\csname tocstyle@commandlist@#1\endcsname\relax
744 \else
745 \expandafter\expandafter\expandafter\@for
746 \expandafter\expandafter\expandafter\@tempa
747 \expandafter\expandafter\expandafter:%
748 \expandafter\expandafter\expandafter=%
749 \csname tocstyle@commandlist@#1\endcsname
750 \do{%
751 \expandafter\let\csname \@tempa\endcsname\relax
752 }%
```

So there are no more known features for this TOC.

```

753 \expandafter\let\csname tocstyle@commandlist@#1\endcsname\relax
754 \fi
```

Activate all known features for this style and TOC

```

755 \@usetocstyle{#2}%
756 \let\settocfeature\@settocfeature
757 \let\settocstylefeature\@settocstylefeature
758 }%
759 }{%
760 \expandafter\ifx\csname tocstyle@deprecated@style@#2\endcsname\@empty
761 \PackageWarning{tocstyle}{%
762 deprecated TOC style '#2'!\MessageBreak
763 You should not longer use this style,\MessageBreak
764 because it will be removed soon.\MessageBreak
765 You should select another TOC style}%
766 \usetocstyle[{#1}]{deprecated@#2}%
767 \else
768 \PackageWarning{tocstyle}{%
769 deprecated TOC style '#2'!\MessageBreak
770 You should use TOC style '\csname
771 tocstyle@deprecated@style@#2\endcsname'\MessageBreak
772 instead of '#2'}%
773 \fi
774 }%
775 }
776 \newcommand*{\@usetocstyle}[1]{%
777 \csname tocstyle@style@#1\endcsname
778 }
```

\tocstyle@featurelist Comma seperated list of all known features

```

779 \newcommand*{\tocstyle@featurelist}{%
780 pagenumberhook,entryhook,dothook,entryvskip,leaders,raggedhook,%
781 spaceafternumber,parfillskip,pagenumberbox,%
782 }
```

\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

```
783 \newcommand*{\tocstyle@feature@pagenumberhook}{}
```

```

784 \let\tocstyle@feature@pagenumberhook\relax
785 \newcommand*{\tocstyle@feature@pagenumberbox}{}
786 \let\tocstyle@feature@pagenumberbox\relax
787 \newcommand*{\tocstyle@feature@entryhook}{}
788 \let\tocstyle@feature@entryhook\relax
789 \newcommand*{\tocstyle@feature@dothook}{}
790 \let\tocstyle@feature@dothook\relax
791 \newcommand*{\tocstyle@feature@entryvskip}{}
792 \let\tocstyle@feature@entryvskip\relax
793 \newcommand*{\tocstyle@feature@leaders}{}
794 \let\tocstyle@feature@leaders\relax
795 \newcommand*{\tocstyle@feature@parfillskip}{}
796 \let\tocstyle@feature@parfillskip\relax
797 \newcommand*{\tocstyle@feature@raggedhook}{}
798 \let\tocstyle@feature@raggedhook\relax
799 \newcommand*{\tocstyle@feature@spaceafternumber}{}
800 \let\tocstyle@feature@spaceafternumber\relax

```

`\iftochasdepth` Uses `\tocstyle@depthlist@<TOC>` to test, if the TOC has the depth already.

```

801 \newcommand*{\iftochasdepth}[2]{%
802   \begingroup
803     \expandafter\let\expandafter\@tempa\csname tocstyle@depthlist@#1\endcsname
804     \ifx\@tempa\relax
805       \aftergroup\@secondoftwo
806     \else
807       \@expandtwoargs\in@{,#2,}{,\@tempa}%
808       \expandafter\aftergroup\ifin@
809       \@firstoftwo
810     \else
811       \@secondoftwo
812     \fi
813   \fi
814 \endgroup
815 }

```

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.

```

816 \providecommand*{\ext@toc}{toc}

817 \newtocstyle{standard}{%
818   \settocfeature{dothook}{\normalfont}%
819   \settocfeature[-1]{entryhook}{\bfseries}%
820   \settocfeature[-1]{entryvskip}{2.25em\@plus\p@}%
821   \settocfeature[-1]{leaders}{\hfill}%
822   \settocfeature[0]{entryvskip}{1em\@plus\p@}%
823   \settocfeature[0]{leaders}{\hfill}%

```

```

824 \settocfeature[0]{entryhook}{\bfseries}
825 \iftochaschapter\else
826   \settocfeature[1]{entryvskip}{1em\@plus\p@}%
827   \settocfeature[1]{leaders}{\hfill}%
828   \settocfeature[1]{entryhook}{%
829     \ifx\tocstyleAliasTOC\ext@toc\bfseries\fi
830   }%
831 \fi
832 }
833 \begingroup\expandafter\expandafter\expandafter\endgroup
834 \expandafter\ifx\csname KOMAClassName\endcsname\relax
835   \newtocstyle{KOMAlike}{%
836     \settocfeature{dothook}{\normalfont}%
837     \settocfeature[-1]{entryhook}{\sffamily\bfseries}%
838     \settocfeature[-1]{entryvskip}{2.25em\@plus\p@}%
839     \settocfeature[-1]{leaders}{\hfill}%
840     \settocfeature[-1]{pagenumberhook}{\sffamily\bfseries}%
841     \settocfeature[0]{entryvskip}{1em\@plus\p@}%
842     \settocfeature[0]{leaders}{\hfill}%
843     \settocfeature[0]{entryhook}{\sffamily\bfseries}
844     \settocfeature[0]{pagenumberhook}{\sffamily\bfseries}%
845     \iftochaschapter\else
846       \settocfeature[1]{entryvskip}{1em\@plus\p@}%
847       \settocfeature[1]{leaders}{\hfill}%
848       \settocfeature[1]{entryhook}{%
849         \ifx\tocstyleAliasTOC\ext@toc\sffamily\bfseries\fi
850       }%
851       \settocfeature[1]{pagenumberhook}{%
852         \ifx\tocstyleAliasTOC\ext@toc\sffamily\bfseries\fi
853       }%
854     \fi
855   }
856 \else
857   \newtocstyle{KOMAlike}{%
858     \settocfeature{dothook}{\normalfont}%
859     \settocfeature[-1]{entryhook}{\usekomafont{partentry}}%
860     \settocfeature[-1]{entryvskip}{2.25em\@plus\p@}%
861     \settocfeature[-1]{leaders}{\hfill}%
862     \settocfeature[-1]{pagenumberhook}{\usekomafont{partentrypagenumber}}%
863     \settocfeature[0]{entryvskip}{1em\@plus\p@}%
864     \settocfeature[0]{leaders}{\hfill}%
865     \settocfeature[0]{entryhook}{\usekomafont{chapterentry}}%
866     \settocfeature[0]{pagenumberhook}{\usekomafont{chapterentrypagenumber}}%
867     \iftochaschapter\else
868       \settocfeature[1]{entryvskip}{1em\@plus\p@}%
869       \settocfeature[1]{leaders}{\hfill}%
870       \settocfeature[1]{entryhook}{%
871         \begingroup
872         \ifx\tocstyleAliasTOC\ext@toc

```

```

873         \def\@tempa{\endgroup\usekomafont{sectionentry}}}%
874     \else
875         \let\@tempa\endgroup
876     \fi
877     \@tempa
878 }%
879 \settocfeature[1]{pagenumberhook}{%
880     \beginingroup
881     \ifx\tocstyleAliasTOC\ext@toc
882         \def\@tempa{\endgroup\usekomafont{sectionentrypagenumber}}}%
883     \else
884         \let\@tempa\endgroup
885     \fi
886     \@tempa
887 }%
888 \fi
889 }
890 \fi
891 \newcommand*{\tocstyle@deprecated@style@KOMAScript}{KOMAlike}%
892 \newtocstyle[KOMAlike]{classic}{%
893     \settocfeature[-1]{pagenumberhook}{\normalfont\normalcolor}%
894     \settocfeature[0]{pagenumberhook}{\normalfont\normalcolor}%
895     \iftochaschapter\else
896         \settocfeature[1]{pagenumberhook}{\normalfont\normalcolor}%
897     \fi
898     \settocfeature{pagenumberhook}{\normalfont\normalcolor}%
899     \settocfeature{raggedhook}{\raggedright}%
900 }
901 \newtocstyle[classic][leaders]{allwithdot}{%
902     \newtocstyle[allwithdot]{noonewithdot}{%
903         \settocfeature{leaders}{\hfill}%
904     }
905     \newtocstyle[classic][leaders]{nopagecolumn}{%
906         \settocfeature{leaders}{\quad}%
907         \settocfeature{parfillskip}{\z@ plus 1fil}%
908         \settocfeature{pagenumberbox}{\hbox}%
909     }

```

9.2.5. Defining Some TOC Styles

Loading a optional configuration file.

```

910 \InputIfFileExists{tocstyle.cfg}{%
911     \PackageInfo{tocstyle}{using tocstyle.cfg}%
912 }{%
913     \PackageInfo{tocstyle}{no tocstyle.cfg found}%
914 }

```

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. Package Status	2
2. How It Works	3
3. Optional Features	3
4. Using TOC Styles	6
5. Setting-up Single Features	8
6. Defining New TOC Styles	10
7. Processing a TOC	10
8. Configuration file	11
9. Implementation	12
9.1. Option	12
9.2. Body	14
9.2.1. Redefining L ^A T _E X Kernel Macros	14
9.2.2. Redefining Class Macros	18
9.2.3. New Macros	23
9.2.4. Defining Some TOC Styles	31
9.2.5. Defining Some TOC Styles	33
A. Examples for the Different TOC Styles	34
A.1. Graduated Versions	34
A.2. Flat Versions	40
A.3. Fullflat Versions	46

A.1.2. KOMALike with Option `tocgraduated`

1. Package Status	2
2. How It Works	3
3. Optional Features	3
4. Using TOC Styles	6
5. Setting-up Single Features	8
6. Defining New TOC Styles	10
7. Processing a TOC	10
8. Configuration file	11
9. Implementation	12
9.1. Option	12
9.2. Body	14
9.2.1. Redefining L ^A T _E X Kernel Macros	14
9.2.2. Redefining Class Macros	18
9.2.3. New Macros	23
9.2.4. Defining Some TOC Styles	31
9.2.5. Defining Some TOC Styles	33
A. Examples for the Different TOC Styles	34
A.1. Graduated Versions	34
A.2. Flat Versions	40
A.3. Fullflat Versions	46

A.1.3. classic with Option tocgraduated

1. Package Status	2
2. How It Works	3
3. Optional Features	3
4. Using TOC Styles	6
5. Setting-up Single Features	8
6. Defining New TOC Styles	10
7. Processing a TOC	10
8. Configuration file	11
9. Implementation	12
9.1. Option	12
9.2. Body	14
9.2.1. Redefining L ^A T _E X Kernel Macros	14
9.2.2. Redefining Class Macros	18
9.2.3. New Macros	23
9.2.4. Defining Some TOC Styles	31
9.2.5. Defining Some TOC Styles	33
A. Examples for the Different TOC Styles	34
A.1. Graduated Versions	34
A.2. Flat Versions	40
A.3. Fullflat Versions	46

A.1.4. allwithdot with Option tocgraduated

1. Package Status	2
2. How It Works	3
3. Optional Features	3
4. Using TOC Styles	6
5. Setting-up Single Features	8
6. Defining New TOC Styles	10
7. Processing a TOC	10
8. Configuration file	11
9. Implementation	12
9.1. Option	12
9.2. Body	14
9.2.1. Redefining L ^A T _E X Kernel Macros	14
9.2.2. Redefining Class Macros	18
9.2.3. New Macros	23
9.2.4. Defining Some TOC Styles	31
9.2.5. Defining Some TOC Styles	33
A. Examples for the Different TOC Styles	34
A.1. Graduated Versions	34
A.2. Flat Versions	40
A.3. Fullflat Versions	46

A.1.5. noonewithdot with Option tocgraduated

1. Package Status	2
2. How It Works	3
3. Optional Features	3
4. Using TOC Styles	6
5. Setting-up Single Features	8
6. Defining New TOC Styles	10
7. Processing a TOC	10
8. Configuration file	11
9. Implementation	12
9.1. Option	12
9.2. Body	14
9.2.1. Redefining L ^A T _E X Kernel Macros	14
9.2.2. Redefining Class Macros	18
9.2.3. New Macros	23
9.2.4. Defining Some TOC Styles	31
9.2.5. Defining Some TOC Styles	33
A. Examples for the Different TOC Styles	34
A.1. Graduated Versions	34
A.2. Flat Versions	40
A.3. Fullflat Versions	46

A.1.6. nopagecolumn with Option tocgraduated

1. Package Status	2
2. How It Works	3
3. Optional Features	3
4. Using TOC Styles	6
5. Setting-up Single Features	8
6. Defining New TOC Styles	10
7. Processing a TOC	10
8. Configuration file	11
9. Implementation	12
9.1. Option	12
9.2. Body	14
9.2.1. Redefining L ^A T _E X Kernel Macros	14
9.2.2. Redefining Class Macros	18
9.2.3. New Macros	23
9.2.4. Defining Some TOC Styles	31
9.2.5. Defining Some TOC Styles	33
A. Examples for the Different TOC Styles	34
A.1. Graduated Versions	34
A.2. Flat Versions	40
A.3. Fullflat Versions	46

A.2. Flat Versions

Now, all flat versions of the table of contents

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

1.	Package Status	2
2.	How It Works	3
3.	Optional Features	3
4.	Using TOC Styles	6
5.	Setting-up Single Features	8
6.	Defining New TOC Styles	10
7.	Processing a TOC	10
8.	Configuration file	11
9.	Implementation	12
9.1.	Option	12
9.2.	Body	14
9.2.1.	Redefining L ^A T _E X Kernel Macros	14
9.2.2.	Redefining Class Macros	18
9.2.3.	New Macros	23
9.2.4.	Defining Some TOC Styles	31
9.2.5.	Defining Some TOC Styles	33
A.	Examples for the Different TOC Styles	34
A.1.	Graduated Versions	34
A.2.	Flat Versions	40
A.3.	Fullflat Versions	46

A.2.2. KOMAlike with Option `tocflat`

1.	Package Status	2
2.	How It Works	3
3.	Optional Features	3
4.	Using TOC Styles	6
5.	Setting-up Single Features	8
6.	Defining New TOC Styles	10
7.	Processing a TOC	10
8.	Configuration file	11
9.	Implementation	12
9.1.	Option	12
9.2.	Body	14
9.2.1.	Redefining \LaTeX Kernel Macros	14
9.2.2.	Redefining Class Macros	18
9.2.3.	New Macros	23
9.2.4.	Defining Some TOC Styles	31
9.2.5.	Defining Some TOC Styles	33
A.	Examples for the Different TOC Styles	34
A.1.	Graduated Versions	34
A.2.	Flat Versions	40
A.3.	Fullflat Versions	46

A.2.3. classic with Option tocflat

1.	Package Status	2
2.	How It Works	3
3.	Optional Features	3
4.	Using TOC Styles	6
5.	Setting-up Single Features	8
6.	Defining New TOC Styles	10
7.	Processing a TOC	10
8.	Configuration file	11
9.	Implementation	12
9.1.	Option	12
9.2.	Body	14
9.2.1.	Redefining L ^A T _E X Kernel Macros	14
9.2.2.	Redefining Class Macros	18
9.2.3.	New Macros	23
9.2.4.	Defining Some TOC Styles	31
9.2.5.	Defining Some TOC Styles	33
A.	Examples for the Different TOC Styles	34
A.1.	Graduated Versions	34
A.2.	Flat Versions	40
A.3.	Fullflat Versions	46

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

1.	Package Status	2
2.	How It Works	3
3.	Optional Features	3
4.	Using TOC Styles	6
5.	Setting-up Single Features	8
6.	Defining New TOC Styles	10
7.	Processing a TOC	10
8.	Configuration file	11
9.	Implementation	12
9.1.	Option	12
9.2.	Body	14
9.2.1.	Redefining \LaTeX Kernel Macros	14
9.2.2.	Redefining Class Macros	18
9.2.3.	New Macros	23
9.2.4.	Defining Some TOC Styles	31
9.2.5.	Defining Some TOC Styles	33
A.	Examples for the Different TOC Styles	34
A.1.	Graduated Versions	34
A.2.	Flat Versions	40
A.3.	Fullflat Versions	46

A.2.5. noonewithdot with Option tocflat

1.	Package Status	2
2.	How It Works	3
3.	Optional Features	3
4.	Using TOC Styles	6
5.	Setting-up Single Features	8
6.	Defining New TOC Styles	10
7.	Processing a TOC	10
8.	Configuration file	11
9.	Implementation	12
9.1.	Option	12
9.2.	Body	14
9.2.1.	Redefining L ^A T _E X Kernel Macros	14
9.2.2.	Redefining Class Macros	18
9.2.3.	New Macros	23
9.2.4.	Defining Some TOC Styles	31
9.2.5.	Defining Some TOC Styles	33
A.	Examples for the Different TOC Styles	34
A.1.	Graduated Versions	34
A.2.	Flat Versions	40
A.3.	Fullflat Versions	46

A.2.6. nopagecolumn with Option tocflat

1.	Package Status	2
2.	How It Works	3
3.	Optional Features	3
4.	Using TOC Styles	6
5.	Setting-up Single Features	8
6.	Defining New TOC Styles	10
7.	Processing a TOC	10
8.	Configuration file	11
9.	Implementation	12
9.1.	Option	12
9.2.	Body	14
9.2.1.	Redefining L ^A T _E X Kernel Macros	14
9.2.2.	Redefining Class Macros	18
9.2.3.	New Macros	23
9.2.4.	Defining Some TOC Styles	31
9.2.5.	Defining Some TOC Styles	33
A.	Examples for the Different TOC Styles	34
A.1.	Graduated Versions	34
A.2.	Flat Versions	40
A.3.	Fullflat Versions	46

A.3. Fullflat Versions

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

A.3.1. standard with Option `tocfullflat`

1. Package Status	2
2. How It Works	3
3. Optional Features	3
4. Using TOC Styles	6
5. Setting-up Single Features	8
6. Defining New TOC Styles	10
7. Processing a TOC	10
8. Configuration file	11
9. Implementation	12
9.1. Option	12
9.2. Body	14
9.2.1. Redefining L ^A T _E X Kernel Macros	14
9.2.2. Redefining Class Macros	18
9.2.3. New Macros	23
9.2.4. Defining Some TOC Styles	31
9.2.5. Defining Some TOC Styles	33
A. Examples for the Different TOC Styles	34
A.1. Graduated Versions	34
A.2. Flat Versions	40
A.3. Fullflat Versions	46

A.3.2. KOMALike with Option `tocfullflat`

1. Package Status	2
2. How It Works	3
3. Optional Features	3
4. Using TOC Styles	6
5. Setting-up Single Features	8
6. Defining New TOC Styles	10
7. Processing a TOC	10
8. Configuration file	11
9. Implementation	12
9.1. Option	12
9.2. Body	14
9.2.1. Redefining L ^A T _E X Kernel Macros	14
9.2.2. Redefining Class Macros	18
9.2.3. New Macros	23
9.2.4. Defining Some TOC Styles	31
9.2.5. Defining Some TOC Styles	33
A. Examples for the Different TOC Styles	34
A.1. Graduated Versions	34
A.2. Flat Versions	40
A.3. Fullflat Versions	46

A.3.3. classic with Option tocfullflat

1. Package Status	2
2. How It Works	3
3. Optional Features	3
4. Using TOC Styles	6
5. Setting-up Single Features	8
6. Defining New TOC Styles	10
7. Processing a TOC	10
8. Configuration file	11
9. Implementation	12
9.1. Option	12
9.2. Body	14
9.2.1. Redefining L ^A T _E X Kernel Macros	14
9.2.2. Redefining Class Macros	18
9.2.3. New Macros	23
9.2.4. Defining Some TOC Styles	31
9.2.5. Defining Some TOC Styles	33
A. Examples for the Different TOC Styles	34
A.1. Graduated Versions	34
A.2. Flat Versions	40
A.3. Fullflat Versions	46

A.3.4. allwithdot with Option tocfullflat

1. Package Status	2
2. How It Works	3
3. Optional Features	3
4. Using TOC Styles	6
5. Setting-up Single Features	8
6. Defining New TOC Styles	10
7. Processing a TOC	10
8. Configuration file	11
9. Implementation	12
9.1. Option	12
9.2. Body	14
9.2.1. Redefining L ^A T _E X Kernel Macros	14
9.2.2. Redefining Class Macros	18
9.2.3. New Macros	23
9.2.4. Defining Some TOC Styles	31
9.2.5. Defining Some TOC Styles	33
A. Examples for the Different TOC Styles	34
A.1. Graduated Versions	34
A.2. Flat Versions	40
A.3. Fullflat Versions	46

A.3.5. noonewithdot with Option tocfullflat

1. Package Status	2
2. How It Works	3
3. Optional Features	3
4. Using TOC Styles	6
5. Setting-up Single Features	8
6. Defining New TOC Styles	10
7. Processing a TOC	10
8. Configuration file	11
9. Implementation	12
9.1. Option	12
9.2. Body	14
9.2.1. Redefining L ^A T _E X Kernel Macros	14
9.2.2. Redefining Class Macros	18
9.2.3. New Macros	23
9.2.4. Defining Some TOC Styles	31
9.2.5. Defining Some TOC Styles	33
A. Examples for the Different TOC Styles	34
A.1. Graduated Versions	34
A.2. Flat Versions	40
A.3. Fullflat Versions	46

A.3.6. nopagecolumn with Option tocfullflat

1. Package Status	2
2. How It Works	3
3. Optional Features	3
4. Using TOC Styles	6
5. Setting-up Single Features	8
6. Defining New TOC Styles	10
7. Processing a TOC	10
8. Configuration file	11
9. Implementation	12
9.1. Option	12
9.2. Body	14
9.2.1. Redefining L ^A T _E X Kernel Macros	14
9.2.2. Redefining Class Macros	18
9.2.3. New Macros	23
9.2.4. Defining Some TOC Styles	31
9.2.5. Defining Some TOC Styles	33
A. Examples for the Different TOC Styles	34
A.1. Graduated Versions	34
A.2. Flat Versions	40
A.3. Fullflat Versions	46

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	
\@@settocfeature	<u>609</u>
\@dottedtocline	<u>397</u>
\@settocfeature	<u>609</u>
\@settocstylefeature	<u>650</u>
\@starttoc	<i>10</i> , <u>94</u>
\@usetocstyle	<u>731</u>
A	
\aliastoc	<i>10</i> , <u>518</u>
C	
chapter (Option)	<u>61</u>
D	
\deactivatetocstyle	<i>8</i> , <u>604</u>
E	
\ext@toc	<u>816</u>
I	
\iftochaschapter	<u>61</u>
\iftochasdepth	<i>11</i> , <u>801</u>
L	
\l@addto@macro	<u>647</u>
\l@chapter	<u>264</u>
\l@figure	<u>264</u>
\l@paragraph	<u>264</u>
\l@part	<u>264</u>
\l@section	<u>264</u>
\l@subparagraph	<u>264</u>
\l@subsection	<u>264</u>
\l@subsubsection	<u>264</u>
\l@table	<u>264</u>
N	
\newtocstyle	<i>10</i> , <u>698</u>
nochapter (Option)	<u>61</u>
\numberline	<u>437</u>
O	
Optionen:	
chapter	<u>61</u>
nochapter	<u>61</u>
tocbreakscareless	<u>67</u>
tocbreaksstrict	<u>67</u>
tocflat	<u>73</u>
tocfullflat	<u>73</u>
tocgraduated	<u>73</u>
tocindentauto	<u>69</u>
tocindentmanual	<u>69</u>
toctextentriesindented ...	<u>71</u>
toctextentriesleft	<u>71</u>
R	
\reactivatetocstyle	<i>8</i> , <u>604</u>
S	
\selecttocstyleoption	<u>10</u>
\settocfeature	<i>8</i> , <u>609</u>
\settocstylefeature	<i>8</i> , <u>650</u>
\showtoc	<i>10</i> , <u>478</u>
T	
\tocbreakscareless	<u>6</u>
tocbreakscareless (Option) ...	<u>67</u>
\tocbreaksstrict	<u>6</u>
tocbreaksstrict (Option)	<u>67</u>
\tocflat	<u>4</u>
tocflat (Option)	<u>73</u>
\tocfullflat	<u>4</u>
tocfullflat (Option)	<u>73</u>
\tocgraduated	<u>4</u>
tocgraduated (Option)	<u>73</u>
\tocindentauto	<u>3</u>
tocindentauto (Option)	<u>69</u>
\tocindentmanual	<u>3</u>
tocindentmanual (Option)	<u>69</u>
\tocstyle@@numberline	<u>261</u>
\tocstyle@activate@all@1 ...	<u>459</u>
\tocstyle@activate@features	<u>655</u>
\tocstyle@copy@toc	<u>502</u>
\tocstyle@dottedtocline ...	<u>102</u>
\tocstyle@feature@<feature>@@	<u>655</u>
\tocstyle@feature@@<feature>@<TOC>@	<u>655</u>

<code>\tocstyle@feature@<feature>@<TOC>@<depth>@<numberline></code>	225
.	655
<code>\tocstyle@feature@<feature>@<depth>@<numberline>@<post>@starttoc</code>	521
.	655
<code>\tocstyle@feature@dothook</code>	783
<code>\tocstyle@feature@entryhook</code>	783
<code>\tocstyle@feature@entryvskip</code>	783
<code>\tocstyle@feature@leaders</code>	783
<code>\tocstyle@feature@pagenumberhook</code>	783
<code>\tocstyle@feature@parfillskip</code>	783
<code>\tocstyle@feature@raggedhook</code>	783
<code>\tocstyle@feature@spaceafternumber</code>	783
<code>\tocstyle@featurelist</code>	779
<code>\tocstyle@l@define</code>	459
<code>\tocstyle@macrochangewarning</code>	446
<code>\tocstyle@numberline</code>	225
<code>\tocstyle@post@starttoc</code>	521
<code>\tocstyle@pre@starttoc</code>	521
<code>\tocstyle@saved@starttoc</code>	94
<code>\tocstyle@saved@dottedtocline</code>	101
<code>\tocstyle@saved@numberline</code>	225
<code>\tocstyle@set@width</code>	578
<code>\tocstyleAliasTOC</code>	11, 601
<code>\tocstyledepth</code>	11, 603
<code>\tocstyleTOC</code>	11, 601
<code>\toctextentriesindented</code>	6
<code>toctextentriesindented</code> (Option)	71
<code>\toctextentriesleft</code>	6
<code>toctextentriesleft</code> (Option)	71
U	
<code>\usetocstyle</code>	6, 731

Change History

v0.1	General: start of new package	1
v0.2a	<code>\usetocstyle</code> : extended for deprecated TOC styles	29
v0.2d	<code>chapter</code> : New	13
	<code>\iftochaschapter</code> : New	13
	<code>\l@part</code> : part level is always -1	18
	<code>nochapter</code> : New	13
	<code>\tocstyle@dottedtocline</code> : fix: use of max-values	15
	<code>\tocstyle@set@width</code> : improve handling of standard classes	25
v0.2e	General: usage of new font elements of KOMA-Scriptat KOMAlite	31
v0.2f	<code>\l@figure</code> : recognize all subs of part, chapter, section and paragraph	18
v0.2g	<code>\l@figure</code> : warning for usage of not recommended package combinations	18
v0.2h	General: <code>\@ifnextchar</code> replaced by <code>\kernel@ifnextchar</code>	1
	<code>\l@figure</code> : lokal <code>\color</code> auf <code>\@gobble</code> setzen	18
	lokal <code>\normalcolor</code> auf <code>\relax</code> setzen	18
	<code>\tocstyle@set@width</code> : use after definition in aux-file	25
v0.2i	<code>\ext@toc</code> : use indirect extensions	31
v0.2j	<code>\@dottedtocline</code> : newer L ^A T _E X uses two <code>\kern</code>	21
	<code>\tocstyle@dottedtocline</code> : newer L ^A T _E X uses two <code>\kern</code>	15