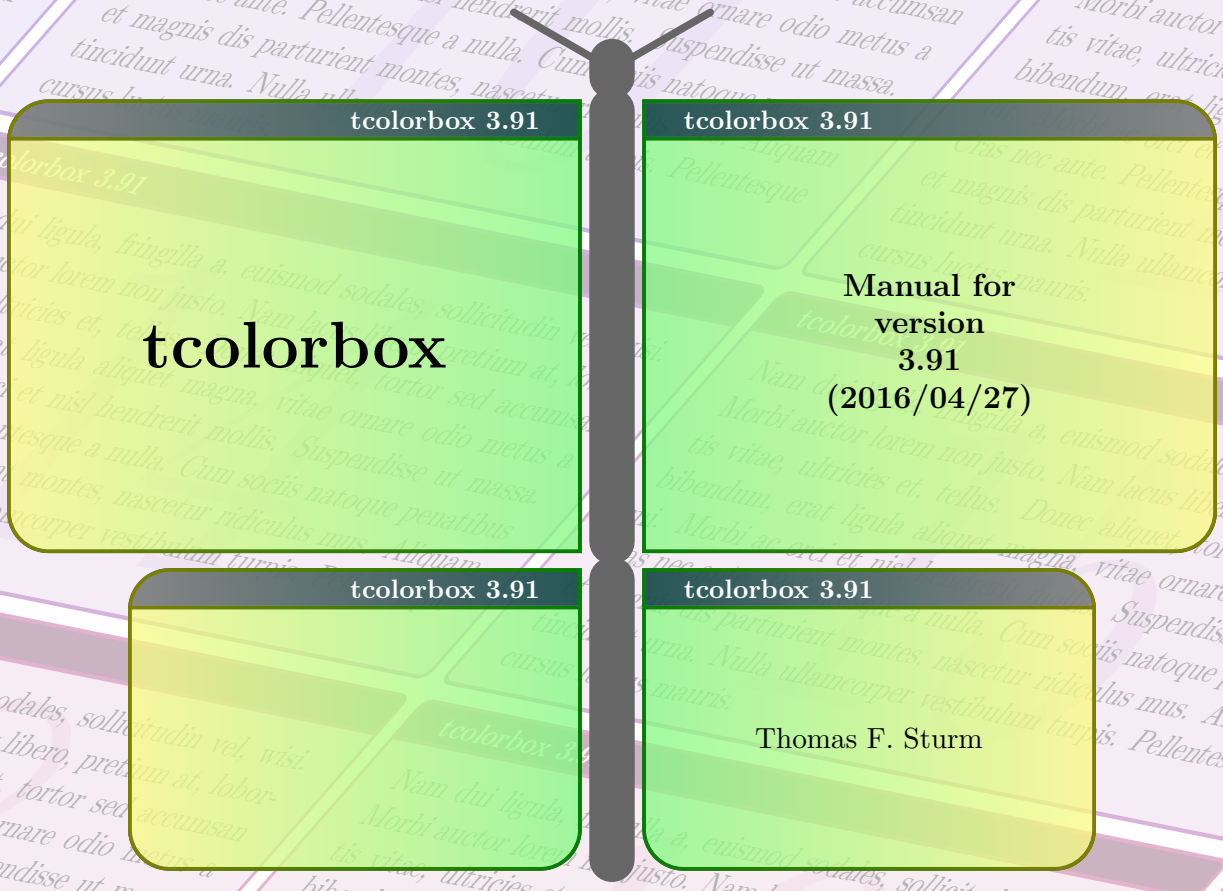

tcolorbox 3.91
tcolorbox
tcolorbox 3.91
Manual for
version
3.91
(2016/04/27)
tcolorbox 3.91
tcolorbox 3.91
Thomas F. Sturm

Cover code

```
% \usepackage{incgraph}
\begin{inctext}
\begin{tikzpicture}
\definecolorseries{boxcol}{rgb}{last}{blue}{red}
\resetcolorseries[28]{boxcol}
\coordinate (A) at (0,0); \coordinate (B) at (21,29.7);
\path[use as bounding box] (A) rectangle coordinate (C) (B);
\node[transform shape,xslant=0.7,rotate=-10,xshift=0cm] at (C) {%
  \begin{tcbraster}[raster columns=4,title=tcolorbox \version,
    fonttitle=\small\bfseries,raster width=50cm]
  \foreach \b in {1,...,28} {\begin{tcolorbox}[enhanced,
    watermark text=\thetcbrasternum,
    colframe=boxcol!30!white,
    colback=boxcol!25!white!30!white,
    colbacktitle=boxcol!!+!50!black!30!white,
    colupper=black!30!white]\lipsum[2]\end{tcolorbox}}
  \end{tcbraster}%
};
\node at (C) {%
  \begin{tcbitemize}[title=tcolorbox \version,fonttitle=\small\bfseries,
    enhanced jigsaw,opacityback=0.5,opacitybacktitle=0.75,
    halign=center,valign=center,arc=5mm,
    raster width=16cm,raster column skip=8mm,raster halign=center,
    raster force size=false,
    raster row 1/.style={height=6cm},
    raster row 2/.style={width=6cm,height=4cm},
    raster column 1/.style={flushright title,
      frame style={left color=yellow!50!black,right color=green!50!black},
      title style={left color=yellow!50!blue,right color=blue!50!green!50!black},
      interior style={left color=yellow!70,right color=green!70},
      underlay={\draw[line width=6mm,line cap=round,black!60]
        ([shift={(0.4,-0.15)}]frame.north east)
        --([shift={(0.4,0.15)}]frame.south east); }},
    raster column 2/.style={
      frame style={left color=green!50!black,right color=yellow!50!black},
      title style={left color=blue!50!green!50!black,right color=yellow!50!blue},
      interior style={left color=green!70,right color=yellow!70}}]
  \tcbitem[fontupper=\Huge\bfseries,sharp corners=east,
    underlay={\draw[line width=6mm,line cap=round,black!60]
      ([shift={(0.4,0.30)}]frame.north east)-- coordinate(A) +(0,0.2);
      \draw[line width=1mm,line cap=round,black!60](A) -- +(30:1.5cm);
      \draw[line width=1mm,line cap=round,black!60](A) -- +(150:1.5cm);}]
    tcolorbox
  \tcbitem[fontupper=\large\bfseries,sharp corners=west]
    Manual for\\ version\\ \version\\(\datum)
  \tcbitem[sharp corners=northeast]
  \tcbitem[sharp corners=northwest] Thomas F.~Sturm
  \end{tcbitemize}%
};
\end{tikzpicture}
\end{inctext}
```

The tcolorbox package

Manual for version 3.91 (2016/04/27)

Thomas F. Sturm[1]

Abstract

tcolorbox provides an environment for colored and framed text boxes with a heading line. Optionally, such a box can be split in an upper and a lower part. The package tcolorbox can be used for the setting of LaTeX examples where one part of the box displays the source code and the other part shows the output. Another common use case is the setting of theorems. The package supports saving and reuse of source code and text parts.

Contents

1 Introduction **8**
- 1.1 Installation . . . 8
- 1.2 Loading the Package . . . 8
- 1.3 Libraries . . . 9

2 Quick Reference **11**

3 Macros for Box Creation **12**

4 Option Keys **18**
- 4.1 Title . . . 18
- 4.2 Subtitle . . . 21
- 4.3 Upper Part . . . 22
- 4.4 Lower Part . . . 24
- 4.5 Colors and Fonts . . . 27
- 4.6 Text Alignment . . . 30
- 4.7 Geometry . . . 34
  - 4.7.1 Width . . . 34
  - 4.7.2 Rules . . . 35
  - 4.7.3 Arcs . . . 36
  - 4.7.4 Spacing . . . 38
  - 4.7.5 Size Shortcuts . . . 43
- 4.8 Corners . . . 47
- 4.9 Transparency . . . 50
- 4.10 Height Control . . . 52
- 4.11 Box Content Additions . . . 62
- 4.12 Overlays . . . 69
- 4.13 Floating Objects . . . 74
- 4.14 Embedding into the Surroundings . . . 76
- 4.15 Bounding Box . . . 80
- 4.16 Layered Boxes and Every Box Settings . . . 87

[1]Prof. Dr. Dr. Thomas F. Sturm, Institut für Mathematik und Informatik, Universität der Bundeswehr München, D-85577 Neubiberg, Germany; email: thomas.sturm@unibw.de

4.17 Capture Mode . . . 90
4.18 Text Characteristics . . . 91
4.19 Files . . . 92
4.20 \tcbox Specials . . . 92
4.21 Counters, Labels, and References . . . 94
4.22 Even and Odd Pages . . . 95
4.23 Externalization . . . 98
4.24 Miscellaneous . . . 98

5 Initialization Option Keys 100
5.1 Numbered Boxes . . . 100
5.2 Lists of tcolorboxes . . . 107

6 Side by Side 108
6.1 Basic Settings . . . 108
6.2 Advanced Settings from the LIB xparse Library . . . 114

7 Saving and Loading of Verbatim Texts 118

8 Recording 119
8.1 Makros . . . 119
8.2 Options . . . 119
8.3 Example: Exercises . . . 120
8.4 Example: Solutions . . . 123

9 Technical Overview and Customization 125
9.1 Skins and Drawing Engines . . . 125
9.2 Code Option Keys . . . 129
9.3 Subskins . . . 132
9.4 Drawing Scheme . . . 133
9.5 Useful Properties . . . 137

10 Library LIB skins 139
10.1 Style Option Keys . . . 139
10.2 Boxed Title Option Keys . . . 146
10.2.1 Boxed Title Placement . . . 146
10.2.2 Options for the Boxed Title Placement . . . 148
10.2.3 Options for the Boxed Title Box . . . 149
10.3 Watermark Option Keys . . . 156
10.4 Clip Environments . . . 163
10.5 Border Line Option Keys . . . 168
10.6 Shadow Option Keys . . . 173
10.6.1 Common Shadows and Halos . . . 173
10.6.2 Lifted Shadows . . . 178
10.6.3 Generic Shadows . . . 179
10.6.4 Ti*k*Z Shadows . . . 182
10.7 Ti*k*Z Picture Option Keys . . . 183
10.8 Underlay Option Keys . . . 186
10.9 Finish Option Keys . . . 188
10.10 Jigsaw Skin Variants . . . 190
10.11 Draft Mode . . . 192
10.12 Skin Family 'standard' . . . 193

Contents

10.13 Skin Family 'enhanced' . . . 195
10.14 Skin Family 'bicolor' . . . 208
10.15 Skin Family 'tile' . . . 213
10.16 Skin Family 'beamer' . . . 217
10.17 Skin Family 'widget' . . . 222
10.18 Skin Family 'empty' . . . 226
10.19 Skin 'spartan' . . . 236
10.20 Skin 'draft' . . . 237
10.21 Skin Family 'freelance' . . . 239

11 Inclusion of Boxed Image Files 240
11.1 Macros . . . 240
11.2 Option Keys . . . 243

12 TikZ Image and Picture Fill Extensions; Auxiliary Macros 244
12.1 Fill Plain . . . 244
12.2 Fill Stretch . . . 245
12.3 Fill Overzoom . . . 246
12.4 Fill Zoom . . . 247
12.5 Fill Shrink . . . 248
12.6 Fill Tile . . . 249
12.7 Filling Options . . . 250
12.8 Straightening of the Arcs . . . 251
12.9 Extracting Node Dimensions . . . 252

13 Library LIB vignette 253
13.1 Vignette Drawing . . . 253
13.2 Generic Geometry Settings . . . 254
13.3 Generic Color and Style Settings . . . 256
13.4 Generic Fading Settings . . . 258
13.5 Vignette as Underlay . . . 260
13.6 Vignette as Finish . . . 262

14 Library LIB raster 265
14.1 Concept of Rasters . . . 265
14.2 Macros of the Library . . . 267
14.3 Option Keys of the Library . . . 271
14.4 Adding Styles for Specific Boxes . . . 276
14.5 Combining Columns or Rows . . . 278
14.6 Rasters inside Rasters . . . 281
14.6.1 Raster Setup . . . 281
14.6.2 Placing Spaces . . . 282

15 Libraries LIB listings, LIB listingsutf8, and LIB minted 286
15.1 Loading the Libraries . . . 286
15.1.1 Loading LIB listings . . . 286
15.1.2 Loading LIB listingsutf8 . . . 286
15.1.3 Loading LIB minted . . . 287
15.2 Common Macros of the Libraries . . . 287
15.3 Option Keys of the LIB listings Library . . . 293
15.4 Option Keys of the LIB listingsutf8 Library . . . 295

15.5 Option Keys of the LIB minted Library . . . 296
15.6 Common Option Keys of all Libraries . . . 298
15.7 Option Keys for Processing and Full Document Examples . . . 307
15.8 Creation of LaTeX Tutorials . . . 313
15.9 Creation of LaTeX Exercises . . . 320
15.10 List of Exercises . . . 323
15.11 Solutions for the given LaTeX Exercises . . . 324

16 Library LIB theorems 326
16.1 Macros of the Library . . . 326
16.2 Option Keys of the Library . . . 330
16.3 Examples for Definitions and Theorems . . . 343
16.4 Using other theorem environments with `tcolorbox` . . . 348

17 Library LIB breakable 349
17.1 Technical Overview . . . 349
17.2 Limitations and Known Bugs . . . 350
17.3 Main Option Keys . . . 351
17.4 Option Keys for the Break Appearance . . . 355
17.5 Extra Options for Partial Boxes . . . 357
17.6 Breakable boxes and the `multicol` package . . . 359
17.7 Break Sequence for the Skins . . . 362
17.8 Break by Hand (Faked Break) . . . 371

18 Library LIB magazine 372
18.1 Creation and Resetting of Box Arrays . . . 372
18.2 Storing Content . . . 373
18.3 Retrieving Content . . . 375
18.4 Box Dimensions . . . 378

19 Library LIB fitting 380
19.1 Macros of the Library . . . 380
19.2 Option Keys of the Library . . . 382

20 Library LIB hooks 391
20.1 Concept of Hooks . . . 391
20.2 Box Content Additions . . . 392
20.3 Embedding into the Surroundings . . . 393
20.4 Overlays . . . 394
20.5 Watermarks . . . 396
20.6 Underlays . . . 398
20.7 Finishes . . . 399
20.8 Skin Code . . . 399
20.9 Extras . . . 401

21 Library LIB xparse 402
21.1 Option Keys . . . 402
21.2 Producing `tcolorbox` Environments and Commands . . . 404
21.3 Producing `tcbox` Commands . . . 407
21.4 Producing `tcblisting` Environments . . . 410
21.5 Producing `tcbinputlisting` Commands . . . 412
21.6 Producing `tboxfit` Commands . . . 413

Contents

22 Library LIB **external** **415**
22.1 Preparation of a Document for Externalization . . . 416
22.2 Marking Externalization Snippets . . . 417
22.3 Customization . . . 422
22.4 Troubleshooting and FAQ . . . 425

23 Library LIB **documentation** **426**
23.1 Macros of the Library . . . 426
23.2 Option Keys of the Library . . . 436
23.3 Predefined Colors of the Library . . . 443

A Picture Credits **444**

References **445**

Index **446**

1 Introduction

The package originates from the first edition of my book «*LaTeX – Einführung in das Textsatzsystem*» [19] in about 2006. For the LaTeX examples and tutorials given there, I wanted to have accentuated and colored boxes to display source code and compiled text in combination. Since, in my opinion, this type of boxes is also quite useful to highlight definitions and theorems, I applied them for my lecture notes in mathematics [16–18] as well. With this package, you are invited to apply these boxes for similar projects.

Starting with version 2.00, for all internal calculations ε-TeX [2] expressions are used in replacement of the package `calc`. The breaking news for version 2.00 is the support for breakable boxes. This new feature allows new applications of the package without affecting the core package too much if you do not need boxes to break automatically. With version 2.20, the often requested 'side by side' mode for listings has been added. With version 3.00, boxed titles are introduced together with improved customization options for overlays, underlays, finishes, and own code extensions.

Since the first public release in 2011, I received a lot of feedback from all over the world. I want to thank all who wrote me for supporting this package by sending bug reports and ideas for new or better features.

1.1 Installation

Typically, `tcolorbox` will be installed as part of a major LaTeX distribution and there is nothing special to do for a user.

If you intend to make a local installation *by hand*, see the `README` file of the `tcolorbox` package for some hints. The short story is: you have to install not only `tcolorbox.sty`, but also all `*.code.tex` files in the local `texmf` tree.

1.2 Loading the Package

The base package `tcolorbox` loads the packages `pgf` [20], `verbatim` [15], `etoolbox` [7], and `environ` [14]. `tcolorbox` itself is loaded in the usual manner in the preamble:

```
\usepackage{tcolorbox}
```

The package takes option keys in the key-value syntax. Alternatively, you may use these keys later in the preamble with `\tcbuselibrary` → P. 9 (see there). For example, the key to typeset listings is:

```
\usepackage[listings]{tcolorbox}
```

1.3 Libraries

The base package `tcolorbox` is extendable by program libraries. This is done by using option keys while loading the package or inside the preamble by applying the following macro with the same set of keys.

`\tcbuselibrary{`⟨*key list*⟩`}`

Loads the libraries given by the ⟨*key list*⟩.

```
\tcbuselibrary{listings,theorems}
```

The following keys are used inside `\tcbuselibrary` respectively `\usepackage` without the key tree path `/tcb/library/`.

`/tcb/library/skins` (LIB skins)

Loads the package `tikz` [20] and provides additional styles (skins) for the appearance of the colored boxes; see Section 10 from page 139.

`/tcb/library/vignette` (LIB vignette)

Provides code for more ornamental; see Section 13 from page 253.

`/tcb/library/raster` (LIB raster)

Provides additional macros and options for typesetting multiple boxes arranged in a kind of raster; see Section 14 from page 265.

`/tcb/library/listings` (LIB listings)

Loads the package `listings` [6] and provides additional macros for typesetting listings which are described in Section 15 from page 286.

`/tcb/library/listingsutf8` (LIB listingsutf8)

Loads the packages `listings` [6] and `listingsutf8` [10] for UTF-8 support. This is a variant of the library LIB listings and is described in Section 15 from page 286.

`/tcb/library/minted` (LIB minted)

Loads the package `minted` [11] to typeset listings with the `Pygments` [13] tool, also see Section 15 on page 286.

`/tcb/library/theorems` (LIB theorems)

Provides additional macros for typesetting theorems which are described in Section 16 from page 326.

`/tcb/library/breakable` (LIB breakable)

Provides support for automatic box breaking from one page to another; see Section 17 on page 349.

`/tcb/library/magazine` (LIB magazine)

Provides support for storing broken box parts to be used later or in interchanged order, Section 18 on page 372.

`/tcb/library/fitting` (LIB fitting)

Provides support for font size adaption of the box content to the box dimensions; see Section 19 from page 380.

`/tcb/library/hooks` (LIB hooks)

Extends several option keys to 'hookable' keys; see Section 20 from page 391.

`/tcb/library/xparse` (LIB xparse)

Provides document command production with `xparse` for `tcolorbox`; see Section 21 from page 402.

`/tcb/library/external` (LIB external)

Provides externalization support for stand-alone document snippets, see Section 22 on page 415.

`/tcb/library/documentation` (LIB documentation)

Provides additional macros for typesetting LaTeX documentations which are described in Section 23 from page 426.

`/tcb/library/many` (style, no value)

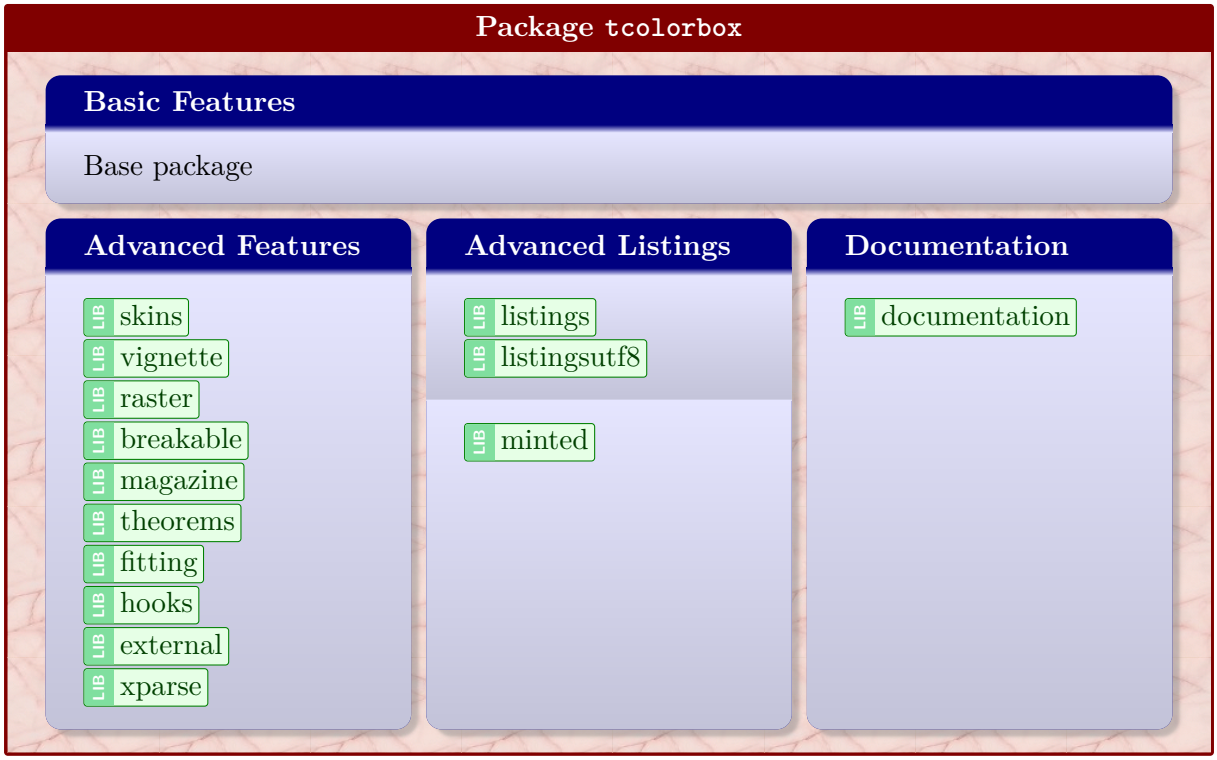
Loads the libraries LIB skins, LIB breakable, LIB raster, LIB hooks, LIB theorems, LIB fitting, and LIB xparse. Use this shortcut, if you want to use all features of `tcolorbox` with exception of typesetting listings and using the specialized LIB documentation library.

`/tcb/library/most` (style, no value)

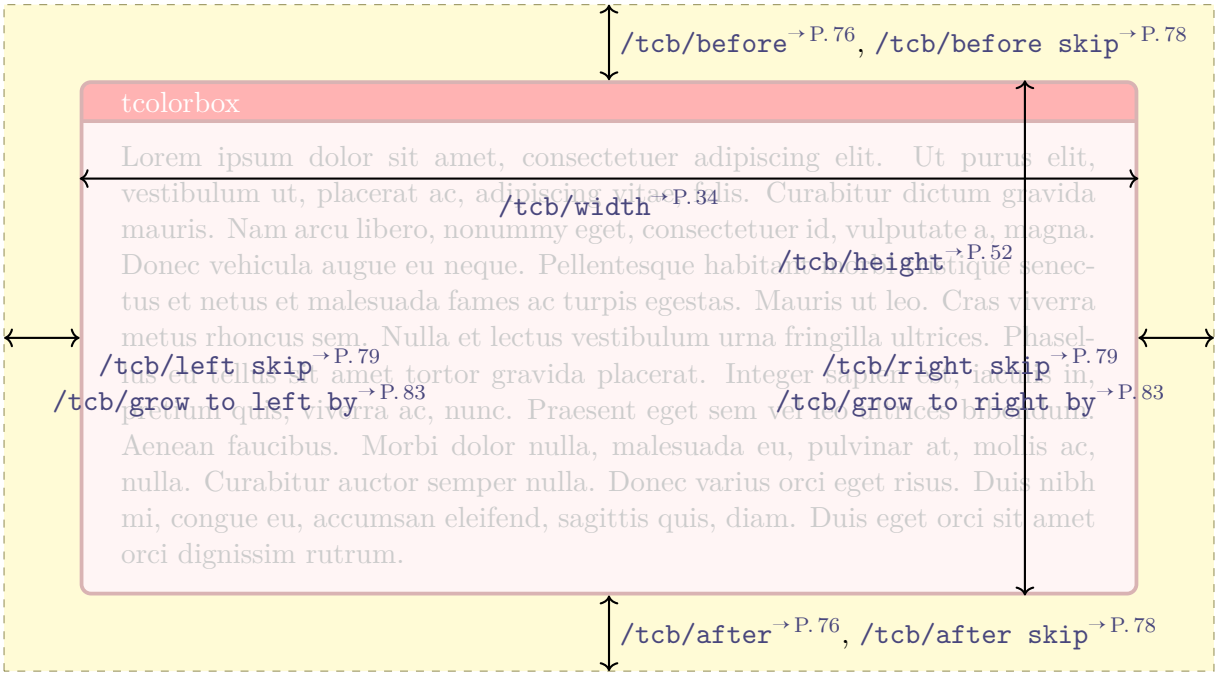
Loads all libraries except LIB minted and LIB documentation. Use this shortcut, if you want to use all features of `tcolorbox` with exception of using the `minted` package and using the specialized LIB documentation library.

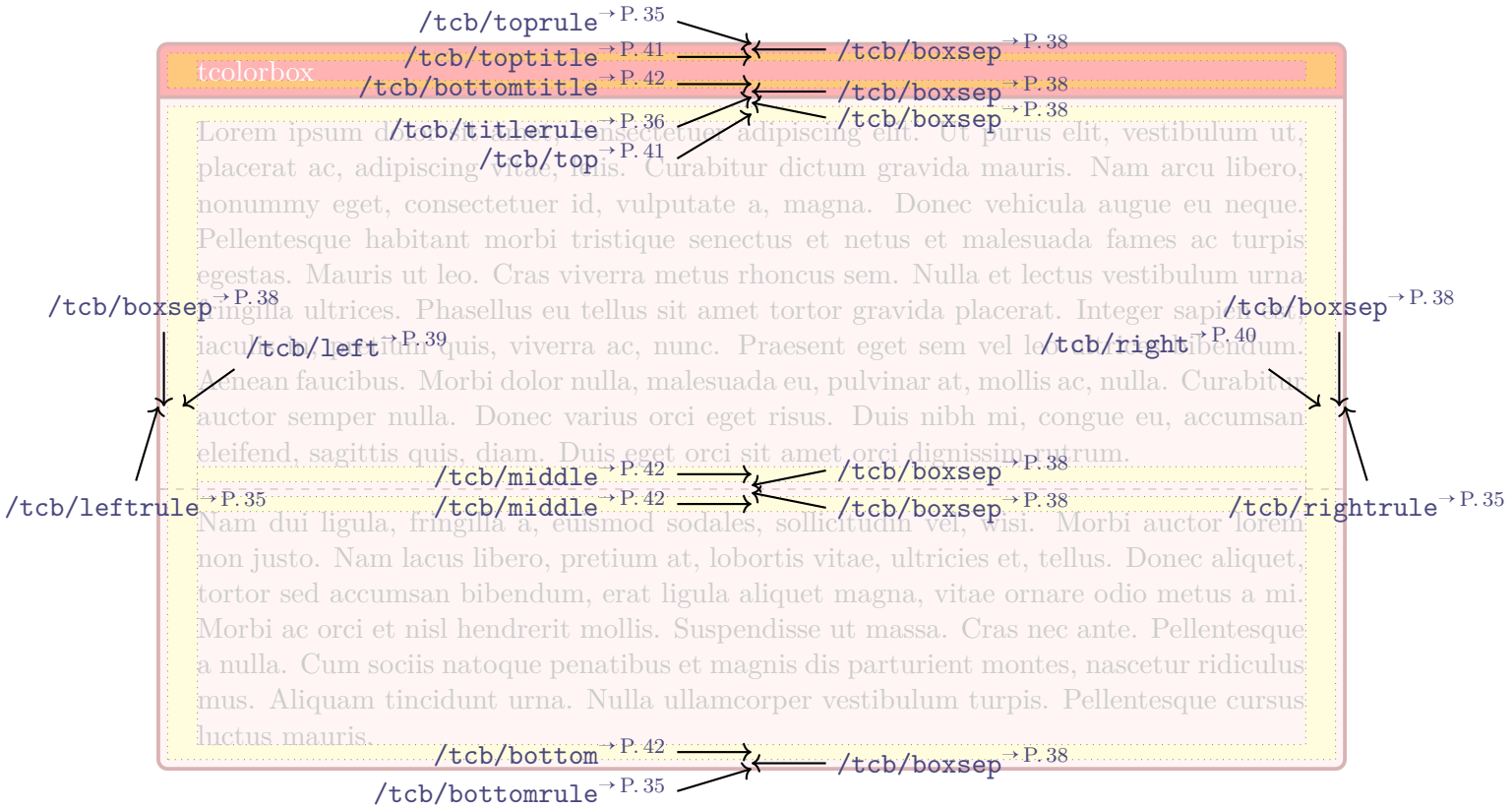
`/tcb/library/all` (style, no value)

Loads all libraries. Use this shortcut only, if you indend to use the LIB documentation library.



2 Quick Reference





3 Macros for Box Creation

\begin{tcolorbox}[⟨*options*⟩]
⟨*environment content*⟩
\end{tcolorbox}

This is the main environment to create an accentuated colored text box with rounded corners and, optionally, two parts. The appearance of this box is controlled by numerous options. In the most simple case the source code

```
\begin{tcolorbox}
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

creates the following compiled text box:

This is a **tcolorbox**.

The text content of the box can be divided in an upper and a lower part by the command `\tcblower`. Visually, both parts are separated by a line. For example:

```
\begin{tcolorbox}
This is another \textbf{tcolorbox}.
\tcblower
Here, you see the lower part of the box.
\end{tcolorbox}
```

This code gives the following box:

This is another **tcolorbox**.

Here, you see the lower part of the box.

The ⟨*options*⟩ control the appearance and several functions of the boxes, see Section 4 on page 18 for the complete list. A quick example is given here:

```
\begin{tcolorbox}[colback=red!5!white,colframe=red!75!black,title=My nice heading]
This is another \textbf{tcolorbox}.
\tcblower
Here, you see the lower part of the box.
\end{tcolorbox}
```

My nice heading

This is another **tcolorbox**.

Here, you see the lower part of the box.

\tcblower

Used inside `tcolorbox` to separate the upper box part from the optional lower box part. The upper and the lower part are treated as separate functional units. If you only want to draw a line, see `\tcbline` → P. 198.

`\tcbset{`⟨*options*⟩`}`

Sets options for every following `tcolorbox` → P. 12 inside the current TEX group. By default, this does not apply to nested boxes, see Section 4.16 on page 87.

For example, the colors of the boxes may be defined for the whole document by this:

```
\tcbset{colback=red!5!white,colframe=red!75!black}
```

`\tcbsetforeverylayer{`⟨*options*⟩`}`

Sets options for every following `tcolorbox` → P. 12 inside the current TEX group. In contrast to `\tcbset`, this does also apply to nested boxes, see Section 4.16 on page 87. Technically, the ⟨*options*⟩ are appended to the default values for every tcolorbox which are applied by `/tcb/reset` → P. 98.

You should not use this macro, if you are not completely sure that you want to have the ⟨*options*⟩ also for boxes in boxes (in boxes in boxes . . .).

```
\tcbset{colback=green!10!white}
\tcbsetforeverylayer{colframe=red!75!black}

\begin{tcolorbox}[title=All options for this box]
  This is a tcolorbox.\par\medskip
  \begin{tcolorbox}[title=Nested box]
    Note that this nested box has a red frame but no green background.
  \end{tcolorbox}
\end{tcolorbox}
\bigskip

\begin{tcolorbox}[reset]
  Options given with |\tcbsetforeverylayer| survive a |reset|.
\end{tcolorbox}
```

All options for this box

This is a tcolorbox.

Nested box

Note that this nested box has a red frame but no green background.

Options given with `\tcbsetforeverylayer` survive a `reset`.

`\tcbox[⟨options⟩]{⟨box content⟩}`

Creates a colored box which is fitted to the width of the given ⟨*box content*⟩. In principle, most ⟨*options*⟩ for a `tcolorbox` → P. 12 can be used for `\tcbox` with some restrictions. A `\tcbox` cannot have a lower part and cannot be broken.

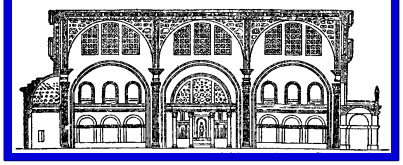
```
\tcbset{colframe=blue!50!black,colback=white,colupper=red!50!black,
        fonttitle=\bfseries,nobeforeafter,center title}

Text \tcbox[tcbox raise base]{Hello World}\hfill
%
\tcbox[left=0mm,right=0mm,top=0mm,bottom=0mm,boxsep=0mm,
  toptitle=0.5mm,bottomtitle=0.5mm,title=My table]{%
  \arrayrulecolor{blue!50!black}\renewcommand{\arraystretch}{1.2}%
  \begin{tabular}{r|c|l}
  One    & Two    & Three \\\hline\hline
  Men    & Mice   & Lions \\\hline
  Upper & Middle & Lower
  \end{tabular}}\hfill
%
\tcbox[colback=blue!85!black,
  left=0mm,right=0mm,top=0mm,bottom=0mm,boxsep=1mm,arc=0mm,boxrule=0.5pt,
  title=My picture]{%
  \includegraphics[width=5cm]{Basilica_5.png}}
```

Text Hello World

My table

One	Two	Three
Men	Mice	Lions
Upper	Middle	Lower

My picture



```
% \usepackage{tikz}
\tcbset{colframe=blue!50!black,colback=white,colupper=red!50!black,
        fonttitle=\bfseries,center title}

% Fixed width box
\begin{tcolorbox}Hello\\World!\end{tcolorbox}

% Fitted width box (like hbox or makebox)
\tcbox{Hello\\World!}

% Fitted width box (using a \tikzname\  node)
\tcbox[tikznode]{Hello\\World!}
```

Hello
World!

HelloWorld!

Hello
World!

See Section 21.2 on page 404 and Section 21.3 on page 407 for more elaborate methods to create new environments and commands.

`\newtcolorbox[⟨init options⟩]{⟨name⟩}[⟨number⟩][⟨default⟩]{⟨options⟩}`

Creates a new environment ⟨*name*⟩ based on `tcolorbox` → P. 12. Basically, `\newtcolorbox` operates like `\newenvironment`. This means, the new environment ⟨*name*⟩ optionally takes ⟨*number*⟩ arguments, where ⟨*default*⟩ is the default value for the optional first argument. The ⟨*options*⟩ are given to the underlying `tcolorbox`. Note that `/tcb/savedelimiter` → P. 26 is set to the given ⟨*name*⟩ automatically. The ⟨*init options*⟩ allow setting up automatic numbering, see Section 5 from page 100.

```
\newtcolorbox{mybox}{colback=red!5!white,
  colframe=red!75!black}

\begin{mybox}
This is my own box.
\end{mybox}
```

This is my own box.

```
\newtcolorbox{mybox}[1]{colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries,
  title=#1}

\begin{mybox}{Hello there}
This is my own box with a mandatory title.
\end{mybox}
```

Hello there

This is my own box with a mandatory title.

```
\newtcolorbox{mybox}[2][]{colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries,
  colbacktitle=red!85!black,enhanced,
attach boxed title to top center={yshift=-2mm},
  title=#2,#1}

\begin{mybox}[colback=yellow]{Hello there}
This is my own box with a mandatory title
and options.
\end{mybox}
```

Hello there

This is my own box with a mandatory title and options.

Definition in the preamble:

```
\newtcolorbox[auto counter,number within=section]{pabox}[2][]{%
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  title=Examp.~\thetcbcounter: #2,#1}
```

```
\begin{pabox}[colback=yellow]{Hello there}
This is my own box with a mandatory
numbered title and options.
\end{pabox}
```

Examp. 3.1: Hello there

This is my own box with a mandatory numbered title and options.

`\renewtcolorbox[⟨init options⟩]{⟨name⟩}[⟨number⟩][⟨default⟩]{⟨options⟩}`

Operates like `\newtcolorbox`, but based on `\renewenvironment` instead of `\newenvironment`. An existing environment is redefined.

`\newtcbox`[⟨*init options*⟩]{\⟨*name*⟩}[⟨*number*⟩][⟨*default*⟩]{⟨*options*⟩}

Creates a new macro \⟨*name*⟩ based on `\tcbox` → P. 14. Basically, `\newtcbox` operates like `\newcommand`. The new macro \⟨*name*⟩ optionally takes ⟨*number*⟩+1 arguments, where ⟨*default*⟩ is the default value for the optional first argument. The ⟨*options*⟩ are given to the underlying `tcbox`. The ⟨*init options*⟩ allow setting up automatic numbering, see Section 5 from page 100.

```
\newtcbox{\mybox}{colback=red!5!white,
  colframe=red!75!black}

\mybox{This is my own box.}
```

This is my own box.

```
\newtcbox{\mybox}[1]{colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries,
  title=#1}

\mybox{Hello there}{This is my own box.}
```

Hello there

This is my own box.

```
\newtcbox{\mybox}[2][]{colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries,
  title=#2,#1}

\mybox[colback=yellow]{Hello there}%
  {This is my own box.}
```

Hello there

This is my own box.

Definition in the preamble:

```
% counter from previous example
\newtcbox[use counter from=pabox]{\pbbox}[2][]{%
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  title=(\thetcbcounter) #2,#1}
```

```
\pbbox[colback=yellow]{Hello there}%
  {This is my own box.}
```

(3.2) Hello there

This is my own box.

```
\newtcbox{\mybox}[1][red]{on line,
  arc=0pt,outer arc=0pt,colback=#1!10!white,colframe=#1!50!black,
  boxsep=0pt,left=1pt,right=1pt,top=2pt,bottom=2pt,
  boxrule=0pt,bottomrule=1pt,toprule=1pt}
\newtcbox{\xmybox}[1][red]{on line,
  arc=7pt,colback=#1!10!white,colframe=#1!50!black,
  before upper={\rule[-3pt]{0pt}{10pt}},boxrule=1pt,
  boxsep=0pt,left=6pt,right=6pt,top=2pt,bottom=2pt}

The \mybox[green]{quick} brown \mybox{fox} \mybox[blue]{jumps} over the
\mybox[green]{lazy} \mybox{dog}.\par
The \xmybox[green]{quick} brown \xmybox{fox} \xmybox[blue]{jumps} over the
\xmybox[green]{lazy} \xmybox{dog}.
```

The quick brown fox jumps over the lazy dog.
The quick brown fox jumps over the lazy dog.

`\renewtcbox`[⟨*init options*⟩]{\⟨*name*⟩}[⟨*number*⟩][⟨*default*⟩]{⟨*options*⟩}

Operates like `\newtcbox`, but based on `\renewcommand` instead of `\newcommand`. An existing macro is redefined.

N 2014-10-20 `\tcolorboxenvironment{⟨name⟩}{⟨options⟩}`

An existing environment ⟨*name*⟩ is redefined to be boxed inside a `tcolorbox` with the given ⟨*options*⟩.

```
% tcbuselibrary{skins}
\newenvironment{myitemize}{%
  \begin{itemize}}{\end{itemize}}

\tcolorboxenvironment{myitemize}{blanker,
  before skip=6pt,after skip=6pt,
  borderline west={3mm}{0pt}{red}}

Some text.
\begin{myitemize}
\item Alpha
\item Beta
\item Gamma
\end{myitemize}
More text.
```

Some text.

- Alpha
- Beta
- Gamma

More text.

See further examples in Section 16.4 on page 348.

4 Option Keys

For the ⟨*options*⟩ in `tcolorbox` → P. 12 respectively `\tcbset` → P. 13 the following `pgf` keys can be applied. The key tree path `/tcb/` is not to be used inside these macros. It is easy to add your own style keys using the syntax for `pgf` keys, see [19, 20] or the examples starting from page 313.

4.1 Title

`/tcb/title`=⟨*text*⟩ (no default, initially empty)

Creates a heading line with ⟨*text*⟩ as content.

```
\begin{tcolorbox}[title=My heading line]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My heading line

This is a **tcolorbox**.

`/tcb/notitle` (no value, initially set)

Removes the title line if set before.

`/tcb/adjusted title`=⟨*text*⟩ (style, no default, initially unset)

Creates a heading line with ⟨*text*⟩ as content. The minimal height of this line is adjusted to fit the text given by `/tcb/adjust text`. This option makes sense for single line headings if boxes are set side by side with equal height. Note that it is very easy to trick this adjustment.

```
\tcbset{colback=White,arc=0mm,width=(\linewidth-4pt)/4,
equal height group=AT,before=,after=\hfill,fonttitle=\bfseries}

The following titles are not adjusted:\\
\foreach \n in {xxx,ggg,AAA,\"Agypten}
{\begin{tcolorbox}[title=\n,colframe=red!75!black]
  Some content.\end{tcolorbox}}
Now, we try again with adjusted titles:\\
\foreach \n in {xxx,ggg,AAA,\"Agypten}
{\begin{tcolorbox}[adjusted title=\n,colframe=blue!75!black]
  Some content.\end{tcolorbox}}
```

The following titles are not adjusted:

xxx	ggg	AAA	Ägypten
Some content.	Some content.	Some content.	Some content.

Now, we try again with adjusted titles:

xxx	ggg	AAA	Ägypten
Some content.	Some content.	Some content.	Some content.

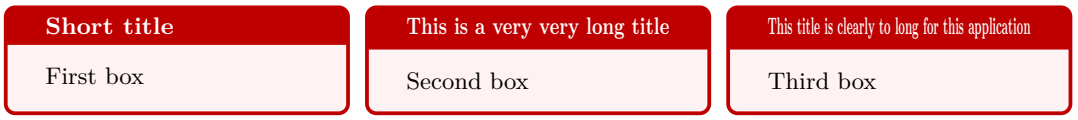
`/tcb/adjust text`=⟨*text*⟩ (no default, initially `Äpgjy`)

This sets the reference text for `/tcb/adjusted title`. If your texts never exceed 'Äpgjy' in depth and height you don't need to care about this option.

N 2014-11-24 **/tcb/squeezed title=**⟨*text*⟩ (style, no default, initially unset)

Creates a single heading line with ⟨*text*⟩ as content. If the ⟨*text*⟩ is longer than the available space, the text is squeezed to fit into the available space.

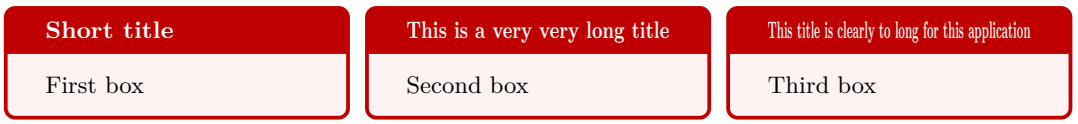
```
% \tcbuselibrary{raster}
\begin{tcbitemize}[raster columns=3,raster equal height,
  colframe=red!75!black,colback=red!5!white,fonttitle=\bfseries]
\tcbitem[squeezed title={Short title}]
  First box
\tcbitem[squeezed title={This is a very very long title}]
  Second box
\tcbitem[squeezed title={This title is clearly to long for this application}]
  Third box
\end{tcbitemize}
```



N 2014-11-24 **/tcb/squeezed title*=**⟨*text*⟩ (style, no default, initially unset)

This is a combination of /tcb/adjusted title → P. 18 and /tcb/squeezed title.

```
% \tcbuselibrary{raster}
\begin{tcbitemize}[raster columns=3,raster equal height,
  colframe=red!75!black,colback=red!5!white,fonttitle=\bfseries]
\tcbitem[squeezed title*={Short title}]
  First box
\tcbitem[squeezed title*={This is a very very long title}]
  Second box
\tcbitem[squeezed title*={This title is clearly to long for this application}]
  Third box
\end{tcbitemize}
```



/tcb/detach title (no value)

Detaches the title from its normal position. The text of the title is stored into **\tcbtitletext** and the formatted title is available by **\tcbtitle**. The main application is to move the title from its usual place to another one.

```
\newtcolorbox{mybox}[2][]{colbacktitle=red!10!white,
  colback=blue!10!white,coltitle=red!70!black,
  title={#2},fonttitle=\bfseries,#1}

\begin{mybox}{My title}
  This is a \textbf{tcolorbox}.
\end{mybox}
\begin{mybox}[detach title,before upper={\tcbtitle\quad}]{My title}
  This is a \textbf{tcolorbox}.
\end{mybox}
\begin{mybox}[detach title,after upper={\par\hfill\tcbtitle}]{My title}
  This is a \textbf{tcolorbox}.
\end{mybox}
```

My title

This is a **tcolorbox**.

My title This is a **tcolorbox**.

This is a **tcolorbox**.

My title

/tcb/attach title (no value)

Attaches the title to its normal position. This option is used to reverse /tcb/detach title.

U 2015-07-08 **/tcb/attach title to upper=**⟨*text*⟩ (style, default empty, initially unset)

Attaches the title to the begin of the upper part of the box content. The optional ⟨*text*⟩ is set between the formatted title and the box content.

```
\newtcolorbox{mybox}[2][]{colbacktitle=red!10!white,
  colback=blue!10!white,coltitle=red!70!black,
  title={#2},fonttitle=\bfseries,#1}

\begin{mybox}[attach title to upper={\ ---\ }]{My title}
  This is a \textbf{tcolorbox}.
\end{mybox}
\begin{mybox}[attach title to upper,after title={:\ }]{My title}
  This is a \textbf{tcolorbox}.
\end{mybox}
```

My title — This is a **tcolorbox**.

My title: This is a **tcolorbox**.

! More title options are documented in Section 4.11 on page 62 and Section 10.2 on page 146.

4.2 Subtitle

Inside the box content, one or more subtitles can be added. In general, a subtitle is a further tcolorbox → P. 12 which inherits some color and geometry options from the enclosing box. It may be customized just like any other tcolorbox → P. 12.

N 2014-10-10 **\tcbsubtitle**[⟨*options*⟩]{⟨*text*⟩}

Used inside a tcolorbox → P. 12 to add a subtitle box with the given ⟨*text*⟩. This is an independent tcolorbox → P. 12 which is formatted by several inherited properties of the enclosing box, by further settings from /tcb/subtitle style, and by the given ⟨*options*⟩.

```
\begin{tcolorbox}[title=My title,
    colback=red!5!white,
    colframe=red!75!black,
    fonttitle=\bfseries]
  This is a \textbf{tcolorbox}.
\tcbsubtitle[before skip=\baselineskip]%
  {My subtitle}
  Further text.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

My subtitle

Further text.

```
\begin{tcolorbox}[title=My title,
    colback=red!5!white,
    colframe=red!75!black,
    colbacktitle=yellow!50!red,
    coltitle=red!25!black,
    fonttitle=\bfseries]
  This is a \textbf{tcolorbox}.
\tcbsubtitle[before skip=\baselineskip]%
  {My subtitle}
  Further text.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

My subtitle

Further text.

N 2014-10-10 **/tcb/subtitle style**=⟨*options*⟩ (no default, initially empty)

Adds tcolorbox ⟨*options*⟩ to the settings for \tcbsubtitle.

```
\begin{tcolorbox}[title=My title,
  colback=red!5!white,
  colframe=red!75!black,
  colbacktitle=yellow!50!red,
  coltitle=red!25!black,
  fonttitle=\bfseries,
  subtitle style={boxrule=0.4pt,
    colback=yellow!50!red!25!white} ]
  This is a \textbf{tcolorbox}.
\tcbsubtitle{My subtitle}
  Further text.
\tcbsubtitle{Second subtitle}
  Further text.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

My subtitle

Further text.

Second subtitle

Further text.

4.3 Upper Part

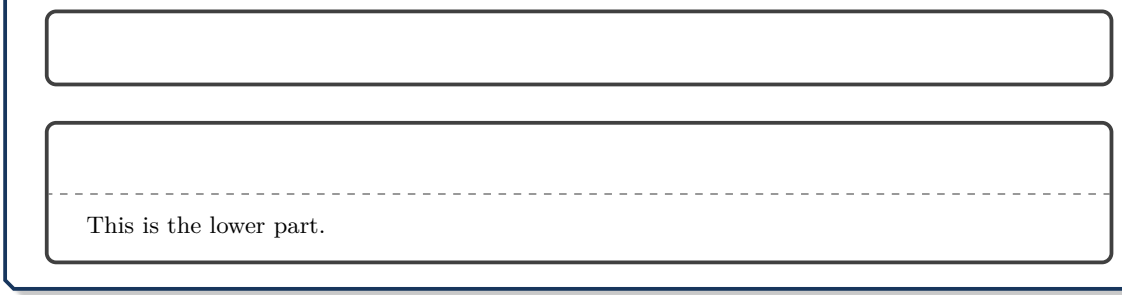
The text content of a `tcolorbox` → P. 12 may be parted into a mandatory *upper part* and an optional *lower part*. These parts are separated by `\tcblower` → P. 12. If there is no `\tcblower` → P. 12 present, there is no *lower part* and the *upper part* forms the complete text content.

N 2015-01-06 **/tcb/upperbox=⟨*mode*⟩** (no default, initially `visible`)

Controls the treatment of the upper part of the box. If there is no lower part, this is the complete text content. Feasible values for ⟨*mode*⟩ are:

- `visible`: usual type setting of the upper part,
- `invisible`: empty space instead of the upper part contents.

```
\begin{tcolorbox}[upperbox=invisible,colback=white]
This is a \textbf{tcolorbox} (but invisible).
\end{tcolorbox}

\bigskip

\begin{tcolorbox}[upperbox=invisible,colback=white]
This is a \textbf{tcolorbox} (but invisible).
\tcblower
This is the lower part.
\end{tcolorbox}
```



N 2015-01-06 **/tcb/visible** (style, no value)

Shortcut for setting `/tcb/upperbox` and `/tcb/lowerbox` → P. 24 to be `visible`.

N 2015-01-06 **/tcb/invisible** (style, no value)

Shortcut for setting `/tcb/upperbox` and `/tcb/lowerbox` → P. 24 to be `invisible`.

```
\begin{tcolorbox}[invisible]
This is a \textbf{tcolorbox} (but invisible).
\end{tcolorbox}
```

N 2015-05-04 **/tcb/saveto=⟨*file name*⟩** (no default, initially empty)

Saves the content of the box into a file for an optional later usage. This is the counterpart of /tcb/savelowerto → P. 24, but is saves not only the upper part but the whole content. If a lower part is present, it is also saved including \tcblower → P. 12.

! This option cannot be combined with /tcb/savelowerto → P. 24.

```
\begin{tcolorbox}[invisible,saveto=\jobname_mysave1.tex,colback=white]
This is a \textbf{tcolorbox} which seems to be empty.
The content is saved for later usage.
\end{tcolorbox}

Now, we load the saved text:\\
\input{\jobname_mysave1.tex}
```

Now, we load the saved text:
This is a **tcolorbox** which seems to be empty. The content is saved for later usage.

```
\begin{tcolorbox}[saveto=\jobname_mysave2.tex]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}

Now, we load the saved text:
\begin{tcolorbox}[colframe=red,colback=red!10,
  coltitle=black,colbacktitle=red!20,sidebyside,
  title=Here we see the saved content including the lower part]
\input{\jobname_mysave2.tex}
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

Now, we load the saved text:

Here we see the saved content including the lower part

This is a **tcolorbox**. | This is the lower part.

4.4 Lower Part

`/tcb/lowerbox`=⟨*mode*⟩ (no default, initially `visible`)

Controls the treatment of the lower part of the box. Feasible values for ⟨*mode*⟩ are:

- `visible`: usual type setting of the lower part,
- `invisible`: empty space instead of the lower part contents,
- `ignored`: the lower part is not used (here).

The last two values are usually applied in connection with `savelowerto`.

```
\begin{tcolorbox}[lowerbox=invisible,colback=white]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part (but invisible).
\end{tcolorbox}

\begin{tcolorbox}[lowerbox=ignored,colback=white]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part (but ignored).
\end{tcolorbox}
```

This is a **tcolorbox**.

This is a **tcolorbox**.

U 2014-11-28 `/tcb/savelowerto`=⟨*file name*⟩ (no default, initially empty)

Saves the content of the lower part into a file for an optional later usage.

```
\begin{tcolorbox}[lowerbox=invisible,savelowerto=\jobname_bspsave.tex,colback=white]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part which may be quite complex:
$\displaystyle f(x)=\frac{1+x^2}{1-x^2}$.
\end{tcolorbox}

Now, we load the saved text:\\
\input{\jobname_bspsave.tex}
```

This is a **tcolorbox**.

Now, we load the saved text:

This is the lower part which may be quite complex: $f(x) = \dfrac{1+x^2}{1-x^2}$.

`/tcb/lower separated=true|false` (default `true`, initially `true`)

If set to `true`, the lower part is visually separated from the upper part. It depends on the chosen skin how the visualization of the separation is done.

```
% \tcbuselibrary{skins,raster}
\begin{tcbraster}[colback=red!5!white,colframe=red!75!black,
  fonttitle=\bfseries,fontlower=\itshape]
%
\begin{tcolorbox}[title=Lower separated]
This is the upper part.
\tcblower
This is the lower part.
\end{tcolorbox}
%
\begin{tcolorbox}[title=Lower not separated,lower separated=false]
This is the upper part.
\tcblower
This is the lower part.
\end{tcolorbox}
%
\begin{tcolorbox}[sidebyside,title=Lower separated]
This is the upper part.
\tcblower
This is the lower part.
\end{tcolorbox}
%
\begin{tcolorbox}[sidebyside,title=Lower not separated,lower separated=false]
This is the upper part.
\tcblower
This is the lower part.
\end{tcolorbox}
%
\begin{tcolorbox}[beamer,title=Lower separated]
This is the upper part.
\tcblower
This is the lower part.
\end{tcolorbox}
%
\begin{tcolorbox}[beamer,title=Lower not separated,lower separated=false]
This is the upper part.
\tcblower
This is the lower part.
\end{tcolorbox}
%
\end{tcbraster}
```

Lower separated

This is the upper part.

This is the lower part.

Lower not separated

This is the upper part.

This is the lower part.

Lower separated

This is the upper part.

This is the lower part.

Lower not separated

This is the upper part.

This is the lower part.

Lower separated

This is the upper part.

This is the lower part.

Lower not separated

This is the upper part.

This is the lower part.

`/tcb/savedelimiter=`⟨*name*⟩ (no default, initially `tcolorbox`)

Used in connection with new environment definitions which extend `tcolorbox` and use or allow the option `savelowerto`. To catch the end of the new box environment ⟨*name*⟩ has to be the name of this environment. Additionally, the environment definition has to use `\tcolorbox` instead of `\begin{tcolorbox}` and `\endtcolorbox` instead of `\end{tcolorbox}`.

```
\newenvironment{mybox}[1]{%
  \tcolorbox[savedelimiter=mybox,
             savelowerto=\jobname_bspsave2.tex,lowerbox=ignored,
             colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
             title=#1]}%
  {\endtcolorbox}

\begin{mybox}{My Example}
Upper part.
\tcblower
Saved lower part!
\end{mybox}

Now, the saved part is used:
\begin{tcolorbox}[colback=green!5]
\input{\jobname_bspsave2.tex}
\end{tcolorbox}
```

My Example

Upper part.

Now, the saved part is used:

Saved lower part!

The `savedelimiter` is used implicitely with `\newtcolorbox` → P. 15 which allows a more convenient usage:

```
\newtcolorbox{mybox}[1]{%
             savelowerto=\jobname_bspsave2.tex,lowerbox=ignored,
             colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
             title=#1}%

\begin{mybox}{My Example}
Upper part.
\tcblower
Saved lower part!
\end{mybox}

Now, the saved part is used:
\begin{tcolorbox}[colback=green!5]
\input{\jobname_bspsave2.tex}
\end{tcolorbox}
```

My Example

Upper part.

Now, the saved part is used:

Saved lower part!

4.5 Colors and Fonts

`/tcb/colframe=`⟨*color*⟩ (no default, initially `black!75!white`)

Sets the frame ⟨*color*⟩ of the box.

```
\begin{tcolorbox}[colframe=red!50!white]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/colback=`⟨*color*⟩ (no default, initially `black!5!white`)

Sets the background ⟨*color*⟩ of the box.

```
\begin{tcolorbox}[colback=red!50!white]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/title filled=true|false` (default `true`, initially `false`)

Switches the drawing of the title background according to the given value. This option is set to `true` automatically by `/tcb/colbacktitle`, `/tcb/opacitybacktitle` → P. 50, and `/tcb/title style` → P. 142, and `/tcb/title code` → P. 131.

```
\begin{tcolorbox}[title=My title,title filled]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[title=My title,
  title filled=false]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

My title

This is a **tcolorbox**.

`/tcb/colbacktitle=`⟨*color*⟩ (no default, initially `black!50!white`)

Sets the background ⟨*color*⟩ of the title area of the box.

```
\begin{tcolorbox}[colbacktitle=red!50!white,
  title=My title,coltitle=black,
  fonttitle=\bfseries]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

`/tcb/colupper`=⟨*color*⟩ (no default, initially `black`)

Sets the text ⟨*color*⟩ of the upper part.

```
\begin{tcolorbox}[colupper=red!75!black]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

`/tcb/collower`=⟨*color*⟩ (no default, initially `black`)

Sets the text ⟨*color*⟩ of the lower part.

```
\begin{tcolorbox}[collower=red!75!black]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

`/tcb/coltext`=⟨*color*⟩ (style, no default, initially `black`)

Sets the text ⟨*color*⟩ of the box. This is an abbreviation for setting `colupper` and `collower` to the same value.

```
\begin{tcolorbox}[coltext=red!75!black]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

`/tcb/coltitle`=⟨*color*⟩ (no default, initially `white`)

Sets the title text ⟨*color*⟩ of the box.

```
\begin{tcolorbox}[coltitle=red!75!black,
  colbacktitle=black!10!white,title=Test]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

Test

This is a **tcolorbox**.

/tcb/fontupper=⟨*text*⟩ (no default, initially empty)

Sets ⟨*text*⟩ before the content of the upper part (e. g. font settings).

```
\begin{tcolorbox}[fontupper=Hello!~\sffamily]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

Hello! This is a **tcolorbox**.

/tcb/fontlower=⟨*text*⟩ (no default, initially empty)

Sets ⟨*text*⟩ before the content of the lower part (e. g. font settings).

```
\begin{tcolorbox}[fontlower=\sffamily\bfseries]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

/tcb/fonttitle=⟨*text*⟩ (no default, initially empty)

Sets ⟨*text*⟩ before the content of the title text (e. g. font settings).

```
\begin{tcolorbox}[fonttitle=\sffamily\bfseries\large,title=Hello]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

Hello

This is a **tcolorbox**.

More color options are provided by using skins documented in Section 10 from page 139.

4.6 Text Alignment

N 2015-05-07 **/tcb/halign=**⟨*alignment*⟩ (no default, initially **justify**)

If there is no lower part, **halign** determines the horizontal ⟨*alignment*⟩ of the text content. Otherwise, **halign** determines the horizontal ⟨*alignment*⟩ of the upper part of the box only. The feasible values for ⟨*alignment*⟩ are more or less identical to the corresponding **/tikz/align** settings, even if the implementation differs.

- **justify**: usual left and right justified type setting.
- **left**: left border justification in analogy to plain TEX.
- **flush left**: left border justification with **\raggedright** of LATEX.
- **right**: right border justification in analogy to plain TEX.
- **flush right**: right border justification with **\raggedleft** of LATEX.
- **center**: centering in analogy to plain TEX.
- **flush center**: centering with **\centering** of LATEX.

The differences between the flush and non-flush version are explained in detail in the TikZ manual [20]. The short story is that the non-flush versions will often look more balanced but with more hyphenations.

```
\tcbset{colback=red!5!white,colframe=red!75!black,size=small,
  fonttitle=\bfseries,width=3.5cm,box align=top,
  nobeforeafter}

\begin{tcolorbox}[adjusted title=flush center,halign=flush center]
This is a demonstration text for showing how line breaking works.
\end{tcolorbox}
\begin{tcolorbox}[adjusted title=flush left,halign=flush left]
This is a demonstration text for showing how line breaking works.
\end{tcolorbox}
\begin{tcolorbox}[adjusted title=flush right,halign=flush right]
This is a demonstration text for showing how line breaking works.
\end{tcolorbox}

\begin{tcolorbox}[adjusted title=center,halign=center]
This is a demonstration text for showing how line breaking works.
\end{tcolorbox}
\begin{tcolorbox}[adjusted title=left,halign=left]
This is a demonstration text for showing how line breaking works.
\end{tcolorbox}
\begin{tcolorbox}[adjusted title=right,halign upper=right]
This is a demonstration text for showing how line breaking works.
\end{tcolorbox}
```

flush center
This is a demonstration text for showing how line breaking works.

flush left
This is a demonstration text for showing how line breaking works.

flush right
This is a demonstration text for showing how line breaking works.

center
This is a demonstration text for showing how line breaking works.

left
This is a demonstration text for showing how line breaking works.

right
This is a demonstration text for showing how line breaking works.

N 2015-05-07 **/tcb/halign upper=**⟨*alignment*⟩ (no default, initially **justify**)

Alias for /tcb/halign.

N 2015-05-07 **/tcb/halign lower=⟨*alignment*⟩** (no default, initially **justify**)

halign lower determines the horizontal ⟨*alignment*⟩ of the lower part of the box. The feasible values for ⟨*alignment*⟩ are the same as for **/tcb/halign** → P. 30.

```
\begin{tcbraster}[raster columns=3,fonttitle=\bfseries,
  colback=red!5!white,colframe=red!75!black]

  \begin{tcolorbox}[adjusted title=flush center,halign lower=flush center]
    Upper part. \tcblower Lower part.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=flush left,halign lower=flush left]
    Upper part. \tcblower Lower part.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=flush right,halign lower=flush right]
    Upper part. \tcblower Lower part.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=center,halign lower=center]
    Upper part. \tcblower Lower part.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=left,halign lower=left]
    Upper part. \tcblower Lower part.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=right,halign lower=right]
    Upper part. \tcblower Lower part.
  \end{tcolorbox}

\end{tcbraster}
```

flush center
Upper part.
Lower part.

flush left
Upper part.
Lower part.

flush right
Upper part.
Lower part.

center
Upper part.
Lower part.

left
Upper part.
Lower part.

right
Upper part.
Lower part.

N 2015-05-07 **/tcb/halign title=⟨*alignment*⟩** (no default, initially **justify**)

halign lower determines the horizontal ⟨*alignment*⟩ of the title of the box. The feasible values for ⟨*alignment*⟩ are the same as for /tcb/halign → P. 30.

```
\begin{tcbraster}[raster columns=3,fonttitle=\bfseries,
  colback=red!5!white,colframe=red!75!black]

  \begin{tcolorbox}[adjusted title=flush center,halign title=flush center]
    This is a \textbf{tcolorbox}.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=flush left,halign title=flush left]
    This is a \textbf{tcolorbox}.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=flush right,halign title=flush right]
    This is a \textbf{tcolorbox}.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=center,halign title=center]
    This is a \textbf{tcolorbox}.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=left,halign title=left]
    This is a \textbf{tcolorbox}.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=right,halign title=right]
    This is a \textbf{tcolorbox}.
  \end{tcolorbox}

\end{tcbraster}
```

flush center
This is a **tcolorbox**.

flush left
This is a **tcolorbox**.

flush right
This is a **tcolorbox**.

center
This is a **tcolorbox**.

left
This is a **tcolorbox**.

right
This is a **tcolorbox**.

U 2015-05-07 **/tcb/flushleft upper** (style, no value)

Shortcut for setting /tcb/halign → P. 30 to **flush left**.

U 2015-05-07 **/tcb/center upper** (style, no value)

Shortcut for setting /tcb/halign → P. 30 to **flush center**.

U 2015-05-07 **/tcb/flushright upper** (style, no value)

Shortcut for setting /tcb/halign → P. 30 to **flush right**.

U 2015-05-07 **/tcb/flushleft lower** (style, no value)

Shortcut for setting /tcb/halign lower → P. 31 to **flush left**.

U 2015-05-07 **/tcb/center lower** (style, no value)

Shortcut for setting /tcb/halign lower → P. 31 to **flush center**.

U 2015-05-07 **/tcb/flushright lower** (style, no value)

Shortcut for setting /tcb/halign lower → P. 31 to **flush right**.

U 2015-05-07 **/tcb/flushleft title** (style, no value)

Shortcut for setting /tcb/halign title to **flush left**.

U 2015-05-07 **/tcb/center title** (style, no value)

Shortcut for setting /tcb/halign title to **flush center**.

U 2015-05-07 **/tcb/flushright title** (style, no value)

Shortcut for setting /tcb/halign title to **flush right**.

! The vertical alignment settings are only relevant for boxes which are larger than their natural height, see Section 4.10 on page 52.

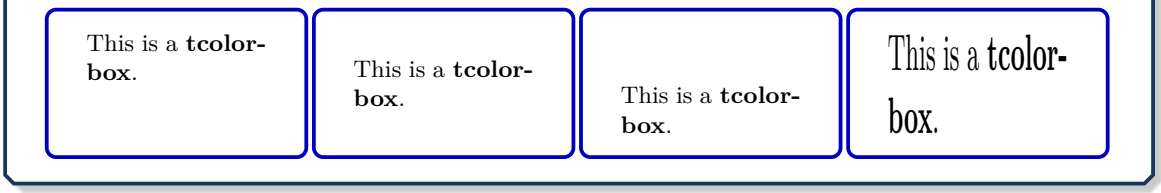
U 2015-07-16 **/tcb/valign=⟨*alignment*⟩** (no default, initially `top`)

If the height of a `tcolorbox` is not the natural height, `valign` determines the vertical ⟨*alignment*⟩ of the upper part. Feasible values are

- `top`: Anchor text at top.
- `center`: Anchor text at center.
- `bottom`: Anchor text at bottom.
- `scale`: Scale text vertically to fit into the available space. This is brutal and may not look very good. Consider Section 19 on page 380 alternatively.
- `scale*`: Like `scale`, but scaling is bounded by `/tcb/valign scale limit`.

For a box with natural height, these settings are meaningless.

```
\tcbset{width=(\linewidth-2mm)/4,before=,after=\hfill,
colframe=blue!75!black,colback=white,height=2cm}

\foreach \myalign in {top,center,bottom,scale}
{\begin{tcolorbox}[valign=\myalign]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}}
```



N 2015-05-07 **/tcb/valign upper=⟨*alignment*⟩** (no default, initially `top`)

Alias for `/tcb/valign`.

/tcb/valign lower=⟨*alignment*⟩ (no default, initially `top`)

This key has the same meaning for the lower part as `valign` for the upper part, i. e., it determines the vertical ⟨*alignment*⟩ of the lower part with feasible values `top`, `center`, `bottom`, `scale`, and `scale*`.

N 2015-07-16 **/tcb/valign scale limit=⟨*real number*⟩** (no default, initially `1.1`)

Sets an upper scale limit for the `scale*` setting in `/tcb/valign` and `/tcb/valign lower`. Note that this value is not reset by `/tcb/reset` → P. 98. So, changes also apply to embedded boxes.

Also see `/tcb/sidebyside align` → P. 109 for alignment settings when upper part and lower part are set side-by-side.

4.7 Geometry

4.7.1 Width

`/tcb/width`=⟨*length*⟩ (no default, initially `\linewidth`)

Sets the total width of the colored box to ⟨*length*⟩. See also `/tcb/height` → P. 52.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[width=\linewidth/2]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

N 2014-10-31 `/tcb/text width`=⟨*length*⟩ (style, no default)

Sets the text width of the upper part to ⟨*length*⟩. See also `/tcb/text height` → P. 53.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[text width=4cm]
This is a \textbf{tcolorbox} where the text has a width of 4cm.
\end{tcolorbox}
```

This is a **tcolorbox** where the text has a width of 4cm.

N 2014-11-07 `/tcb/add to width`=⟨*length*⟩ (style, no default)

Adds ⟨*length*⟩ to the current total width of the colored box.

```
\tcbset{width=4cm,colback=red!5!white,
  colframe=red!75!black}

\begin{tcolorbox}
This is a \textbf{tcolorbox}.
\end{tcolorbox}

\begin{tcolorbox}[add to width=1cm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is a **tcolorbox**.

See Section 4.10 on page 52 for setting fixed height values.

4.7.2 Rules

`/tcb/toprule`=⟨*length*⟩ (no default, initially `0.5mm`)

Sets the line width of the top rule to ⟨*length*⟩.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[toprule=3mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/bottomrule`=⟨*length*⟩ (no default, initially `0.5mm`)

Sets the line width of the bottom rule to ⟨*length*⟩.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[bottomrule=3mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/leftrule`=⟨*length*⟩ (no default, initially `0.5mm`)

Sets the line width of the left rule to ⟨*length*⟩.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[leftrule=3mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/rightrule`=⟨*length*⟩ (no default, initially `0.5mm`)

Sets the line width of the right rule to ⟨*length*⟩.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[rightrule=3mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

/tcb/titlerule=⟨*length*⟩ (no default, initially 0.5mm)

Sets the line width of the rule below the title to ⟨*length*⟩.

```
\tcbset{enhanced,colback=red!5!white,colframe=red!75!black,
  colbacktitle=red!90!black}

\begin{tcolorbox}[titlerule=3mm,title=This is the title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is the title

This is a **tcolorbox**.

/tcb/boxrule=⟨*length*⟩ (style, no default, initially 0.5mm)

Sets all rules of the frame to ⟨*length*⟩, i. e. /tcb/toprule → P. 35, /tcb/bottomrule → P. 35, /tcb/leftrule → P. 35, /tcb/rightrule → P. 35, and /tcb/titlerule.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[boxrule=3mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

! More options for drawing a /tcb/borderline → P. 168 are provided by using skins documented in Section 10 from page 139.

4.7.3 Arcs

/tcb/arc=⟨*length*⟩ (no default, initially 1mm)

Sets the inner radius of the four frame arcs to ⟨*length*⟩.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[arc=0mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[arc=3mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is a **tcolorbox**.

N 2015-05-05 **/tcb/circular arc** (style, no value)

Sets /tcb/arc → P. 36 to match the half of the inner width of the colored box. If width and height of the box are identical, this gives a circle.

! If the height of the box is smaller than the width, the result will look quite ugly.

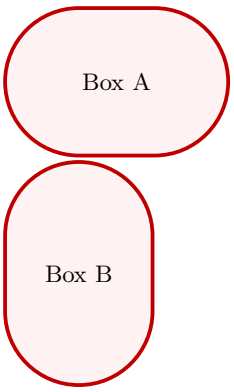
```
\begin{tcolorbox}[width=3cm,
  colback=red!5!white,
  colframe=red!75!black,
  halign=center,valign=center,
  square,circular arc]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

N 2015-05-05 **/tcb/bean arc** (style, no value)

Sets /tcb/arc → P. 36 to match the smaller value of the half of the inner width and of the inner height of the colored box.

! This only works for a fixed /tcb/height → P. 52. Also, /tcb/bean arc must be used *after* width and height are set by option keys.

```
\tcbset{size=fbox,boxrule=0.5mm,
  colback=red!5!white,
  colframe=red!75!black,
  halign=center,valign=center}

\begin{tcolorbox}[width=3cm,height=2cm,
  bean arc]
Box A
\end{tcolorbox}

\begin{tcolorbox}[width=2cm,height=3cm,
  bean arc]
Box B
\end{tcolorbox}
```



N 2015-05-05 **/tcb/octogon arc** (style, no value)

Sets /tcb/arc → P. 36 to match $\frac{1}{2+\sqrt{2}}$ of the inner width of the colored box. If width and height of the box are identical, the interior is a regular octogon.

```
\begin{tcolorbox}[enhanced,
  size=minimal,auto outer arc,
  width=2.1cm,octogon arc,
  colback=red,colframe=white,colupper=white,
  fontupper=\fontsize{7mm}{7mm}\selectfont\bfseries\sffamily,
  halign=center,valign=center,
  square,arc is angular,
  borderline={0.2mm}{-1mm}{red}  ]
STOP
\end{tcolorbox}
```

N 2015-05-05 **/tcb/arc is angular** (no value, initially unset)

Using this options applies a patch which straightens the corners arcs of the boxes. The little arcs are replaced by little straight lines.

> **!** This patch is considered as an experimental feature. It changes some of the original Ti*k*Z code. This change may break with future updates of Ti*k*Z.

```
\tcbset{colback=red!5!white,colframe=red!75!black,
  arc=3mm}

\begin{tcolorbox}[arc is angular]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[arc is curved]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is a **tcolorbox**.

N 2015-05-05 **/tcb/arc is curved** (no value, initially set)

This option resets the patch from /tcb/arc is angular. The original Ti*k*Z code is activated.

/tcb/outer arc=⟨*length*⟩ (no default, initially unset)

Sets the outer radius of the four frame arcs to ⟨*length*⟩.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[arc=4mm,outer arc=1mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

/tcb/auto outer arc (no value, initially set)

Sets the outer radius of the four frame arcs automatically in dependency of the inner radius given by /tcb/arc → P. 36.

4.7.4 Spacing

/tcb/boxsep=⟨*length*⟩ (no default, initially 1mm)

Sets a common padding of ⟨*length*⟩ between the text content and the frame of the box. This value is added to the key values of left, right, top, bottom, and middle at the appropriate places.

```
\tcbset{colback=red!5!white,colframe=red!75!black,width=(\linewidth-4mm)/2,
  before=,after=\hfill}

\begin{tcolorbox}[boxsep=5mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[boxsep=5mm,draft]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

frame: w=195.33255pt, h=48.97505pt

upper: w=141.2724pt, h=6.296pt

interior: w=192.4873pt, h=46.1298pt

`/tcb/left`=⟨*length*⟩ (style, no default, initially `4mm`)

Sets the left space between all text parts and frame (additional to `boxsep`). This is an abbreviation for setting `lefttitle`, `leftupper`, and `leftlower` to the same value.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[left=0mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/lefttitle`=⟨*length*⟩ (no default, initially `4mm`)

Sets the left space between title text and frame (additional to `boxsep`).

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[lefttitle=3cm,title=My Title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My Title

This is a **tcolorbox**.

`/tcb/leftupper`=⟨*length*⟩ (no default, initially `4mm`)

Sets the left space between upper text and frame (additional to `boxsep`).

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[leftupper=3cm,title=My Title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My Title

This is a **tcolorbox**.

`/tcb/leftlower`=⟨*length*⟩ (no default, initially `4mm`)

Sets the left space between lower text and frame (additional to `boxsep`).

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[leftlower=3cm]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

`/tcb/right`=⟨*length*⟩ (style, no default, initially `4mm`)

Sets the right space between all text parts and frame (additional to `boxsep`). This is an abbreviation for setting `righttitle`, `rightupper`, and `rightlower` to the same value.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[width=5cm,right=2cm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/righttitle`=⟨*length*⟩ (no default, initially `4mm`)

Sets the right space between title text and frame (additional to `boxsep`).

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[width=5cm,righttitle=2cm,title=My very long title text]
This is a \textbf{tcolorbox} with standard upper box dimensions.
\end{tcolorbox}
```

My very long title text

This is a **tcolorbox** with standard upper box dimensions.

`/tcb/rightupper`=⟨*length*⟩ (no default, initially `4mm`)

Sets the right space between upper text and frame (additional to `boxsep`).

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[width=5cm,rightupper=2cm,title=My very long title text]
This is a \textbf{tcolorbox} with compressed upper box dimensions.
\end{tcolorbox}
```

My very long title text

This is a **tcolorbox** with compressed upper box dimensions.

/tcb/rightlower=⟨*length*⟩ (no default, initially 4mm)

Sets the right space between lower text and frame (additional to boxsep).

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[width=5cm,rightlower=2cm]
This is a \textbf{tcolorbox} with standard upper box dimensions.
\tcblower
This is the lower part with large space at right.
\end{tcolorbox}
```

This is a **tcolorbox** with standard upper box dimensions.

This is the lower part with large space at right.

/tcb/top=⟨*length*⟩ (no default, initially 2mm)

Sets the top space between text and frame (additional to boxsep).

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[top=0mm]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

/tcb/toptitle=⟨*length*⟩ (no default, initially 0mm)

Sets the top space between title and frame (additional to boxsep).

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[toptitle=3mm,title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

/tcb/bottom=⟨*length*⟩ (no default, initially 2mm)

Sets the bottom space between text and frame (additional to boxsep).

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[bottom=0mm]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

/tcb/bottomtitle=⟨*length*⟩ (no default, initially 0mm)

Sets the bottom space between title and frame (additional to boxsep).

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[bottomtitle=3mm,title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

/tcb/middle=⟨*length*⟩ (no default, initially 2mm)

Sets the space between upper and lower text to the separation line (additional to boxsep).

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[middle=0mm,boxsep=0mm]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.
This is the lower part.

4.7.5 Size Shortcuts

`/tcb/size=`⟨*name*⟩ (no default, initially `normal`)

Sets all geometry keys with exception of `/tcb/width` → P. 34 to predefined length values. For ⟨*name*⟩, the following values are feasible:

- `normal`: normal sized boxes e.g. of width `\linewidth`.
- `title`: title line sized boxes.
- `small`: small boxes e.g. for keyword highlighting.
- `fbox`: identical to the standard `\fbox`.
- `tight`: no padding space at all.
- `minimal`: no padding space, no box rules.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\foreach \s in {minimal,tight,fbox,small,title,normal} {
  \tcbox[size=\s,on line]{\s} }

\foreach \s in {minimal,tight,fbox,small,title,normal} {
  \tcbox[size=\s,on line,title=Test]{\s} }

\foreach \s in {minimal,tight,fbox,small,title,normal} {
  \begin{tcolorbox}[size=\s,on line,title=Test,width=2.2cm]
    \s \tcblower lower\end{tcolorbox} }
```

minimal tight fbox small title normal

Test minimal Test tight Test fbox Test small Test title Test normal

Test minimal lower Test tight lower Test fbox lower Test small lower Test title lower Test normal lower

Predefined values

	normal	title	small	fbox	tight	minimal
boxrule	0.5mm	0.4mm	0.3mm	0.4pt	0.4pt	0.0pt
boxsep	1.0mm	1.0mm	1.0mm	3.0pt	0.0pt	0.0pt
left	4.0mm	2.0mm	1.0mm	0.0pt	0.0pt	0.0pt
right	4.0mm	2.0mm	1.0mm	0.0pt	0.0pt	0.0pt
top	2.0mm	0.25mm	0.0mm	0.0pt	0.0pt	0.0pt
bottom	2.0mm	0.25mm	0.0mm	0.0pt	0.0pt	0.0pt
toptitle	0.0mm	0.0mm	0.0mm	0.0pt	0.0pt	0.0pt
bottomtitle	0.0mm	0.0mm	0.0mm	0.0pt	0.0pt	0.0pt
middle	2.0mm	0.75mm	0.5mm	1.0pt	0.2pt	0.0pt
arc	1.0mm	0.75mm	0.5mm	1.0pt	0.0pt	0.0pt
outer arc	auto	auto	auto	auto	0.0pt	0.0pt

`/tcb/oversize=`⟨*length*⟩ (style, default `0pt`)

Sets the text width of the upper part to the current line width plus an optional ⟨*length*⟩. This is achieved by changing the keys `/tcb/width` → P. 34 `/tcb/enlarge left by` → P. 81, and `/tcb/enlarge right by` → P. 81 appropriately. The resulting box is overlapping into the left and right margin of the page. Note that this style option has to be given *after* all other geometry keys!

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\textit{Normal text for comparison:}\\
\lipsum[2]

\begin{tcolorbox}[oversize,title=Oversized box]
\lipsum[2]
\end{tcolorbox}

\begin{tcolorbox}[title=Normal box]
\lipsum[2]
\end{tcolorbox}
```

Normal text for comparison:

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Oversized box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Normal box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

U 2015-11-13 **/tcb/toggle left and right=**⟨*toggle preset*⟩ (default **evenpage**, initially **none**)

According to the ⟨*toggle preset*⟩, the left and the right settings of the **tcolorbox** are switched or not. Feasible values are:

- **none**: no switching.
- **forced**: the values of the left and right rules and spaces are switched.
- **evenpage**: if the page is an even page, the values of the left and right rules and spaces are switched. This value also sets /tcb/check odd page → P. 95 to **true**.

! Horizontal bounding box enlargements are not toggled by this option. They can be toggled independently by /tcb/toggle enlargement → P. 84. For example, /tcb/oversize → P. 44 changes the bounding box.

```
% \usepackage{lipsum}
% \usetikzlibrary{patterns}
% \tcbuselibrary{skins,breakable}
\begin{tcolorbox}[skin=enhancedmiddle,breakable,
  toggle left and right,
  boxrule=0mm,top=0mm,bottom=0mm,left=1mm,right=1mm,
  rightrule=1cm,colupper=blue!25!black,
  interior style={fill overzoom image=lichtspiel.jpg,fill image opacity=0.25},
  frame style={pattern=crosshatch dots light steel blue},
  overlay={%
    \tcbifoddpage{\coordinate (X) at ([xshift=-5mm]frame.east);}
                 {\coordinate (X) at ([xshift=5mm]frame.west);}
    \fill[shading=ball,ball color=blue!50!white,opacity=0.5] (X) circle (4mm);}]
\lipsum[1-6]
\end{tcolorbox}
```

This example switches a **1cm** thick rule from the left to the right side depending on the page number. Thereby, the rule is always on the outer side of the double-sided paper. Additionally, a ball is drawn on the outer side with help of an overlay.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit

ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

4.8 Corners

The four corners of any `tcolorbox` can be set individually as `/tcb/sharp corners` or as `/tcb/rounded corners` → P. 48. These settings are also reflected in the behavior of `/tcb/borderline` → P. 168 and `/tcb/shadow` → P. 179 as one would expect.

By default, all four corners are *rounded*. So, only the `/tcb/sharp corners` option will be necessary for most use cases. The `/tcb/rounded corners` → P. 48 option can be used to revert a `/tcb/sharp corners` setting.

`/tcb/sharp corners`=⟨*position*⟩ (default `all`, initially unset)

The ⟨*position*⟩ denotes one or more of the four box corners to be set as *sharp* corners. The not assigned corners will retain their mode. Feasible values for ⟨*position*⟩ are:

- `northwest`
- `northeast`
- `southwest`
- `southeast`
- `north`
- `south`
- `east`
- `west`
- `downhill`
- `uphill`
- `all`

```
\begin{tcolorbox}[colback=red!5!white,
  colframe=red!75!black,
  sharp corners=northwest ]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

```
\begin{tcolorbox}[colback=red!5!white,
  colframe=red!75!black,
  sharp corners ]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/rounded corners`=⟨*position*⟩ (default `all`, initially `all`)

The `/tcb/rounded corners` can be used to revert a `/tcb/sharp corners` → P. 47 setting. The ⟨*position*⟩ denotes one or more of the four box corners to be set as *rounded* corners. The not assigned corners will retain their mode. Feasible values for ⟨*position*⟩ are[2]:

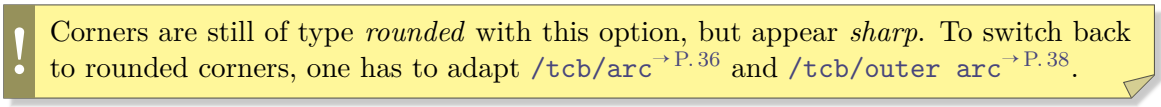
- `northwest`
- `northeast`
- `southwest`
- `southeast`
- `north`
- `south`
- `east`
- `west`
- `downhill`
- `uphill`
- `all`

```
\begin{tcolorbox}[colback=red!5!white,
  colframe=red!75!black,sharp corners,
  rounded corners=northwest ]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/sharpish corners` (style, no value)

Shortcut for setting `/tcb/arc` → P. 36 and `/tcb/outer arc` → P. 38 to `0pt`. With this setting, rounded corners will appear as quasi-sharp, but e. g. the shadow will be somewhat rounder than the shadow of really sharp corners.




```
\begin{tcolorbox}[colback=red!5!white,
  colframe=red!75!black,
  sharpish corners ]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

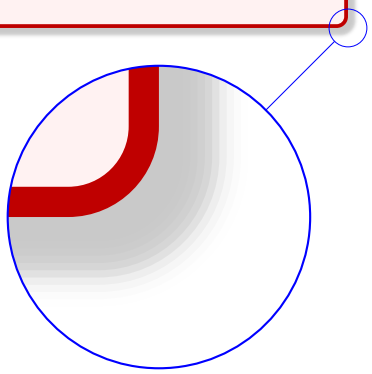
[2]The graphical examples assume that the boxes where set to have sharp corners before.

The following examples will show the differences between /tcb/rounded corners → P. 48, /tcb/sharpish corners → P. 48, and /tcb/sharp corners → P. 47. The later two give the same core box, but /tcb/borderline → P. 168 and /tcb/shadow → P. 179 settings are slightly different. The following examples use /tcb/drop fuzzy shadow → P. 173.

My title

This is a **tcolorbox**.

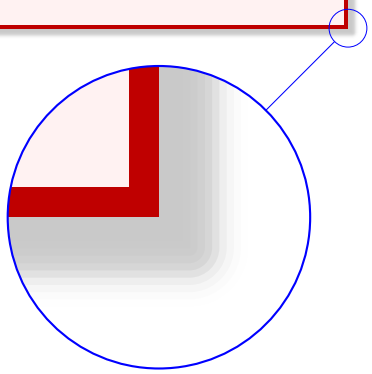
rounded corners



My title

This is a **tcolorbox**.

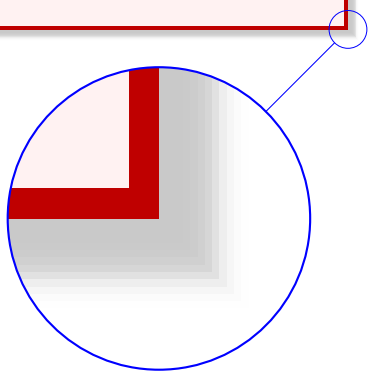
sharpish corners



My title

This is a **tcolorbox**.

sharp corners



4.9 Transparency

Transparency effects are likely to be used in conjunction with *jigsaw* skin variants, see Section 10.10 on page 190.

`/tcb/opacityframe`=⟨*fraction*⟩ (no default, initially `1.0`)

Sets the frame opacity of the box to the given ⟨*fraction*⟩.

```
\begin{tcolorbox}[opacityframe=0.25,
  colframe=red]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/opacityback`=⟨*fraction*⟩ (no default, initially `1.0`)

Sets the background opacity of the box to the given ⟨*fraction*⟩.

```
\begin{tcolorbox}[standard jigsaw,colframe=red,
  opacityframe=0.5, opacityback=0.5]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/opacitybacktitle`=⟨*fraction*⟩ (no default, initially `1.0`)

Sets the title background opacity of the box to the given ⟨*fraction*⟩.

```
\begin{tcolorbox}[standard jigsaw,colframe=red,
  opacityframe=0.5, opacitybacktitle=0.5,
  title filled, title=This is a title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a title

This is a **tcolorbox**.

`/tcb/opacityfill`=⟨*fraction*⟩ (style, no default, initially `1.0`)

Sets the fill opacity for frame, interior and optionally the title background to the given ⟨*fraction*⟩.

```
\begin{tcolorbox}[standard jigsaw,colframe=red,
  opacityfill=0.7, title=This is a title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a title

This is a **tcolorbox**.

/tcb/opacityupper=⟨*fraction*⟩ (no default, initially 1.0)

Sets the text opacity of the upper box part to the given ⟨*fraction*⟩.

```
\begin{tcolorbox}[enhanced,opacityupper=0.5,
  interior style={pattern=checkerboard light gray}]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

/tcb/opacitylower=⟨*fraction*⟩ (no default, initially 1.0)

Sets the text opacity of the lower box part to the given ⟨*fraction*⟩.

```
\begin{tcolorbox}[enhanced,opacitylower=0.5,
  interior style={pattern=checkerboard light gray}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

/tcb/opacitytext=⟨*fraction*⟩ (no default, initially 1.0)

Sets the text opacity of the upper and the lower box part to the given ⟨*fraction*⟩.

```
\begin{tcolorbox}[enhanced,opacitytext=0.5,
  interior style={pattern=checkerboard light gray}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

/tcb/opacitytitle=⟨*fraction*⟩ (no default, initially 1.0)

Sets the text opacity of the box title to the given ⟨*fraction*⟩.

```
\begin{tcolorbox}[enhanced,opacitytitle=0.7,
  coltitle=black,
  fonttitle=\bfseries,title=This is a title,
  title style={pattern=checkerboard light gray}]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a title

This is a **tcolorbox**.

```
\begin{tcolorbox}[enhanced jigsaw,fonttitle=\bfseries,title=This is a title,
  opacityframe=0.5,opacityback=0.25,opacitybacktitle=0.25,opacitytext=0.8,
  colback=red!5!white,colframe=red!75!black,colbacktitle=yellow!20!red]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a title

This is a **tcolorbox**.

4.10 Height Control

In a typical usage scenario, the height of a `tcolorbox` is computed automatically to fit the content. Nevertheless, the height can be set to a fixed value or to fit commonly for several boxes, e. g. if boxes are set side by side.

> **!** The height control keys are only applicable to unbreakable boxes. If a box is set to be `/tcb/breakable` → P. 351, the height is always computed according to the *natural height*.

`/tcb/natural height` (no value, initially set)

Sets the total height of the colored box to its natural height depending on the box content.

`/tcb/height`=⟨*length*⟩ (no default)

Sets the total height of the colored box to ⟨*length*⟩ independent of the box content. ⟨*length*⟩ is the minimum height of the box, if `/tcb/height plus` is larger than zero.

```
\tcbset{width=(\linewidth-2mm)/3,before=,after=\hfill,
colframe=blue!75!black,colback=white}

\begin{tcolorbox}[height=1cm,valign=center]
  This box has a height of 1cm.
\end{tcolorbox}
\begin{tcolorbox}[height=2cm,valign=center]
  This box has a height of 2cm.
\end{tcolorbox}
\begin{tcolorbox}[height=3cm,split=0.5,valign=center,valign lower=center]
  This box has a height of 3cm.
  \tcblower
  Lower part.
\end{tcolorbox}
```

This box has a height of 1cm.

This box has a height of 2cm.

This box has a height of 3cm.

Lower part.

`/tcb/height plus`=⟨*length*⟩ (no default, initially `0pt`)

The box may extend a given fixed `/tcb/height` up to the given ⟨*length*⟩.

```
\tcbset{colback=red!5!white,colframe=red!75!black,left=1mm,top=1mm,bottom=1mm,
  right=1mm,boxsep=0mm,width=3cm,nobeforeafter}

\begin{tcolorbox}[height=1cm]
This is a tcolorbox.
\end{tcolorbox}
\begin{tcolorbox}[height=1cm,height plus=1cm]
This is a tcolorbox.
\end{tcolorbox}
\begin{tcolorbox}[height=1cm,height plus=1cm]
This is a tcolorbox. This is a tcolorbox. This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

This is a tcolorbox.

This is a tcolorbox. This is a tcolorbox. This is a tcolorbox.

`/tcb/height from=`⟨*min*⟩ `to` ⟨*max*⟩ (style, no default)

Sets the box height to a dimension between ⟨*min*⟩ and ⟨*max*⟩.

```
% \usepackage{lipsum}
\newtcolorbox{mybox}{colback=red!5!white,colframe=red!75!black,left=1mm,top=1mm,
  bottom=1mm,right=1mm,boxsep=0mm,width=4.5cm,nobeforeafter,
  height from=2cm to 8cm}

\begin{mybox}
This is a tcolorbox.
\end{mybox}
\begin{mybox}
This is a tcolorbox. This is a tcolorbox. This is a tcolorbox.
\end{mybox}
\begin{mybox}
\lipsum[2]
\end{mybox}
```

This is a tcolorbox.

This is a tcolorbox. This is a tcolorbox. This is a tcolorbox.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

N 2014-10-31 `/tcb/text height=`⟨*length*⟩ (style, no default)

Sets the text height to ⟨*length*⟩. This is the length from the top of the upper part to the bottom of the optional lower part. See also `/tcb/text width` → P. 34.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[text height=2cm]
This is a \textbf{tcolorbox} where the text area has a height of 2cm.
\end{tcolorbox}
```

This is a **tcolorbox** where the text area has a height of 2cm.

N 2014-11-07 **/tcb/add to height=⟨*length*⟩** (style, no default)

Adds ⟨*length*⟩ to the current height of the colored box. /tcb/height → P. 52 has to be set before this key is used! If this option is used several times, then the /tcb/height → P. 52 is also increased several times.

```
\tcbset{height=1cm,
  valign=center,width=(\linewidth-2mm)/2,
  before=,after=\hfill,colframe=blue!75!black,colback=white}

\begin{tcolorbox}
  This box has a height of 1cm.
\end{tcolorbox}
\begin{tcolorbox}[add to height=1cm]
  This box has a height of 2cm.
\end{tcolorbox}
```

This box has a height of 1cm.

This box has a height of 2cm.

N 2016-02-16 **/tcb/add to natural height=⟨*length*⟩** (style, no default)

The application of this option generates a box with natural height plus the given ⟨*length*⟩. If this option is used several times, then the last setting of ⟨*length*⟩ wins. The resulting box is not considered a fixed height box and the implementation is quite different to /tcb/add to height.

```
\tcbset{valign=center,width=(\linewidth-2mm)/2,
  before=,after=\hfill,colframe=blue!75!black,colback=white}

\begin{tcolorbox}
  This box has natural height.
\end{tcolorbox}
\begin{tcolorbox}[add to natural height=1cm]
  This box has natural height plus 1 cm.
\end{tcolorbox}
```

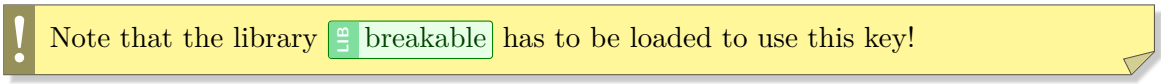
This box has natural height.

This box has natural height plus 1 cm.

N 2014-09-22
U 2016-02-17

`/tcb/height fill=true|false|maximum` (default `true`, initially `false`)

If set to `true`, the height of the `tcolorbox` is set to the rest of the available vertical space of the current page. If set to `maximum`, the page is compressed as much as possible. Note that the `tcolorbox` is always set as its own paragraph using this option. Also see `/tcb/text fill` → P. 65.



This height control key is only applicable to unbreakable boxes but it uses code from the library LIB breakable. The counterpart for breakable boxes is `/tcb/height fixed for` → P. 356.

This option can and should not be used for boxes in boxes, but it can be used for boxes inside a `tcbraster` → P. 267.

```
% \usepackage{lipsum}
% \tcbuselibrary{breakable}
\begin{tcolorbox}[height fill,
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  title=Box which fills the rest of the page]
\lipsum[1]
\end{tcolorbox}
```

Box which fills the rest of the page

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

N 2015-05-05 **/tcb/square** (style, no value)

Sets /tcb/height → P. 52 to match the width of the colored box.

```
\begin{tcolorbox}[width=3cm,
  colback=red!5!white,
  colframe=red!75!black,
  halign=center,valign=center,
  square]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

/tcb/space=⟨*fraction*⟩ (no default, initially 0)

If the height of a `tcolorbox` is not the natural height, the space difference between the forced and the natural size is distributed between the upper and the lower part of the box. This space could also be negative. ⟨*fraction*⟩ with a value between 0 and 1 is the amount of space which is added to the upper part, the rest is added to the lower part. If there is no lower part, then all of the space is added to the upper part always.

```
\tcbset{width=(\linewidth-2mm)/3,before=,after=\hfill,
colframe=blue!75!black,colback=white,height=3cm}

\foreach \f in {0.2,0.4,0.7}
{\begin{tcolorbox}[space=\f]
    This is the upper part.
    \tcblower
    This is the lower part.
\end{tcolorbox}}
```

This is the upper part.

This is the lower part.

This is the upper part.

This is the lower part.

This is the upper part.

This is the lower part.

/tcb/space to upper (style)

This is an abbreviation for `space=1`, i. e. all extra space is added to the upper part.

/tcb/space to lower (style, initially set)

This is an abbreviation for `space=0`, i. e. all extra space is added to the lower part (if there is any).

`/tcb/space to both` (style)

This is an abbreviation for `space=0.5`, i. e. the extra space equally distributed between the upper and the lower part.

```
\tcbset{width=(\linewidth-2mm)/3,before=,after=\hfill,
colframe=blue!75!black,colback=white,height=3cm}

\foreach \myspace in {space to upper,space to both,space to lower}
{\begin{tcolorbox}[\myspace]
    This is the upper part.
    \tcblower
    This is the lower part.
\end{tcolorbox}}
```

This is the upper part.

This is the lower part.

This is the upper part.

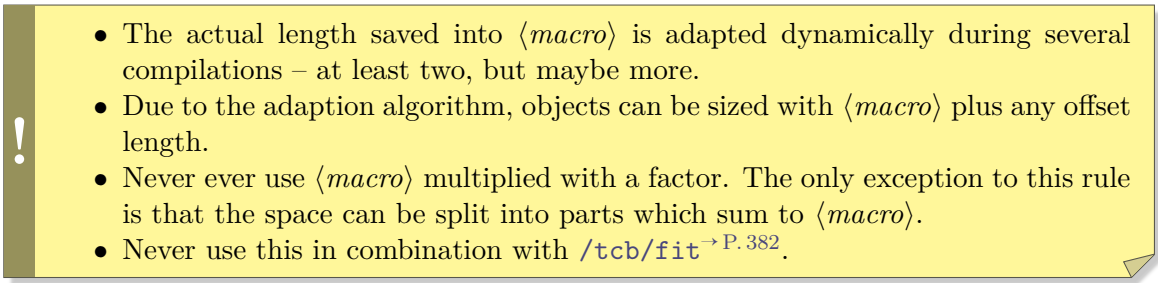
This is the lower part.

This is the upper part.

This is the lower part.

N 2015-02-15 `/tcb/space to=`⟨*macro*⟩ (no default, initially unset)

If the height of a `tcolorbox` is not the natural height, the space difference between the forced and the natural size is saved into the given local ⟨*macro*⟩. This ⟨*macro*⟩ can and should be used inside the box content to add content which is vertically sized to match ⟨*macro*⟩.



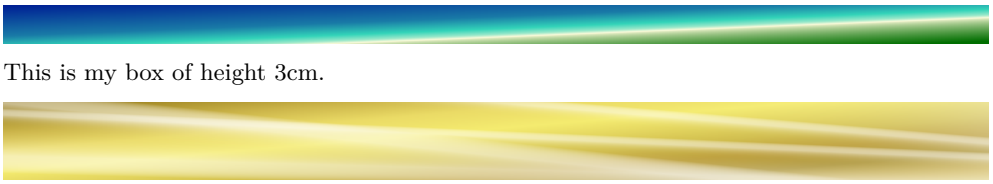
- The actual length saved into ⟨*macro*⟩ is adapted dynamically during several compilations – at least two, but maybe more.
- Due to the adaption algorithm, objects can be sized with ⟨*macro*⟩ plus any offset length.
- Never ever use ⟨*macro*⟩ multiplied with a factor. The only exception to this rule is that the space can be split into parts which sum to ⟨*macro*⟩.
- Never use this in combination with `/tcb/fit` → P. 382.

```
\begin{tcolorbox}[colframe=blue!75!black,colback=white,height=3cm,
    space to=\myspace]
  This is my box of height 3cm. The space is filled with a picture:\\[2mm]
  \includegraphics[width=\linewidth,height=\myspace]{goldshade.png}\\[1mm]
  This is some other text.
\end{tcolorbox}
```

This is my box of height 3cm. The space is filled with a picture:

This is some other text.

```
\begin{tcolorbox}[colframe=blue!75!black,colback=white,height=3cm,
    space to=\myspace]
  \includegraphics[width=\linewidth,
    height=0.33\dimexpr\myspace]{blueshade.png}\\[1mm]
  This is my box of height 3cm.\\[2mm]
  \includegraphics[width=\linewidth,
    height=0.67\dimexpr\myspace]{goldshade.png}
\end{tcolorbox}
```



This is my box of height 3cm.

/tcb/split=⟨*fraction*⟩ (no default)

If the height of a `tcolorbox` is not the natural height, the ⟨*fraction*⟩ with a value between 0 and 1 determines the positioning of the segmentation between the upper and the lower part. Here, 0 stands for top and 1 for bottom. Note that the box is split regardless of the actual dimensions of the text parts!

```
\tcbset{width=(\linewidth-2mm)/3,before=,after=\hfill,height=3cm,
colback=white,colframe=blue!75!black,valign=center,valign lower=center}

\foreach \f in {0.1,0.5,0.8}
{\begin{tcolorbox}[split=\f]
This is the upper part.
\tcblower
This is the lower part with a lot of text in several lines.
\end{tcolorbox}}
```

This is the upper part.

This is the lower part with a lot of text in several lines.

This is the upper part.

This is the lower part with a lot of text in several lines.

This is the upper part.

This is the lower part with a lot of text in several lines.

U 2014-11-07 **/tcb/equal height group=**⟨*id*⟩ (no default)

Boxes which are members of an **equal height group** will all get the same height, i. e. the maximum of all their natural heights. The ⟨*id*⟩ serves to distinguish between different height groups. Note that you have to compile twice to see changes and that height groups are global definitions.

```
\tcbset{width=(\linewidth-2mm)/3,before=,after=\hfill,arc=0mm,
colframe=blue!75!black,colback=white,fonttitle=\bfseries}

\begin{tcolorbox}[equal height group=A,adjusted title={One}]
  My smallest box.
\end{tcolorbox}%
\begin{tcolorbox}[equal height group=A,adjusted title={Two}]
  This box is also small.
  \tcblower
  But with a lower part.
\end{tcolorbox}%
\begin{tcolorbox}[equal height group=A,adjusted title={Three}]
  This box contains a lot of text just to fill the space
  with word flowing and flowing and flowing until the box
  is filled with all of it.
\end{tcolorbox}\linebreak
%
\tcbset{width=(\linewidth-1mm)/2,before=,after=\hfill,arc=0mm,
colframe=red!75!black,colback=white}
%
\begin{tcolorbox}[equal height group=B]
  Now, we use another equal height group.
\end{tcolorbox}%
\begin{tcolorbox}[equal height group=B,after=]
  \begin{equation*}
    \int\limits_{0}^{1} x^2 = \frac13.
  \end{equation*}
\end{tcolorbox}
```

One

My smallest box.

Two

This box is also small.

But with a lower part.

Three

This box contains a lot of text just to fill the space with word flowing and flowing and flowing until the box is filled with all of it.

Now, we use another equal height group.

$$\int_0^1 x^2 = \frac{1}{3}.$$

See Section 14 on page 265 for more equal height options.

`/tcb/minimum for equal height group=`⟨*id*⟩`:`⟨*length*⟩ (no default, initially unset)

Plants a ⟨*length*⟩ into the equal height group with the given ⟨*id*⟩. This ensures that the height will not drop below ⟨*length*⟩. Note that you cannot reduce a computed height value by using this key with a small value. The difference to applying `/tcb/height` → P. 52 directly is that the boxes are never too small for their content.

```
\tcbset{colframe=blue!75!black,colback=white,arc=0mm,
  before=,after=\hfill,fonttitle=\bfseries,left=2mm,right=2mm,
  width=3.5cm,
  equal height group=C,
  minimum for equal height group=C:3.5cm}

\begin{tcolorbox}
  My first box. All boxes will get 3.5cm times 3.5cm
  if the content height is not too large.
\end{tcolorbox}%
\begin{tcolorbox}
  My second box.
  \tcblower
  This is the lower part.
\end{tcolorbox}%
\begin{tcblisting}{}
\textbf{Mixed}
with a listing.
\end{tcblisting}
\begin{tcolorbox}[title={Fourth box}]
  My final box.
\end{tcolorbox}%
```

My first box. All boxes will get 3.5cm times 3.5cm if the content height is not too large.

My second box.

This is the lower part.

`\textbf{Mixed}` with a listing.

Mixed with a listing.

Fourth box

My final box.

N 2016-03-24 `/tcb/minimum for current equal height group=`⟨*length*⟩ (no default, initially unset)

Sets `/tcb/minimum for equal height group` for the current equal height group. Apparently, this only works for an already known equal height group, i.e. `/tcb/equal height group` → P. 59 has to be set *before* this option is used. This option is likely to be used in combination with `/tcb/raster equal height` → P. 275

```
% \tcbuselibrary{raster}
\begin{tcbitemize}[raster equal height,colframe=blue!75!black,colback=white,
  raster every box/.style={minimum for current equal height group=2cm}]
  \tcbitem A
  \tcbitem B
\end{tcbitemize}
```

A

B

N 2015-11-27
U 2016-02-22

/tcb/use height from group=⟨*id*⟩ (style, default current group)

Sets the current box to a fixed /tcb/height → P. 52 which is copied from an equal height group with the given ⟨*id*⟩. If this height is not available during the current compilation, no fixed height setting is used. If ⟨*id*⟩ is omitted, the current equal height group is used which has to be set before by /tcb/equal height group → P. 59.

Note that the natural height of the current box is not considered for computation of the group height. The main application for /tcb/use height from group is that the height can be adapted further by /tcb/add to height → P. 54.

```
\begin{tcolorbox}[use height from group=C,add to height=-2cm,
  colframe=blue!75!black,colback=white]
Height from group 'C' of the previous example, but reduced by 2cm.
\end{tcolorbox}%
```

Height from group 'C' of the previous example, but reduced by 2cm.

```
% \tcbuselibrary{raster}
Every line is inside an equal height group:
\begin{tcbraster}[raster equal height=rows,
    title=Box \thetcbrasternum,
    enhanced,size=small,colframe=red!50!black,colback=red!10!white]
  \begin{tcolorbox}First line\\second line\\
    The height of this box rules.\end{tcolorbox}
  \begin{tcolorbox}[use height from group]Test\end{tcolorbox}
  \begin{tcolorbox}[use height from group]
    First line\\second line\end{tcolorbox}
  \begin{tcolorbox}The height of this box rules.\end{tcolorbox}
\end{tcbraster}
```

Every line is inside an equal height group:

Box 1
First line
second line
The height of this box rules.

Box 2
Test

Box 3
First line
second line

Box 4
The height of this box rules.

N 2015-11-27

\tcbheightfromgroup{⟨*macro*⟩}{⟨*id*⟩}

Saves the height from an equal height group with the given ⟨*id*⟩ to a ⟨*macro*⟩. If this height is not available during the current compilation, ⟨*macro*⟩ is set to `0pt`.

4.11 Box Content Additions

The following options introduce some arbitrary ⟨*code*⟩ to the content of a `tcolorbox`. These additions can be given at the beginning or at the ending of the title, the upper part, or the lower part.

`/tcb/before title=`⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is placed *after* the color and font settings and *before* the content of the title.

```
\tcbset{before title={\textcolor{yellow}{\large Important:}~},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

Important: My title

This is a **tcolorbox**.

`/tcb/after title=`⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is placed *after* the content of the title.

```
\tcbset{after title={\hfill\colorbox{Navy}{approved}},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title approved

This is a **tcolorbox**.

`/tcb/before upper=`⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is placed *after* the color and font settings and *before* the content of the upper part.

```
\tcbset{before upper={\textit{The story:}\par},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

The story:
This is a **tcolorbox**.

U 2015-11-27 `/tcb/after upper=`⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is placed *after* the content of the upper part.

```
\tcbset{after upper={\par\hfill\textit{Read more next week}},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

Read more next week

N 2015-11-27 `/tcb/after upper*=`⟨*code*⟩ (no default, initially unset)

An `\unskip` and the given ⟨*code*⟩ is placed *after* the content of the upper part.

```
\begin{tcolorbox}[before upper=\flqq,after upper*=\frqq,
  colback=red!5!white,colframe=red!75!black]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

«This is a **tcolorbox**.»

! Prior to version 3.80, `/tcb/after upper` had the functionality of `/tcb/after upper*`.

`/tcb/before lower=`⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is placed *after* the color and font settings and *before* the content of the lower part.

```
\tcbset{before lower=\textit{Behold:~},colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

Behold: This is the lower part.

U 2015-11-27 `/tcb/after lower=`⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is placed *after* the content of the lower part.

```
\begin{tcolorbox}[before lower=$,after lower=$,
  colback=red!5!white,colframe=red!75!black]
This is a \textbf{tcolorbox}.
\tcblower
\sin^2(x)+\cos^2(x)=1.
\end{tcolorbox}
```

This is a **tcolorbox**.

$\sin^2(x) + \cos^2(x) = 1.$

N 2015-11-27 `/tcb/after lower*=`⟨*code*⟩ (no default, initially unset)

An `\unskip` and the given ⟨*code*⟩ is placed *after* the content of the lower part.

```
\begin{tcolorbox}[after lower*=\ \textit{This is the end.},
  colback=red!5!white,colframe=red!75!black]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part. *This is the end.*

Prior to version 3.80, `/tcb/after lower` had the functionality of `/tcb/after lower*`.

If /tcb/text fill is used, one cannot have a lower part and the box is unbreakable.

N 2015-07-15 **/tcb/text fill** (style, no value)

This style sets /tcb/before upper →P. 63 and /tcb/after upper →P. 63 to embedd the upper part with a minipage. If a fixed height was applied e.g. by /tcb/height →P. 52 or /tcb/height fill →P. 55, this minipage gets a matching height. This allows to use vertical glue macros like \vfill to act like expected. If the box has no fixed height, setting /tcb/text fill has no other effect as making the box unbreakable.

```
\begin{tcolorbox}[colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  height=8cm,text fill,
  title=My filled box]
This is a \textbf{tcolorbox}.
\par\vfill
\begin{center}
  My middle text.
\end{center}
\par\vfill
This is the end of my box.
\end{tcolorbox}
```

My filled box

This is a **tcolorbox**.

My middle text.

This is the end of my box.

If /tcb/tabularx or /tcb/tabularx* are used, one cannot have a lower part.

/tcb/tabularx=⟨*preamble*⟩ (style)

This style sets /tcb/before upper → P. 63 and /tcb/after upper → P. 63 and several geometry keys to support a tabularx with the given ⟨*preamble*⟩. The packages tabularx [4], array, and colortbl have to be loaded separately.

```
% \usepackage{array,tabularx}
% \usepackage{colortbl} - or - \usepackage[table]{xcolor}
\newcolumntype{Y}{>{\raggedleft\arraybackslash}X}% see tabularx
\tcbset{enhanced,fonttitle=\bfseries\large,fontupper=\normalsize\sffamily,
  colback=yellow!10!white,colframe=red!50!black,colbacktitle=Salmon!30!white,
  coltitle=black,center title}

\begin{tcolorbox}[tabularx={X||Y|Y|Y|Y||Y},title=My table]
Group & One     & Two     & Three    & Four     & Sum\\\hline\hline
Red   & 1000.00 & 2000.00 &  3000.00 &  4000.00 & 10000.00\\\hline
Green & 2000.00 & 3000.00 &  4000.00 &  5000.00 & 14000.00\\\hline
Blue  & 3000.00 & 4000.00 &  5000.00 &  6000.00 & 18000.00\\\hline\hline
Sum   & 6000.00 & 9000.00 & 12000.00 & 15000.00 & 42000.00
\end{tcolorbox}
```

My table

Group	One	Two	Three	Four	Sum
Red	1000.00	2000.00	3000.00	4000.00	10000.00
Green	2000.00	3000.00	4000.00	5000.00	14000.00
Blue	3000.00	4000.00	5000.00	6000.00	18000.00
Sum	6000.00	9000.00	12000.00	15000.00	42000.00

/tcb/tabularx*={⟨*code*⟩}{⟨*preamble*⟩} (style)

This is a variant of /tcb/tabularx which adds some ⟨*code*⟩ before the table starts.

```
% \usepackage{array,tabularx}
% \usepackage{colortbl} - or - \usepackage[table]{xcolor}
\tcbset{enhanced,fonttitle=\bfseries\large,fontupper=\normalsize\sffamily,
  colback=yellow!10!white,colframe=red!50!black,colbacktitle=Salmon!30!white,
  coltitle=black,center title}

\begin{tcolorbox}[tabularx*={\arrayrulewidth0.5mm}{X|X|X},title=My table]
One     & Two     & Three \\\hline\hline
1000.00 & 2000.00 &  3000.00\\\hline
2000.00 & 3000.00 &  4000.00
\end{tcolorbox}
```

My table

One	Two	Three
1000.00	2000.00	3000.00
2000.00	3000.00	4000.00

/tcb/tikz upper=⟨*options*⟩ (style)

This style adds a centered `tikzpicture` environment to the start and end of the upper part. The ⟨*options*⟩ may be given as Ti*k*Z picture options.

```
% \usepackage{tikz}

\begin{tcolorbox}[tikz upper,fonttitle=\bfseries,colback=white,colframe=black,
                  title=\tikzname\ drawing]
  \path[fill=yellow,draw=yellow!75!red] (0,0) circle (1cm);
  \fill[red] (45:5mm) circle (1mm);
  \fill[red] (135:5mm) circle (1mm);
  \draw[line width=1mm,red] (215:5mm) arc (215:325:5mm);
\end{tcolorbox}
```

Ti*k*Z drawing

/tcb/tikz lower=⟨*options*⟩ (style)

This style adds a centered `tikzpicture` environment to the start and end of the lower part. The ⟨*options*⟩ may be given as Ti*k*Z picture options.

```
% \usepackage{tikz}
% \tcbuselibrary{skins,listings}
\tcbset{tikz lower,listing side text,fonttitle=\bfseries,
  bicolor,colback=LightBlue!50!white,colbacklower=white,colframe=black,
  righthand width=3cm}

\begin{tcblisting}{title=\tikzname\  drawing}
\path[fill=yellow,draw=yellow!75!red]
    (0,0) circle (1cm);
\fill[red] (45:5mm) circle (1mm);
\fill[red] (135:5mm) circle (1mm);
\draw[line width=1mm,red]
    (215:5mm) arc (215:325:5mm);
\end{tcblisting}
```

Ti*k*Z drawing

```
\path[fill=yellow,draw=yellow!75!red]
    (0,0) circle (1cm);
\fill[red] (45:5mm) circle (1mm);
\fill[red] (135:5mm) circle (1mm);
\draw[line width=1mm,red]
    (215:5mm) arc (215:325:5mm);
```

/tcb/tikznode upper=⟨*options*⟩ (style)

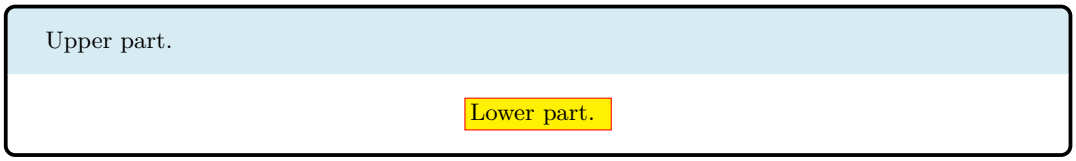
This style places the upper part content into a centered Ti*k*Z node. The ⟨*options*⟩ may be given as Ti*k*Z node options. This style is especially useful for boxes with multiline texts which are fitted to the text width.

```
% \usepackage{tikz}
\newtcbox{\headline}[1][]{enhanced,before=\begin{center},after=\end{center},
  ignore nobreak,fontupper=\Large\bfseries,
  colframe=red!50!black,colback=red!10!white,
  drop fuzzy shadow=yellow,tikznode upper,#1}

\headline{Important\\Headline}
```

/tcb/tikznode lower=⟨*options*⟩ (style)

This style places the lower part content into a centered Ti*k*Z node. The ⟨*options*⟩ may be given as Ti*k*Z node options.

```
% \usepackage{tikz}
\begin{tcolorbox}[bicolor,colback=LightBlue!50!white,colbacklower=white,
  colframe=black,tikznode lower={inner sep=2pt,draw=red,fill=yellow}]
Upper part.
\tcblower
Lower part.
\end{tcolorbox}
```



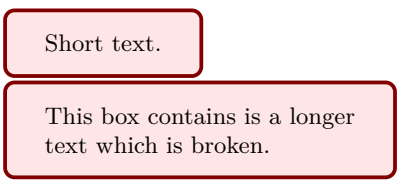
/tcb/tikznode=⟨*options*⟩ (style)

Shortcut for setting /tcb/tikznode upper and /tcb/tikznode lower the same time.

/tcb/varwidth upper=⟨*length*⟩ (style, default /tcb/width → P. 34)

This style places the upper part content into a `varwidth` environment. This style needs the `varwidth` package [1] to be loaded manually. The resulting box has a maximal width of ⟨*length*⟩. This option is only senseful for a \tcbox → P. 14.

```
% \usepackage{varwidth}
\newtcbox{\varbox}{colframe=red!50!black,
  colback=red!10!white,varwidth upper}

\varbox{Short text.}
\varbox{This box contains is a longer text
  which is broken.}
```



4.12 Overlays

With an overlay, arbitrary ⟨*graphical code*⟩ can be added to a `tcolorbox`. This code is executed *after* the frame and interior are drawn and *before* the text content is drawn. Therefore, you can decorate the `tcolorbox` with your own extensions. Common special cases are *watermarks* which are implemented using overlays. See Subsection 10.3 from page 156 if you want to add *watermarks*.

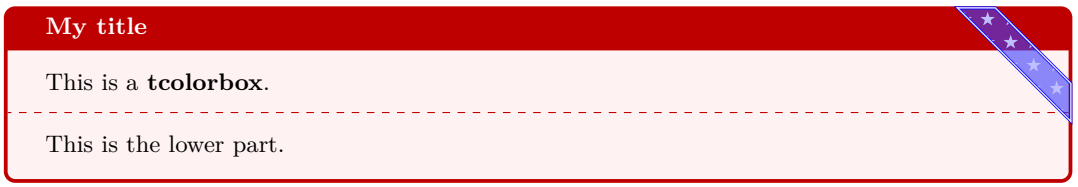
> **!** If you use the core package only, the ⟨*graphical code*⟩ has to be `pgf` code and there is not much assistance for positioning. Therefore, the usage of the `/tcb/enhanced` → P. 195 mode from the library skins is recommended which allows `tikz` code and gives access to `/tcb/geometry nodes` → P. 128 for positioning.

`/tcb/overlay=`⟨*graphical code*⟩ (no default, initially unset)

Adds ⟨*graphical code*⟩ to the box drawing process. This ⟨*graphical code*⟩ is drawn *after* the frame and interior and *before* the text content.

```
% \tcbuselibrary{skins} % preamble
\tcbset{frogbox/.style={enhanced,colback=green!10,colframe=green!65!black,
  enlarge top by=5.5mm,
  overlay={\foreach \x in {2cm,3.5cm} {
    \begin{scope}[shift={([xshift=\x]frame.north west)}]
      \path[draw=green!65!black,fill=green!10,line width=1mm] (0,0) arc (0:180:5mm);
      \path[fill=black] (-0.2,0) arc (0:180:1mm);
    \end{scope}}}}}

\begin{tcolorbox}[frogbox,title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

```
% \usetikzlibrary{patterns} % preamble
% \tcbuselibrary{skins}     % preamble
\tcbset{ribbonbox/.style={enhanced,colback=red!5!white,colframe=red!75!black,
  fonttitle=\bfseries,
  overlay={\path[fill=blue!75!white,draw=blue,double=white!85!blue,
    preaction={opacity=0.6,fill=blue!75!white},
    line width=0.1mm,double distance=0.2mm,
    pattern=fivepointed stars,pattern color=white!75!blue]
    ([xshift=-0.2mm,yshift=-1.02cm]frame.north east)
    -- ++(-1,1) -- ++(-0.5,0) -- ++(1.5,-1.5) -- cycle;}}}

\begin{tcolorbox}[ribbonbox,title=My title]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



/tcb/no overlay (style, no default, initially set)

Removes the overlay if set before.

/tcb/overlay broken=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be /tcb/breakable→ P. 351 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process. /tcb/overlay→ P. 69 overwrites this key.

/tcb/overlay unbroken=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be /tcb/breakable→ P. 351 but *is not* broken actually or if the box is set to be /tcb/unbreakable→ P. 352, then the ⟨*graphical code*⟩ is added to the box drawing process. /tcb/overlay→ P. 69 overwrites this key.

/tcb/overlay first=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be /tcb/breakable→ P. 351 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *first* part of the break sequence. /tcb/overlay→ P. 69 overwrites this key.

/tcb/overlay middle=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be /tcb/breakable→ P. 351 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *middle* parts (if any) of the break sequence. /tcb/overlay→ P. 69 overwrites this key.

/tcb/overlay last=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be /tcb/breakable→ P. 351 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *last* part of the break sequence. /tcb/overlay→ P. 69 overwrites this key.

/tcb/overlay unbroken and first=⟨*graphical code*⟩ (no default, initially unset)

This is an optimized abbreviation for setting /tcb/overlay unbroken and /tcb/overlay first together. /tcb/overlay→ P. 69 overwrites this key.

/tcb/overlay middle and last=⟨*graphical code*⟩ (no default, initially unset)

This is an optimized abbreviation for setting /tcb/overlay middle and /tcb/overlay last together. /tcb/overlay→ P. 69 overwrites this key.

/tcb/overlay unbroken and last=⟨*graphical code*⟩ (no default, initially unset)

This is an optimized abbreviation for setting /tcb/overlay unbroken and /tcb/overlay last together. /tcb/overlay→ P. 69 overwrites this key.

N 2014-09-19 **/tcb/overlay first and middle=**⟨*graphical code*⟩ (no default, initially unset)

This is an optimized abbreviation for setting /tcb/overlay first and /tcb/overlay middle together. /tcb/overlay→ P. 69 overwrites this key.

This example demonstrates the application of break sequence specific overlay options. Here, we define an environment `myexample` based on `tcolorbox` where the visible drawing is done totally by overlay keys.
Here, the first application of `myexample` produces an unbroken `tcolorbox`. The frame is drawn by the code given with /tcb/overlay unbroken.
The second application of `myexample` is broken into several parts which are drawn by the codes given with /tcb/overlay first, /tcb/overlay middle, and /tcb/overlay last.

```
% Preamble:
%\usepackage{tikz,lipsum}
%\tcbuselibrary{skins,breakable}
%\newcounter{example}
\colorlet{colexam}{red!75!black}
\newtcolorbox[use counter=example]{myexample}{%
```

```
  empty,title={Example \thetcbcounter},attach boxed title to top left,
  boxed title style={empty,size=minimal,toprule=2pt,top=4pt,
    overlay={\draw[colexam,line width=2pt]
      ([yshift=-1pt]frame.north west)--([yshift=-1pt]frame.north east);}},
  coltitle=colexam,fonttitle=\Large\bfseries,
  before=\par\medskip\noindent,parbox=false,boxsep=0pt,left=0pt,right=3mm,top=4pt,
  breakable,pad at break*=0mm,vfill before first,
  overlay unbroken={\draw[colexam,line width=1pt]
    ([yshift=-1pt]title.north east)--([xshift=-0.5pt,yshift=-1pt]title.north-|frame.east)
    --([xshift=-0.5pt]frame.south east)--(frame.south west); },
  overlay first={\draw[colexam,line width=1pt]
    ([yshift=-1pt]title.north east)--([xshift=-0.5pt,yshift=-1pt]title.north-|frame.east)
    --([xshift=-0.5pt]frame.south east); },
  overlay middle={\draw[colexam,line width=1pt] ([xshift=-0.5pt]frame.north east)
    --([xshift=-0.5pt]frame.south east); },
  overlay last={\draw[colexam,line width=1pt] ([xshift=-0.5pt]frame.north east)
    --([xshift=-0.5pt]frame.south east)--(frame.south west);},%
}

\begin{myexample}
\lipsum[1]
\end{myexample}

\begin{myexample}
\lipsum[2-11]
\end{myexample}

\lipsum[12]% following text
```

Example 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Example 2

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent

euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in,

sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Floating box from `floatplacement`

This floating box is placed at the top of a page.

4.13 Floating Objects

`/tcb/floatplacement`=⟨*values*⟩ (no default, initially `htb`)

Sets ⟨*values*⟩ as default values for the usage of `/tcb/float` and `/tcb/float*`. Feasible are the usual parameters for floating objects.

```
\tcbset{enhanced,colback=red!5!white,colframe=red!75!black,
   watermark color=red!15!white}

\begin{tcolorbox}[floatplacement=t,float,
                  title=Floating box from |floatplacement|,
                  watermark text={I am floating}]
  This floating box is placed at the top of a page.
\end{tcolorbox}
```

`/tcb/float`=⟨*values*⟩ (default from `floatplacement`)

Turns the box to a floating object where ⟨*values*⟩ are the usual parameters for such floating objects. If they are not used, the placement uses the default values given by `floatplacement`.

```
\begin{tcolorbox}[float, title=Floating box from |float|,
    enhanced,watermark text={I'm also floating}]
  This box floats to a feasible place automatically. You do not have to
  use a numbering for this floating object.
\end{tcolorbox}
```

Floating box from `float`

This box floats to a feasible place automatically. You do not have to use a numbering for this floating object.

`/tcb/float*`=⟨*values*⟩ (default from `floatplacement`)

Identical to `/tcb/float`, but for wide boxes spanning the whole page width of two column documents or in conjunction with the packages `multicol` or `paracol`. Note that you have to set `width=\textwidth` additionally, if the box should span the whole page width in these cases!

```
\begin{tcolorbox}[float*=b, title=Floating box from |float*|,width=\textwidth,
    enhanced,watermark text={I'm also floating}]
  In this single column document, you will see no difference to |float|.
\end{tcolorbox}
```

`/tcb/nofloat` (style, initially set)

Turns the floating behavior off.

Floating box from `float*`

In this single column document, you will see no difference to `float`.

N 2014-09-19 **/tcb/every float=**⟨*code*⟩ (no default, initially empty)

For floating objects, the /tcb/before → P. 76 and /tcb/after → P. 76 settings are ignored. Instead, the given ⟨*code*⟩ is inserted before a floating box. If the box is /tcb/breakable → P. 351, the given ⟨*code*⟩ is inserted before every part of the break sequence. The most common use case is `every float=\centering`.

```
\tcbox[float=htb,title={Floating box},every float=\centering,
  colback=blue!50!black,colframe=blue!50!white,colbacktitle=blue!10!white,
  coltitle=black,center title]
  {\includegraphics[height=6cm]{lichtspiel.jpg}}
```

Floating box

4.14 Embedding into the Surroundings

Typically, but not necessarily, a `tcolorbox` is put inside a separate paragraph and has some vertical space before and after it. This behavior is controlled by the keys `before` and `after`.

`/tcb/before`=⟨*code*⟩ (no default, initially see `/tcb/autoparskip`)
Sets the ⟨*code*⟩ which is executed before the colored box. It is not used for floating boxes. Also, it is not used, if the box follows a heading immediately and `/tcb/ignore nobreak` → P. 79 is set to `false`.

`/tcb/after`=⟨*code*⟩ (no default, initially see `/tcb/autoparskip`)
Sets the ⟨*code*⟩ which is executed after the colored box. It is not used for floating boxes.

`/tcb/parskip` (style, no value)
Sets the keys `before` and `after` to values which are recommended, if the package `parskip` *is* used and there is no better idea for `before` and `after`. This is similar to:

```
\tcbset{parskip/.style={before={\par\pagebreak[0]\parindent=0pt},
                        after={\par}}}
```

U 2016-02-14 `/tcb/noparskip` (style, no value)
Sets the keys `before` and `after` to values which are recommended, if the package `parskip` is *not* used and there is no better idea for `before` and `after`. This is similar to:

```
\tcbset{noparskip/.style={before={\par\pagebreak[0]\smallskip\parindent=0pt},
                          after={\par\smallskip}}}
```

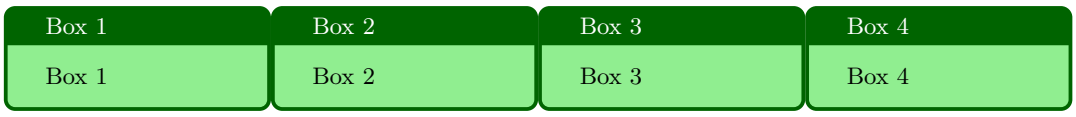
`/tcb/autoparskip` (style, no value, initially set)
Tries to detect the usage of the package `parskip` and sets the keys `before` and `after` accordingly. Actually, the following is done:

- If the length of `\parskip` is greater than `0pt` at the beginning of the document, `/tcb/parskip` is executed. Here, the usage of package `parskip` is *assumed*.
- Otherwise, if the length of `\parskip` is not greater than `0pt` at the beginning of the document, `/tcb/noparskip` is executed. Here, the absence of package `parskip` is *assumed*.

`autoparskip` is the default for the package `tcolorbox`, if `before` or `after` are not changed otherwise.

`/tcb/nobeforeafter` (style, no value)
Abbreviation for clearing the keys `before` and `after`. The colored box is not put into a paragraph and there is no space before or after the box.

```
\tcbset{myone/.style={colback=LightGreen,colframe=DarkGreen,
  equal height group=nobefaf,width=\linewidth/4,nobeforeafter}}
\begin{tcolorbox}[myone,title=Box 1]Box 1\end{tcolorbox}%
\begin{tcolorbox}[myone,title=Box 2]Box 2\end{tcolorbox}%
\begin{tcolorbox}[myone,title=Box 3]Box 3\end{tcolorbox}%
\begin{tcolorbox}[myone,title=Box 4]Box 4\end{tcolorbox}
```



`/tcb/forces nobeforeafter` (style, no value)
Forces the setting of `/tcb/nobeforeafter` even if `/tcb/before` and `/tcb/after` are set to other values later. Do not use this option globally unless you *really* know what you do. Note that embedded boxes do not inherit this forced clearance.

`/tcb/baseline=`⟨*length*⟩ (no default, initially `0pt`)

Used to set the `\pgfsetbaseline` value of the resulting `tcolorbox`.

```
\tcbset{colframe=red!50!white,width=4cm,nobeforeafter}
Some text\dotfill
\begin{tcolorbox}[baseline=3mm]
One line.
\end{tcolorbox}
\begin{tcolorbox}[baseline=3mm]
First line.\\Second line.
\end{tcolorbox}
```

Some text . . . . . One line. First line. Second line.

N 2014-10-10 `/tcb/box align=`⟨*alignment*⟩ (style, no default, initially `bottom`)

Used to set the `/tcb/baseline` value of the resulting `tcolorbox`. Feasible values for ⟨*alignment*⟩ are:

- `bottom`: alignment with the box bottom,
- `top`: alignment with the box top,
- `center`: alignment with the box center,
- `base`: alignment with the box content base. This option is not applicable for a `tcolorbox` → P. 12 but for a `\tcbox` → P. 14 only. It is an alias for `/tcb/tcbox raise base` → P. 92.

```
\tcbset{colframe=red!50!white,width=4cm,nobeforeafter}
Some text\dotfill
\begin{tcolorbox}[box align=bottom]
One line.
\end{tcolorbox}
\begin{tcolorbox}[box align=bottom]
First line.\\Second line.
\end{tcolorbox}
```

Some text . . . . . One line. First line. Second line.

```
\tcbset{colframe=red!50!white,width=4cm,nobeforeafter}
Some text\dotfill
\begin{tcolorbox}[box align=top]
One line.
\end{tcolorbox}
\begin{tcolorbox}[box align=top]
First line.\\Second line.
\end{tcolorbox}
```

Some text . . . . . One line. First line. Second line.

```
\tcbset{colframe=red!50!white,width=4cm,nobeforeafter}
Some text\dotfill
\begin{tcolorbox}[box align=center]
One line.
\end{tcolorbox}
\begin{tcolorbox}[box align=center]
First line.\\Second line.
\end{tcolorbox}
```

Some text . . . . . One line. First line. Second line.

```
\tcbset{colframe=red!50!white,nobeforeafter}
Some text\dotfill
\tcbox[nobeforeafter,box align=base]{One line}
\tcbox[nobeforeafter,box align=base,size=fbox]{Another line}
```

Some text . . . . . One line Another line

N 2014-10-10
U 2015-03-16

/tcb/before skip=⟨*glue*⟩ (style, no default)

Inserts some vertical space of the given ⟨*glue*⟩ before the colored box. This style sets /tcb/before → P. 76.

```
Some text.
\begin{tcolorbox}[before skip=1cm,
    colframe=red!50!white]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

Some text.

This is a **tcolorbox**.

N 2014-10-10
U 2015-03-16

/tcb/after skip=⟨*glue*⟩ (style, no default)

Inserts some vertical space of the given ⟨*glue*⟩ after the colored box. This style sets /tcb/after → P. 76.

```
\begin{tcolorbox}[after skip=1cm,
    colframe=red!50!white]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
Some text.
```

This is a **tcolorbox**.

Some text.

N 2014-10-10

/tcb/beforeafter skip=⟨*glue*⟩ (style, no default)

Inserts some vertical space of the given ⟨*glue*⟩ before *and* after the colored box. This style sets /tcb/before → P. 76 and /tcb/after → P. 76.

```
\tcbset{beforeafter skip=0pt,
  colframe=red!50!white}
\begin{tcolorbox}
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}
  Second box.
\end{tcolorbox}
```

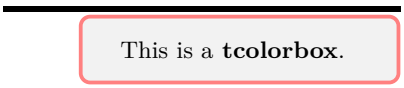
This is a **tcolorbox**.

Second box.

N 2014-11-07 **/tcb/left skip=⟨*length*⟩** (style, no default, initially 0mm)

Inserts some horizontal space of the given ⟨*length*⟩ before the colored box. This style sets /tcb/grow to left by →P. 83 with the negated ⟨*length*⟩, i.e. the bounding box and box width are changed.

```
\noindent\rule{\linewidth}{2pt}

\begin{tcolorbox}[left skip=1cm,
    colframe=red!50!white]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



N 2014-11-07 **/tcb/right skip=⟨*length*⟩** (style, no default, initially 0mm)

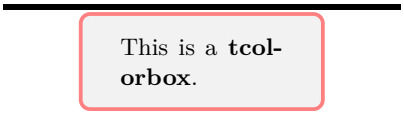
Inserts some horizontal space of the given ⟨*length*⟩ after the colored box. This style sets /tcb/grow to right by →P. 83 with the negated ⟨*length*⟩, i.e. the bounding box and box width are changed.

```
\noindent\rule{\linewidth}{2pt}

\begin{tcolorbox}[right skip=1cm,
    colframe=red!50!white]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

N 2014-10-10 **/tcb/leftright skip=⟨*length*⟩** (style, no default)

Inserts some horizontal space of the given ⟨*length*⟩ before *and* after the colored box. This style changes the bounding box and the box width.

```
\noindent\rule{\linewidth}{2pt}

\begin{tcolorbox}[leftright skip=1cm,
    colframe=red!50!white]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



N 2014-12-11 **/tcb/ignore nobreak=true|false** (default true, initially false)

After a heading, LaTeX tries to avoid a break by setting a nobreak boolean value. Starting from version 3.33, the /tcb/before →P. 76 respectively /tcb/before skip →P. 78 settings are not used after a heading if /tcb/ignore nobreak is set to **false**. For an unbreakable box, /tcb/before nobreak is used instead. Further, a /tcb/breakable →P. 351 box will also try to avoid a break between a heading and a directly following first part of a break sequence. Set /tcb/ignore nobreak to **true**, if nobreak should be ignored as prior to version 3.33. Also, such a setting may be used locally to enforce the /tcb/before →P. 76 setting.

N 2014-12-16 **/tcb/before nobreak=⟨*code*⟩** (no default, initially \noindent)

Sets the ⟨*code*⟩ which is executed before the colored box if it is unbreakable, if /tcb/ignore nobreak is not set, and if the box follows a heading.

4.15 Bounding Box

Normally, every `tcolorbox` has a bounding box which fits exactly to the dimensions of the outer frame. Therefore, LaTeX reserves exactly the space needed for the box. This behavior can be changed by enlarging (or shrinking) the bounding box. If the bounding box is enlarged, the `tcolorbox` will get some clearance around it. If the bounding box is shrunk, i. e. enlarged with negative values, the `tcolorbox` will overlap to other parts of the page. For example, the `tcolorbox` could be stretched into the page margin.

! The following examples use `/tcb/show bounding box` → P. 170 to display the actual bounding box. For this, the library LIB skins has to be included and `/tcb/enhanced` → P. 195 has to be set.

`/tcb/enlarge top initially by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the top of the box by ⟨*length*⟩. If the box is *breakable*, only the first box of the break sequence gets enlarged. `/tcb/enlarge top by` → P. 81 overwrites this key.

```
\tcbset{colframe=blue!75!black,colback=white}

\begin{tcolorbox}[enlarge top initially by=-5mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[enlarge top initially by=5mm,enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is a **tcolorbox**.

`/tcb/enlarge bottom finally by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the bottom of the box by ⟨*length*⟩. If the box is *breakable*, only the last box of the break sequence gets enlarged. `/tcb/enlarge bottom by` → P. 81 overwrites this key.

```
\tcbset{colframe=blue!75!black,colback=white}

\begin{tcolorbox}[enlarge bottom finally by=5mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[enlarge bottom finally by=-5mm,enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is a **tcolorbox**.

`/tcb/enlarge top at break by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the top of the box by ⟨*length*⟩, *if* the box is `/tcb/breakable` → P. 351. In this case, it is applied to *middle* and *last* parts in a break sequence. `/tcb/enlarge top by` overwrites this key.

`/tcb/enlarge bottom at break by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the bottom of the box by ⟨*length*⟩, *if* the box is `/tcb/breakable` → P. 351. In this case, it is applied to *first* and *middle* parts in a break sequence. `/tcb/enlarge bottom by` overwrites this key.

`/tcb/enlarge top by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the top of the box by ⟨*length*⟩. `/tcb/enlarge top initially by` → P. 80 and `/tcb/enlarge top at break by` are set to ⟨*length*⟩.

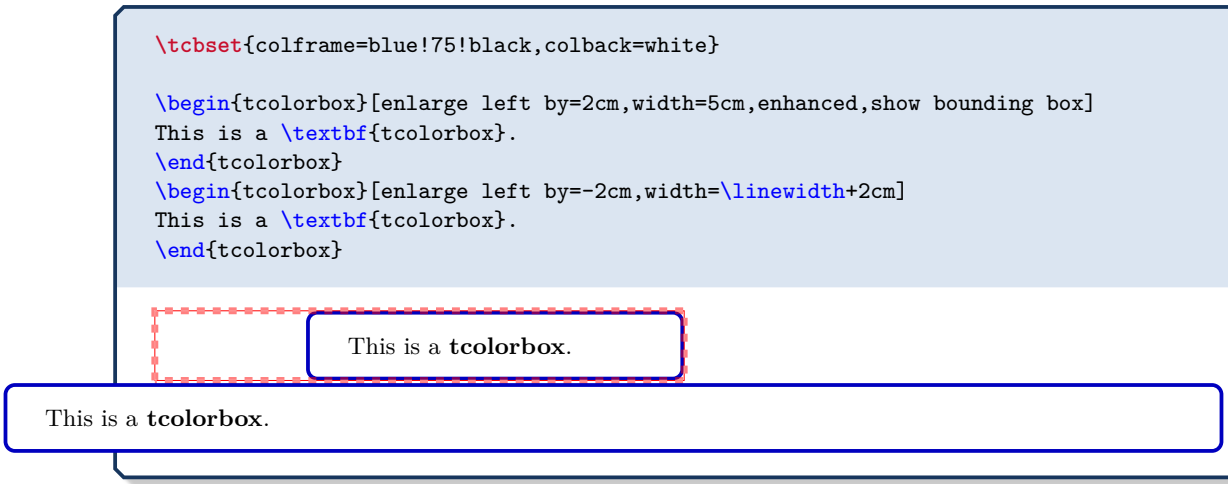
`/tcb/enlarge bottom by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the bottom of the box by ⟨*length*⟩. `/tcb/enlarge bottom finally by` → P. 80 and `/tcb/enlarge bottom at break by` are set to ⟨*length*⟩.

`/tcb/enlarge left by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the left side of the box by ⟨*length*⟩.

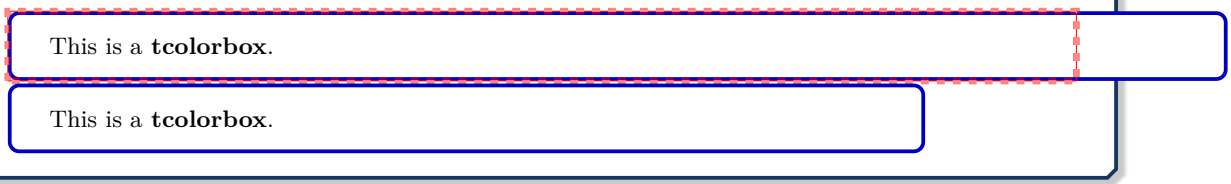
```
\tcbset{colframe=blue!75!black,colback=white}

\begin{tcolorbox}[enlarge left by=2cm,width=5cm,enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[enlarge left by=-2cm,width=\linewidth+2cm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



`/tcb/enlarge right by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the right side of the box by ⟨*length*⟩.

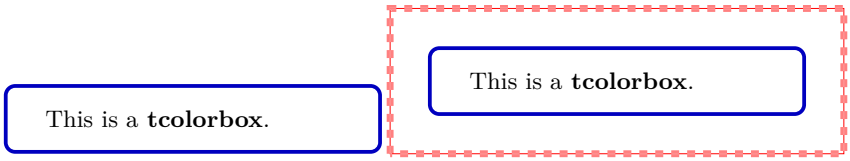
```
\tcbset{colframe=blue!75!black,colback=white}

\begin{tcolorbox}[enlarge right by=-2cm,width=\linewidth+2cm,
  enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[enlarge right by=2cm,width=\linewidth-2cm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



`/tcb/enlarge by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to all sides of the box by ⟨*length*⟩.

```
\tcbset{colframe=blue!75!black,colback=white,width=5cm,nobeforeafter}

\begin{tcolorbox}
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[enlarge by=5mm,enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



N 2015-11-20 `/tcb/flush left` (style, no value)

Enlarges the bounding box to the right side to fill the line completely.

```
\tcbset{colframe=blue!75!black,colback=white}

\begin{tcolorbox}[flush left,width=5cm,enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

N 2015-11-20 `/tcb/flush right` (style, no value)

Enlarges the bounding box to the left side to fill the line completely.

```
\tcbset{colframe=blue!75!black,colback=white}

\begin{tcolorbox}[flush right,width=5cm,enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

N 2015-11-20 `/tcb/center` (style, no value)

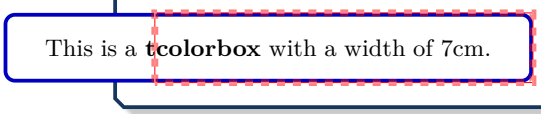
Enlarges the bounding box equally to both sides to fill the line completely.

```
\tcbset{colframe=blue!75!black,colback=white}

\begin{tcolorbox}[center,width=5cm,enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

`/tcb/grow to left by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the current box width by ⟨*length*⟩ and enlarges (shrinks) the bounding box distance to the left side of the box by −⟨*length*⟩. Also see `/tcb/left skip` → P. 79.

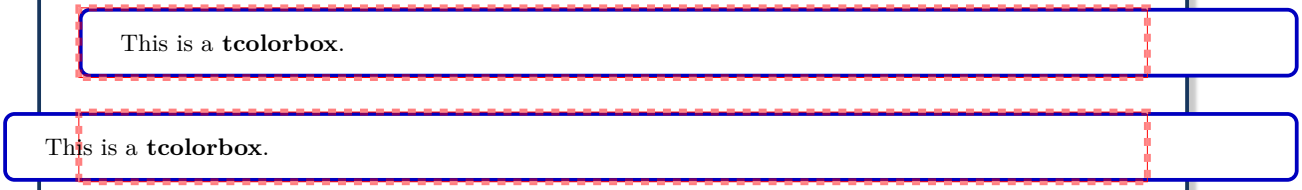
```
\tcbset{colframe=blue!75!black,colback=white}

\begin{tcolorbox}[width=5cm,grow to left by=2cm,enhanced,show bounding box]
This is a \textbf{tcolorbox} with a width of 7cm.
\end{tcolorbox}
```



`/tcb/grow to right by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the current box width by ⟨*length*⟩ and enlarges (shrinks) the bounding box distance to the right side of the box by −⟨*length*⟩. Also see `/tcb/right skip` → P. 79.

```
\tcbset{colframe=blue!75!black,colback=white}

\begin{tcolorbox}[grow to right by=2cm,enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}

\bigskip

\begin{tcolorbox}[grow to right by=2cm,grow to left by=1cm,
  enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



Floating box from `toggle enlargement`

This page is an even page. Therefore, the left and right enlargements are toggled. This box stretches to the right margin on odd pages and to the left margin on even pages. The current document is one-sided – this feature makes sense for two-sided documents only.

U 2015-11-13 `/tcb/toggle enlargement`=⟨*toggle preset*⟩ (default `evenpage`, initially `none`)

According to the ⟨*toggle preset*⟩, the left and the right enlargements of the bounding box are switched or not. Feasible values are:

- `none`: no switching.
- `forced`: the values of the left and right enlargement are switched.
- `evenpage`: if the page is an even page, the values of the left and right enlargement are switched. This value also sets `/tcb/check odd page` → P. 95 to `true`.

! See `/tcb/toggle left and right` → P. 45 to toggle geometry settings.

```
\tcbset{colframe=blue!75!black,colback=white,
  grow to left by=20mm,grow to right by=-5mm}

\begin{tcolorbox}[toggle enlargement=none,enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[toggle enlargement=forced]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[toggle enlargement=evenpage]
  This page is an \tcbifoddpage{odd}{even} page.
  Therefore, the left and right enlargements
  \tcbifoddpage{are not}{are} toggled.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is a **tcolorbox**.

This page is an even page. Therefore, the left and right enlargements are toggled.

```
\begin{tcolorbox}[colframe=red!60!black,colback=red!15!white,
  fonttitle=\bfseries,title=Floating box from \texttt{toggle enlargement},
  width=\textwidth,grow to right by=2cm,toggle enlargement,float=t]
  This page is an \tcbifoddpage{odd}{even} page.
  Therefore, the left and right enlargements \tcbifoddpage{are not}{are} toggled.
  This box stretches to the right margin on odd pages and to the left
  margin on even pages. The current document is one-sided -- this feature makes
  sense for two-sided documents only.
\end{tcolorbox}
```

The following keys should not be used with breakable boxes or boxes with a lower part.

/tcb/shrink tight (style, no value, initially unset)

The total colored box is shrunk to the dimensions of the upper part. There should be no lower part and no title. This style sets the /tcb/boxsep → P. 38 to 0pt and other geometry keys to fitting values. This option is likely to be used with the following extrusion keys.

```
\tcbset{colframe=blue!75!black,colback=white,arc=0mm,boxrule=0.4pt,
        nobeforeafter,tcbox raise base,shrink tight}

\begin{tcolorbox}
This is a \textbf{tcolorbox}.
\end{tcolorbox}

Lorem \tcbox{ipsum} dolor sit amet, consectetuer adipiscing elit.
```

This is a **tcolorbox**.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

U 2014-09-19 **/tcb/extrude left by=⟨*length*⟩** (style, no default, initially unset)

The (upper part of the) colored box is extruded by the given ⟨*length*⟩ to the left side. The inner width and the bounding box is kept unchanged and the operation is additive!

```
\tcbset{enhanced,colframe=red,colback=yellow!25!white,
  frame style={opacity=0.25},interior style={opacity=0.5},
  nobeforeafter,tcbox raise base,shrink tight,extrude by=2mm}

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.
\tcbox[extrude left by=1cm]{Curabitur} dictum gravida mauris.
Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

U 2014-09-19 **/tcb/extrude right by=⟨*length*⟩** (style, no default, initially unset)

The (upper part of the) colored box is extruded by the given ⟨*length*⟩ to the right side. The inner width and the bounding box is kept unchanged and the operation is additive!

```
\tcbset{enhanced,colframe=red,colback=yellow!25!white,
  frame style={opacity=0.25},interior style={opacity=0.5},
  nobeforeafter,tcbox raise base,shrink tight,extrude by=2mm}

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.
\tcbox[extrude right by=1cm]{Curabitur} dictum gravida mauris.
Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

/tcb/extrude top by=⟨*length*⟩ (style, no default, initially unset)

The (upper part of the) colored box is extruded by the given ⟨*length*⟩ to the top side. The inner width and the bounding box is kept unchanged and the operation is additive!

```
\tcbset{enhanced,colframe=red,colback=yellow!25!white,
  frame style={opacity=0.25},interior style={opacity=0.5},
  nobeforeafter,tcbox raise base,shrink tight,extrude by=2mm}

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.
\tcbox[extrude top by=1cm]{Curabitur} dictum gravida mauris.
Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

/tcb/extrude bottom by=⟨*length*⟩ (style, no default, initially unset)

The (upper part of the) colored box is extruded by the given ⟨*length*⟩ to the bottom side. The inner width and the bounding box is kept unchanged and the operation is additive!

```
\tcbset{enhanced,colframe=red,colback=yellow!25!white,
  frame style={opacity=0.25},interior style={opacity=0.5},
  nobeforeafter,tcbox raise base,shrink tight,extrude by=2mm}

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.
\tcbox[extrude bottom by=1cm]{Curabitur} dictum gravida mauris.
Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

/tcb/extrude by=⟨*length*⟩ (style, no default, initially unset)

The (upper part of the) colored box is extruded by the given ⟨*length*⟩ to all sides. The inner width and the bounding box is kept unchanged and the operation is additive!

```
\tcbset{enhanced,colframe=red,colback=yellow!25!white,
  frame style={opacity=0.25},interior style={opacity=0.5},
  nobeforeafter,tcbox raise base,shrink tight,extrude by=2mm}

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis. \tcbox{Curabitur} dictum
gravida mauris. \tcbox[colframe=Green,interior style={opacity=0.0}]{Nam}
arcu libero, nonummy eget, consectetuer id, \tcbox{vulputate} a, magna. Donec
vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus
et malesuada fames ac turpis egestas. \tcbox{Mauris ut leo.}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo.

4.16 Layered Boxes and Every Box Settings

A `tcolorbox` may contain another `tcolorbox` and so on. The package takes track of the nesting level using a counter `tcblayer`. Counter values may be used for doing some fancy things, but you should never change the counter value yourself.

The package takes special care for the first four layers or nesting levels, called managed layers. Here, footnote texts are administrated to find their intended place and specific layer dependent options may be set by changing `/tcb/every box on layer n` → P. 88. If needed, the number of managed layers can be increased by setting `\tcbsetmanagedlayer` → P. 88 to a higher value than 4.

The following styles have a considerable influence on how layered boxes are processed. Note especially that nested boxes are getting a `/tcb/reset` → P. 98 by default. You can change this, but be prepared for suprises if you do.

If the defaults are *not changed*, a `tcolorbox` gets its options in the following order. Following options overwrite preceding options.

1. On package load, all options are set to default values.
2. Every `\tcbset` → P. 13 command adds or changes options for the following boxes inside the current TEX group.
3. While entering a `tcolorbox`, a `/tcb/every box on layer n` → P. 88 or `/tcb/every box on higher layers` → P. 88 option list is applied. With default settings this means:
   - For layer 1 (lowest layer), the `/tcb/every box` option list is applied. Not overwritten options given by a preceding `\tcbset` → P. 13 survive.
   - For layer 2 and above (nested boxes), a `/tcb/reset` → P. 98 followed by `/tcb/every box` option list is applied. Every resettable options given by a preceding `\tcbset` → P. 13 and by the sourrounding box(es) are reset.
4. The ⟨*options*⟩ given to the `tcolorbox` are applied. Or, if the box was generated by `\newtcolorbox` → P. 15 or friends, the ⟨*options*⟩ given there are applied.
5. If the box was generated by `\newtcolorbox` → P. 15 or friends, some automated options are applied.

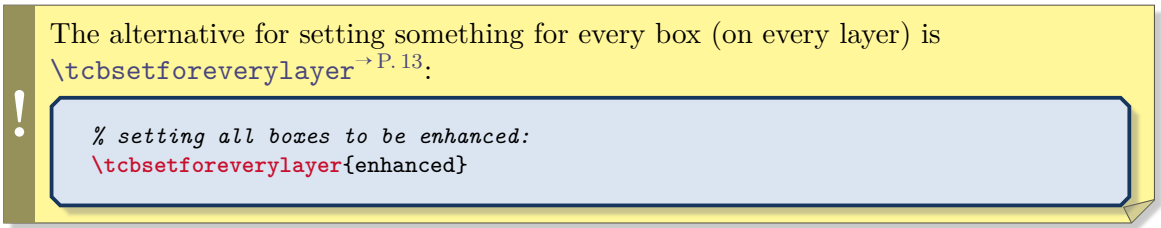
`/tcb/every box` (style)

By default, this style is empty.

```
% default setting:
\tcbset{every box/.style={}}
```

It may be changed by redefining this style.

```
% setting all boxes to be enhanced:
\tcbset{every box/.style={enhanced}}
```



/tcb/every box on layer n (style)

Here, n has to be replaced by a number ranging from 1 to the highest managed layer number (4 by default).

```
% default settings:
\tcbset{
  every box on layer 1/.style={every box},
  every box on layer 2/.style={reset,every box},
  every box on layer 3/.style={reset,every box},
  every box on layer 4/.style={reset,every box},
  }
```

/tcb/every box on higher layers (style)

Higher layers are layers above the highest managed layer number (4 by default).

```
\tcbset{every box on higher layers/.style={reset,every box}}
```

\tcbsetmanagedlayer{⟨*number*⟩**}**

Replaces the highest managed layer number by ⟨*number*⟩ where 4 is the default. This macro can only be used inside the preamble. Using a ⟨*number*⟩ lower than 4 typically makes no sense, but is not forbidden.

```
% \usepackage{lipsum}
% \tcbuselibrary{skins,breakable}
\tcbset{colframe=red!75!black,fonttitle=\bfseries,
  colback=red!5!white,
  every box/.style={enhanced,watermark text=\thetcblayer,
    before=\par\smallskip,after=\par\smallskip},
  every box on layer 2/.style={reset,every box,colback=yellow!10!white,
    drop fuzzy shadow}}
\begin{tcolorbox}[enhanced jigsaw,breakable,title=Layer 1 Box]
Here comes a footnote\footnote{Footnote from layer 1 box}.
\lipsum[2]
  \begin{tcolorbox}[title=Layer 2 Box]
  abc\footnote{The footnote of abc}
  \end{tcolorbox}
  \begin{tcolorbox}[title=Another Box,ams equation]
    \tcbhighmath{\sum\limits_{n=1}^{\infty} \frac{1}{n}} = \infty.
  \end{tcolorbox}
Some text\footnote{Footnote from some text}.
  \begin{tcolorbox}[title=Yet Another Box]
    \tcboxfit[height=2cm]{\lipsum[1]}
    My text.
    \begin{tcolorbox}
      Another lipsum text\footnote{A lipsum text}. \lipsum[3]
      \begin{tcolorbox}[title=Layer 4,colframe=blue,colback=white]
        Layer 4\footnote{Layer 4 footnote}
      \end{tcolorbox}
      The End\footnote{Last footnote}.
    \end{tcolorbox}
  \end{tcolorbox}
\end{tcolorbox}
```

Layer 1 Box

Here comes a footnote[a]. Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras

nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Layer 2 Box

abc[a]

[a]The footnote of abc

Another Box

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty. \tag{1}$$

Some text[b].

Yet Another Box

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

My text.

Another lipsum text[a]. Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Layer 4

Layer 4[a]

[a]Layer 4 footnote

The End[b].

[a]A lipsum text
[b]Last footnote

[a]Footnote from layer 1 box
[b]Footnote from some text

4.17 Capture Mode

`/tcb/capture`=⟨*mode*⟩ (no default, initially `minipage`)

The capture ⟨*mode*⟩ defines how the box content is processed.

Feasible values for ⟨*mode*⟩ are:

- `minipage`:
  This is the default ⟨*mode*⟩ for `tcolorbox` → P. 12. The content may have an upper and a lower part. Optionally, the box can be `/tcb/breakable` → P. 351. The box content is put into a minipage or into something similar to a minipage.
- `hbox`:
  This is the default ⟨*mode*⟩ for `\tcbox` → P. 14. The content cannot have a lower part and cannot be broken. The colored box is sized according to the dimensions of the content. A shortcut to set this mode is `/tcb/hbox`.
- `fitbox`: (needs the LIB fitting library)
  This is the default ⟨*mode*⟩ for `\tcboxfit` → P. 380. The content cannot have a lower part and cannot be broken. The content is sized according to the dimensions of the colored box. A shortcut to set this mode is `/tcb/fit` → P. 382.

```
\tcbset{colframe=blue!75!black,colback=white}

\begin{tcolorbox}[capture=minipage]
This is a tcolorbox.
\end{tcolorbox}

\begin{tcolorbox}[capture=hbox]
This is a tcolorbox.
\end{tcolorbox}

\begin{tcolorbox}[capture=fitbox,height=9mm]% needs the 'fitting' library
This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

This is a tcolorbox.

This is a tcolorbox.

`/tcb/hbox` (style, no default)

Shortcut for `capture=hbox`.

```
\tcbset{colframe=blue!75!black,colback=white}

\begin{tcolorbox}[hbox]
This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/minipage` (style, no default)

Shortcut for `capture=minipage`.

4.18 Text Characteristics

U 2015-10-14 `/tcb/parbox=true|false` (default `true`, initially `true`)

The text inside a `tcolorbox` is formatted using a LaTeX `minipage` if the box is unbreakable. If breakable, the box tries a mimicry of a `minipage`. In a `minipage` or `parbox`, paragraphs are formatted slightly different as the main text. If the key value is set to `false`, the normal main text behavior is restored. In some situations, this has some unwanted side effects. It is recommended that you use this experimental setting only where you really want to have this feature.

```
% \usepackage{lipsum}  % preamble
\tcbset{width=(\linewidth-2mm)/2,nobeforeafter,arc=1mm,
  colframe=blue!75!black,colback=white,fonttitle=\bfseries,fontupper=\small,
  left=2mm,right=2mm,top=1mm,bottom=1mm,equal height group=parbox}

\begin{tcolorbox}[parbox,adjusted title={parbox=true (normal)}]
  \lipsum[1-2]
\end{tcolorbox}\hfill%
\begin{tcolorbox}[parbox=false,adjusted title={parbox=false}]
  \lipsum[1-2]
\end{tcolorbox}%
```

parbox=true (normal)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

parbox=false

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/hyphenationfix=true|false` (default `true`, initially `false`)

Long words at the beginning of paragraphs in very narrow boxes will not be hyphenated using `pdflatex`. This problem is circumvented by applying the `hyphenationfix` option.

```
\tcbset{colframe=blue!75!black,
  fontupper=\normalsize,
  colback=blue!5!white,width=4cm}

\begin{tcolorbox}
Rechnungsadjunktentochter.\par
Statthaltereikonzipist.
\end{tcolorbox}

\begin{tcolorbox}[hyphenationfix]
Rechnungsadjunktentochter.\par
Statthaltereikonzipist.
\end{tcolorbox}
```

Rechnungsadjunktentochter.
Statthaltereikonzipist.

Rechnungsad-
junktentochter.
Statthal-
tereikonzipist.

! `parbox=false` and `hyphenationfix` should not be used together. They are targeting different box types and they do not blend very well.

4.19 Files

`/tcb/tempfile=`⟨*file name*⟩ (no default, initially `\jobname.tcbtemp`)

Sets ⟨*file name*⟩ as name for the temporary file which is used inside `tcbwritetemp` → P. 118 and `\tcbusetemp` → P. 118 implicitely.

4.20 `\tcbox` Specials

The following options are applicable for `\tcbox` → P. 14 and `\tcboxmath` → P. 328 only.

`/tcb/tcbox raise=`⟨*length*⟩ (no default, initially `0pt`)

Raises the `\tcbox` → P. 14 by the given ⟨*length*⟩.

```
\tcbset{colframe=blue!50!black,colback=white,colupper=red!50!black,
        fonttitle=\bfseries,nobeforeafter,center title}

Test\dotfill
\tcbox[tcbox raise base]{Hello World 1}\dotfill
\tcbox{Hello World 2}\dotfill
\tcbox[tcbox raise=5mm]{Hello World 3}
```

Test . . . . . Hello World 1 . . . . . Hello World 2 . . . . . Hello World 3

`/tcb/tcbox raise base` (style, no value, initially unset)

Raises the `\tcbox` → P. 14 such that the base of its content matches the base of the environmental line; see example above.

`/tcb/on line` (style, no value, initially unset)

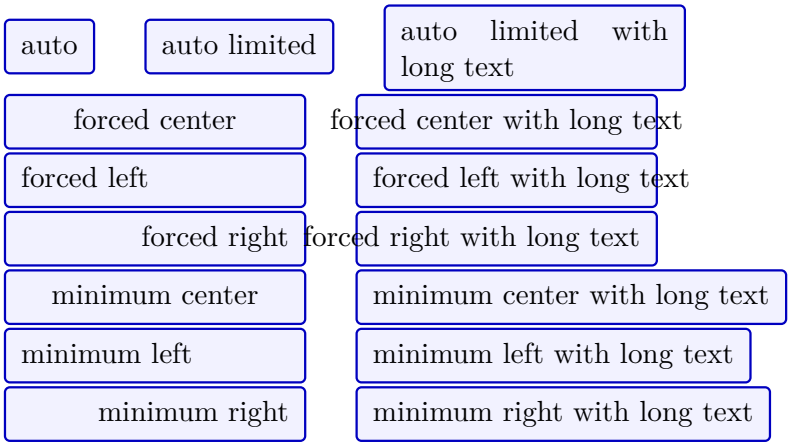
Combines `/tcb/tcbox raise base` with `/tcb/nobeforeafter` → P. 76. The resulting box behaves analogue to `\fbox`.

N 2015-03-23 **/tcb/tcbox width=⟨*mode*⟩** (no default, initially `auto`)

Controls how \tcbox → P. 14 respects a /tcb/width → P. 34 setting. Feasible values for ⟨*mode*⟩ are:

- **auto** (initial setting): ignore /tcb/width → P. 34 and set box width according to its content.
- **auto limited**: Set box width according to its content, if it is smaller than /tcb/width → P. 34. Otherwise, the content is set like in a tcolorbox → P. 12 with line breaks.
- **forced center**: Set box width according to /tcb/width → P. 34. The content is centered and may overlap the box borders.
- **forced left**: Set box width according to /tcb/width → P. 34. The content is left aligned and may overlap the box borders.
- **forced right**: Set box width according to /tcb/width → P. 34. The content is right aligned and may overlap the box borders.
- **minimum center**: Set box width according to /tcb/width → P. 34, if the content fits into. The content is centered and the box width may grow beyond /tcb/width → P. 34.
- **minimum left**: Set box width according to /tcb/width → P. 34, if the content fits into. The content is left aligned and the box width may grow beyond /tcb/width → P. 34.
- **minimum right**: Set box width according to /tcb/width → P. 34, if the content fits into. The content is right aligned and the box width may grow beyond /tcb/width → P. 34.

```
\tcbset{size=small,on line,before upper=\strut,
  colframe=blue!75!black,colback=blue!5!white,
  fontupper=\normalsize,width=4cm}

\tcbox[tcbox width=auto]{auto}\qquad
\tcbox[tcbox width=auto limited]{auto limited}\qquad
\tcbox[tcbox width=auto limited]{auto limited with long text}\\
\tcbox[tcbox width=forced center]{forced center}\qquad
\tcbox[tcbox width=forced center]{forced center with long text}\\
\tcbox[tcbox width=forced left]{forced left}\qquad
\tcbox[tcbox width=forced left]{forced left with long text}\\
\tcbox[tcbox width=forced right]{forced right}\qquad
\tcbox[tcbox width=forced right]{forced right with long text}\\
\tcbox[tcbox width=minimum center]{minimum center}\qquad
\tcbox[tcbox width=minimum center]{minimum center with long text}\\
\tcbox[tcbox width=minimum left]{minimum left}\qquad
\tcbox[tcbox width=minimum left]{minimum left with long text}\\
\tcbox[tcbox width=minimum right]{minimum right}\qquad
\tcbox[tcbox width=minimum right]{minimum right with long text}
```



4.21 Counters, Labels, and References

/tcb/phantom=⟨*code*⟩ (no default, initially unset)

The ⟨*code*⟩ is put in a box at the upper left corner of the `tcolorbox`. If the `tcolorbox` is breakable, the ⟨*code*⟩ is executed for the first box of the break sequence only. If there already was some phantom code given, the new ⟨*code*⟩ is appended.

The ⟨*code*⟩ is intended to be used for counter stepping, labelling, and related operations which do not produce visible text.

- The ⟨*code*⟩ is executed before the title and box content, i. e. counter values are ensured to be increased before usage.
- Labels are ensured to reference the correct page number.
- The ⟨*code*⟩ is executed only once even during fitting operations for title and box content.
- In combination with the `hyperref` package, the hyper anchor is set to the upper left corner of the `tcolorbox`, i. e. links inside the pdf document will jump to the box pleasantly.
- Since the ⟨*code*⟩ is executed inside a TEX group, only global operations can survive this group.

Examples for the `phantom` usage are given in Section 15.9 from page 320, e. g. Example 15.1 on page 321.

/tcb/nophantom (no value, initially set)

Removes the phantom code if set before.

/tcb/label=⟨*marker*⟩ (no default, initially unset)

The ⟨*marker*⟩ is set as label text for a reference with the `\ref` macro. Typically, this option is used for numbered boxes, see Subsection 5.1 from page 100, e. g. `/tcb/new/auto counter` → P. 100.

N 2014-11-28 **/tcb/phantomlabel**=⟨*marker*⟩ (no default, initially unset)

Equivalent to `/tcb/label` for an *unnumbered* box. A `\phantomsection` from the package `hyperref` is used to set a correct hyperlink target. This is not needed for a numbered box.

/tcb/label type=⟨*type*⟩ (no default, initially unset)

This option key can be used only in conjunction with the `cleveref` package [5] which has to be loaded separately. ⟨*type*⟩ has to be a cross-reference type *known* to `cleveref` like `theorem`, `algorithm`, `result`, etc. References made with `cleveref` will use this type. Note that using `label type` will result in compilation errors, if `cleveref` is not loaded. For an example, see Theorem 16.3.5 on page 345.

/tcb/no label type (no value, initially set)

Removes a `/tcb/label type`, if set before.

/tcb/step=⟨*counter*⟩ (no default, initially unset)

Shortcut for `phantom={\refstepcounter{#1}}`. The given ⟨*counter*⟩ is increased and ready for labelling. This option is not needed when using the convenient automated numbering introduced with version 2.40, see Subsection 5.1 from page 100.

/tcb/step and label={⟨*counter*⟩}{⟨*marker*⟩} (no default, initially unset)

Shortcut for using `/tcb/step` and `/tcb/label`. This option is not needed when using the convenient automated numbering introduced with version 2.40, see Subsection 5.1 from page 100.

`/tcb/list entry=`⟨*text*⟩ (no default, initially unset)

If the «list of tcolorbox(es)» feature described in Subsection 5.2 from page 107 is used, this key describes the ⟨*text*⟩ for an entry into the generated list, e. g.

```
list entry={\protect\numberline{\thetcbcounter}My beautiful Example}
```

See Section 15.9 from page 320 for a complete example.

N 2014-09-19 `/tcb/list text=`⟨*text*⟩ (style, no default)

This is a shortcut for setting `/tcb/list entry` to `\protect\numberline{\thetcbcounter}`⟨*text*⟩. So, the following settings are identical:

```
list text={My beautiful Example},
list entry={\protect\numberline{\thetcbcounter}My beautiful Example}
```

See Section 15.9 from page 320 for a complete example.

`/tcb/add to list={`⟨*list*⟩`}{`⟨*type*⟩`}` (no default, initially unset)

If the «list of tcolorbox(es)» feature described in Subsection 5.2 from page 107 is used, list entries are generated automatically. With this key, you can enforce an entry to the given ⟨*list*⟩ with the given ⟨*type*⟩. This issues:
`\addcontentsline{`⟨*list*⟩`}{`⟨*type*⟩`}{`⟨*entry text*⟩`}`

4.22 Even and Odd Pages

! Also see `/tcb/toggle left and right` → P. 45 and `/tcb/toggle enlargement` → P. 84 for further even/odd options.

U 2015-11-13 `/tcb/check odd page=true|false` (default `true`, initially `false`)

If set to `true`, a precise even/odd page testing for the current box is applied. This is done by using labels. If a box moves to another page, the document has to be compiled twice for the correct settings. If set to `false`, even/odd page tests may give wrong results for the first box of a page.
`/tcb/toggle left and right` → P. 45, `/tcb/toggle enlargement` → P. 84, and `/tcb/if odd page` automatically set `check odd page`, but for `\tcbifoddpage` → P. 96 this option has to be set explicitely.

N 2015-11-13 `/tcb/if odd page={`⟨*odd options*⟩`}{`⟨*even options*⟩`}` (style, no default)

If the current box is on an odd page, the ⟨*odd options*⟩ are applied. On an even page, the ⟨*even options*⟩ are applied. `/tcb/check odd page` is automatically set for precise even/odd page testing.

```
\begin{tcolorbox}[if odd page={colback=yellow!50}{colback=red!50}]
  This box is colored in yellow on an odd page
  and is colored in red on an even page.
\end{tcolorbox}
```

This box is colored in yellow on an odd page and is colored in red on an even page.

! If a box is `/tcb/breakable` → P. 351, using `/tcb/if odd page` only acts upon the *first* box. If the setting should be repeated for every partial box of the break sequence, the option should be packed into `/tcb/extras` → P. 357. In this case, `/tcb/check odd page` has to be set explicitely! Also see `/tcb/if odd page*` → P. 96.

N 2015-11-13 **/tcb/if odd page*=**{⟨*odd options*⟩}{⟨*even options*⟩} (style, no default)

> ! This option needs the LIB breakable library, see Section 17 on page 349.

For breakable boxes, if the current partial box is on an odd page, the ⟨*odd options*⟩ are applied. On an even page, the ⟨*even options*⟩ are applied. /tcb/check odd page →P. 95 is automatically set for precise even/odd page testing.
In contrast to /tcb/if odd page →P. 95, /tcb/if odd page* is used on *every* partial box of a break sequences and not only on the *first* box. Another difference is that /tcb/if odd page* is applied quite *late* during option processing, while /tcb/if odd page →P. 95 is applied immediately.
/tcb/if odd page* is implemented as /tcb/if odd page →P. 95 packed into /tcb/extras →P. 357.

```
% \tcbuselibrary{breakable}
\begin{tcolorbox}[breakable,if odd page*={colback=yellow!50}{colback=red!50}]
  This breakable box is colored in yellow on an odd page
  and is colored in red on an even page. For every partial box, the
  test is repeated, i.e. this would give a yellow, red, yellow, red, \ldots\
  sequence for a long content.
\end{tcolorbox}
```

This breakable box is colored in yellow on an odd page and is colored in red on an even page. For every partial box, the test is repeated, i.e. this would give a yellow, red, yellow, red, . . . sequence for a long content.

N 2015-11-13 **\tcbifoddpage**{⟨*odd code*⟩}{⟨*even code*⟩}

If the current box is on an odd page, the ⟨*odd code*⟩ is executed. On an even page, the ⟨*even code*⟩ is executed. For precise even/odd page testing, the /tcb/check odd page →P. 95 has to be set manually inside the box options.
The macro \tcbifoddpage can be used inside underlay, overlay, or watermark code to test if the box is on an odd page. This will work also for boxes in a break sequence.
The macro can also be used inside the box **content text**. For unbreakable boxes, the correct page test is applied. But for /tcb/breakable →P. 351 boxes, \tcbifoddpage will always give the result for the page of the *first* box inside the box **content text**. If needed, the methods from the packages `changepage` or `ifoddpage` could be used here.

```
\tcbset{colframe=blue!75!black,colback=white,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,check odd page,
  title={Example for a box on an \tcbifoddpage{odd}{even} page},
  watermark text={\tcbifoddpage{Odd}{Even} page!}]
\lipsum[1]
\end{tcolorbox}
```

Example for a box on an even page

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Even page!

N 2015-11-13 **\thetcolorboxnumber**

This is a unique identifier (arabic number) for a tcolorbox. It is locally defined inside boxes and has no meaning outside. It is used for precise even/odd page testing, but may also be valuable for elaborate user code.

```
\begin{tcolorbox}[colback=yellow!5,title=Box \thetcolorboxnumber]
  This box is \thetcolorboxnumber.
  \tcbox[on line,size=fbox]{This box is \thetcolorboxnumber} and
  \tcbox[on line,size=fbox]{this box is \thetcolorboxnumber}.
  This box is \thetcolorboxnumber.
\end{tcolorbox}
```

Box 883

This box is 883. This box is 884 and this box is 885. This box is 883.

N 2015-11-13 **\thetcolorboxpage**

This macro contains the expanded arabic page number of the current tcolorbox. It is locally defined inside boxes and has no meaning outside. It is precise only, if `/tcb/check odd page` → P. 95 was set.

```
\begin{tcolorbox}[colback=yellow!5,check odd page,
    title=Box on page~\thetcolorboxpage]
  This box is located on page~\thetcolorboxpage.
\end{tcolorbox}
```

Box on page 97

This box is located on page 97.

4.23 Externalization

See Section 22 on page 415 for the LIB external library of tcolorbox.

If the *externalization* library of the tikz package is used and /tcb/graphical environment →P. 125 is set to tikzpicture, a tcolorbox could trigger the externalization process which will arise a compilation error.

To avoid this, there are two possible strategies:

- Ensure, that \tikzexternaldisable is set before a tcolorbox is used. If you typically use the pattern \tikzexternalenable *some picture* \tikzexternaldisable, there is nothing to care about.
- If *externalization* is enabled globally, use /tcb/shield externalize to shield any tcolorbox. The preamble code could look like this:

```
\usetikzlibrary{external}
\tikzexternalize
\tcbset{shield externalize}
```

/tcb/shield externalize=true|false (default true, initially false)

If set to true, the drawing part of the tcolorbox is not being externalized which is a good thing at the current state of art. Nevertheless, if the tcolorbox contains a tikzpicture, this picture is still externalized. Pictures drawn with help of /tcb/tikz upper →P. 67 or alike are *not* externalized.

If a tcolorbox is used inside a node of an encircling tikzpicture which is externalized, do *not* use \tikzexternaldisable in front of the tcolorbox. /tcb/shield externalize is deactivated automatically inside a tikzpicture.

/tcb/external=⟨*file name*⟩ (no default, initially unset)

Convenience option which calls \tikzsetnextfilename{⟨*file name*⟩}. Typically, it may be used inside the option list of a tcolorbox to set the externalization ⟨*file name*⟩ for the first tikzpicture which is discovered *inside* the box content. The package tikz [20] or the library LIB skins has to be loaded to use this option. Additionally, \usetikzlibrary{external} has to be used.

/tcb/remake=true|false (default true, initially false)

Convenience option which calls /tikz/external/remake next. Typically, it may be used inside the option list of a tcolorbox to force the remake of the first tikzpicture which is discovered *inside* the box content. The package tikz [20] or the library LIB skins has to be loaded to use this option. Additionally, \usetikzlibrary{external} has to be used.

4.24 Miscellaneous

/tcb/reset (no value, initially set)

Sets (nearly) all tcolorbox settings (including loaded libraries) back to their default values *plus* any settings given by \tcbsetforeverylayer →P. 13. /tcb/savedelimiter →P. 26 and /tcb/capture →P. 90 keep their values. Also, all raster values (see Section 14 on page 265) are not resetted.

This option is useful for boxes in boxes where the inner box should not inherit the settings of the outer box. Note that for boxes inside boxes the reset is done automatically, if the standard settings of the package are used (v2.40 and above), see Section 4.16 from page 87.

N 2015-01-09 **/tcb/only=<⟨*overlay specification*⟩>{⟨*options*⟩}** (style, no default, initially unset)

Sets the given `tcolorbox` ⟨*options*⟩ in dependency of a `beamer` ⟨*overlay specification*⟩. Note that this needs the `beamer` class. The ⟨*options*⟩ will only be used on the specified `beamer` frames.

```
\documentclass{beamer}
\usepackage[many]{tcolorbox}
\begin{document}

\begin{frame}
  \begin{tcolorbox}[title=My title,fonttitle=\bfseries,
      enhanced,colframe=red!50!black,colback=red!10,colbacktitle=red,
      sidebyside,righthand width=3cm,
      lowerbox=invisible,lower separated=false,
      drop lifted shadow,
      only=<1>{colbacktitle=yellow,coltitle=red!50!black,colframe=red},
      only=<3>{colback=yellow!50,watermark text={Attention!}},
      only=<3->{lowerbox=visible}  ]
    This is a test.
    \begin{itemize}[<+->]
      \item One
      \item Two
      \item \alert<3>{Three}
      \item Four
    \end{itemize}
  \tcblower
    \begin{equation*}
    \int\limits_{1}^{x} \frac{1}{t}~dt = \ln(x).
    \end{equation*}
  \end{tcolorbox}
\end{frame}

\end{document}
```

/tcb/code=⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is executed immediately. This option is useful to place some arbitrary code into an option list.

```
\tcbset{colback=red!5!white,colframe=red!75!black,
  code={Useless at this spot but functional.},
  fonttitle=\bfseries}

\begin{tcolorbox}[code={\newcommand{\mycommand}{\textit{working}}},
  title=My \mycommand\ title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

Useless at this spot but functional.

My *working* title

This is a **tcolorbox**.

5 Initialization Option Keys

The *initialization* options are only applicable for the generation of new environments and commands based on `tcolorbox` and friends. Particularly, they can be used for

- `\newtcolorbox` → P. 15,
- `\newtcbox` → P. 16,
- `\newtcblisting` → P. 290,
- `\newtcbinputlisting` → P. 292,
- `\newtcbtheorem` → P. 326, and
- `\newtcboxfit` → P. 381.

! Typically, these options may generate counters and alike. It is **strongly** recommended that you use initialization options inside the preamble only. Otherwise, you may get trouble when using LaTeX's `\include` features.

5.1 Numbered Boxes

Counters assigned using the initialization options are administrated automatically. Especially, they are increased for each new box. Independent from the real counter name, the counter value can be referenced by `\thetcbcounter`, e. g. inside the title of the box. The real counter name is stored inside `\tcbcounter`.

`/tcb/new/auto counter` (no value, initially unset)

Creates a new counter automatically. With `/tcb/new/number format` → P. 102 and `/tcb/new/number within` → P. 102, the appearance and behavior of the counter can be changed. The counter value is referenced by `\thetcbcounter`.

Definition in the preamble:

```
\newtcolorbox[auto counter,number within=section]{pabox}[2][]{%
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  title=Examp.~\thetcbcounter: #2,#1}
```

```
\begin{pabox}[label={myautocounter}]{Title with number}
This box is automatically numbered with \ref{myautocounter} on page
\pageref{myautocounter}. Inside the box, the \thetcbcounter\ can
also be referenced by |\thetcbcounter|.
The real counter name is \texttt{\tcbcounter}.
\end{pabox}
```

Examp. 5.1: Title with number

This box is automatically numbered with 5.1 on page 100. Inside the box, the 5.1 can also be referenced by `\thetcbcounter`. The real counter name is `tcb@cnt@pabox`.

`/tcb/new/use counter from=`⟨*tcolorbox*⟩ (no default, initially unset)

Here, a counter from another ⟨*tcolorbox*⟩ is reused. Note that the settings for `/tcb/new/number format` → P. 102 and `/tcb/new/number within` → P. 102 are inherited and cannot be changed. The counter value is referenced by `\thetcbcounter`.

```
\newtcolorbox[use counter from=pabox]{mybox}[2][]{%
colback=blue!5!white,colframe=blue!75!black,fonttitle=\bfseries,
title=Some Box \thetcbcounter: #2,#1}

\begin{mybox}[label={myusecounterfrom}]{Title with continued number}
This box is automatically numbered with \ref{myusecounterfrom} on page
\pageref{myusecounterfrom}. Inside the box, the \thetcbcounter\ can
also be referenced by |\thetcbcounter|.
The real counter name is \texttt{\tcbcounter}.
\end{mybox}
```

Some Box 5.2: Title with continued number

This box is automatically numbered with 5.2 on page 101. Inside the box, the 5.2 can also be referenced by `\thetcbcounter`. The real counter name is `tcb@cnt@pabox`.

`/tcb/new/use counter=`⟨*counter*⟩ (no default, initially unset)

Here, an ordinary existing LaTeX ⟨*counter*⟩ is used for numbering. With `/tcb/new/number format` → P. 102 and `/tcb/new/number within` → P. 102, the appearance and behavior of the counter can be changed. The counter value is referenced by `\thetcbcounter`.

```
% \newcounter{myexample}%  preamble
\newtcolorbox[use counter=myexample,number format=\Alph]{mybox}[2][]{%
colback=green!5!white,colframe=green!55!black,fonttitle=\bfseries,
title=Some Box \thetcbcounter: #2,#1}

\begin{mybox}[label={myusecounter}]{Title with \LaTeX\ number}
This box is automatically numbered with \ref{myusecounter} on page
\pageref{myusecounter}. Inside the box, the \thetcbcounter\ can
also be referenced by |\thetcbcounter|.
The real counter name is \texttt{\tcbcounter}.
\end{mybox}
```

Some Box A: Title with LaTeX number

This box is automatically numbered with A on page 101. Inside the box, the A can also be referenced by `\thetcbcounter`. The real counter name is `myexample`.

N 2014-09-19 `/tcb/new/use counter*=`⟨*counter*⟩ (no default, initially unset)

An existing LaTeX ⟨*counter*⟩ is used for numbering. In contrast to `/tcb/new/use counter`, the options `/tcb/new/number format` → P. 102 and `/tcb/new/number within` → P. 102 are ignored. Use this for counters which are already configured outside the `tcolorbox` package, e. g. the standard `figure` counter.

`/tcb/new/no counter` (no value, initially set)

The created boxes are not numbered. This is the default. The option may be used to overrule a previous option.

`/tcb/new/number within`=⟨*counter*⟩ (no default, initially unset)

The automatic counter is set to zero, if ⟨*counter*⟩ is increased. Additionally, during output, the value of ⟨*counter*⟩ is prepended to the value of the automatic counter.

To prepend the automatic counter with the chapter number and to reset it with every new chapter, use:

```
number within=chapter
```

See `/tcb/new/use counter` → P. 101 for a complete example.

`/tcb/new/number format`=⟨*format macro*⟩ (no default, initially `\arabic`)

Declares the format of the automatic counter. The ⟨*format macro*⟩ can be any valid LaTeX number formatting macro like `\arabic`, `\roman`, etc.

To display the counter value in large roman numbers, use:

```
number format=\Roman
```

See `/tcb/new/auto counter` → P. 100 for a complete example.

`/tcb/new/number freestyle`=⟨*code*⟩ (no default, initially unset)

Allows advanced control over the complete number format. This option overrules the format given by `/tcb/new/number within` and `/tcb/new/number format`. Nevertheless, you can combine it with `/tcb/new/number within` to get the desired reset property.

The ⟨*code*⟩ is some formatting code which should contain `\tcbcounter` to reference the automated counter. Since this ⟨*code*⟩ is expanded, you have to secure each macro with `\noexpand` with *exception* of `\tcbcounter`.

Definition in the preamble:

```
\newtcolorbox[auto counter,number within=section,
  number freestyle={(Q/\noexpand\thesection/\noexpand\Alph{\tcbcounter})},
  ]{phbox}[2][]{%
  colback=yellow!15!white,colframe=blue!75!black,fonttitle=\bfseries,
  title=Question~\thetcbcounter: #2,#1}
```

```
\begin{phbox}[label={myfreestyle}]{Title with freestyle number}
This box is automatically numbered with \ref{myfreestyle} on page
\pageref{myfreestyle}. Inside the box, the \thetcbcounter\ can
also be referenced by |\thetcbcounter|.
The real counter name is \texttt{\tcbcounter}.
\end{phbox}
```

Question (Q/5/A): Title with freestyle number

This box is automatically numbered with (Q/5/A) on page 102. Inside the box, the (Q/5/A) can also be referenced by `\thetcbcounter`. The real counter name is `tcb@cnt@phbox`.

The following options /tcb/new/crefname and /tcb/new/Crefname need to be set inside the preamble.

U 2014-12-01 **/tcb/new/crefname**={⟨*singular*⟩}{⟨*plural*⟩} (no default, initially unset)

This option key can be used only in conjunction with the `cleveref` package [5] which has to be loaded separately. It creates a cross-reference type for the new `tcolorbox`'es, where the lowercase ⟨*singular*⟩ and ⟨*plural*⟩ forms of the cross-reference are given. This type is the environment or macro name and /tcb/label type → P. 94 is set automatically. See /tcb/label type → P. 94 and [5] for more information.

U 2014-12-01 **/tcb/new/Crefname**={⟨*singular*⟩}{⟨*plural*⟩} (no default, initially unset)

This option key can be used only in conjunction with the `cleveref` package [5] which has to be loaded separately. It creates a cross-reference type for the new `tcolorbox`'es, where the uppercase ⟨*singular*⟩ and ⟨*plural*⟩ forms of the cross-reference are given. This type is the environment or macro name and /tcb/label type → P. 94 is set automatically. See /tcb/label type → P. 94 and [5] for more information.

Definition in the preamble:

```
% \usepackage{cleveref}
\newtcolorbox[auto counter,number within=section,
  crefname={bluebox}{blueboxes}]%
  {mybluebox}[2][]{colback=blue!5!white,colframe=blue!75!black,fonttitle=\bfseries,
     title=Bluebox \thetcbcounter: #2,#1}
```

```
% \usepackage{varioref}
% \usepackage{cleveref}
\begin{mybluebox}[label={myreference}]{My title}
This is an example.
\end{mybluebox}

\Cref{myreference}, \cref{myreference}.\\
\Cpageref{myreference}, \cpageref{myreference}.\\
\nameCref{myreference}, \namecref{myreference}.\\
\labelcref{myreference}, \labelcpageref{myreference}.\\
With \texttt{varioref}:\\
\Vref{myreference}, \vref{myreference}.\\
\Vref*{myreference}, \vref*{myreference}.
```

Bluebox 5.1: My title

This is an example.

Bluebox 5.1, bluebox 5.1.
Page 103, page 103.
Bluebox, bluebox.
5.1, 103.
With `varioref`:
Bluebox 5.1, bluebox 5.1.
Bluebox 5.1, bluebox 5.1.

N 2014-09-19 **/tcb/new/blend into=**⟨*name*⟩ (style, no default, initially unset)

Used to comfortably blend into an existing schema of naming and numbering for some selected cases. For example, a `tcolorbox` can be used to display and entitle an image pretending to be a standard `figure` environment. Here, `/tcb/title` → P. 18 is used instead of the standard `\caption` and `/tcb/list text` → P. 95 can be used instead of the optional parameter of the standard `\caption`.

Feasible values for ⟨*name*⟩ are:

- `figures`: blend into the standard `figure` environment.
- `tables`: blend into the standard `table` environment.
- `listings`: blend into the standard `lstlisting` environment of the package `listings` [6].

> **!** Note that `blend into=listings` can only be used in the document content or, preferably, inside a `\AtBeginDocument` clause! Using it without `\AtBeginDocument` inside the preamble does not work since the `listings` packages initializes its counter also inside `\AtBeginDocument`.

```
\begin{figure}[htb]
  \centering\includegraphics[height=4cm]{lichtspiel.jpg}
  \caption{A standard figure}
\end{figure}

\newtcolorbox[blend into=figures]{myfigure}[2][]{float=htb,capture=hbox,
  title={#2},every float=\centering,#1}

\begin{myfigure}{A tcolorbox figure}
  \includegraphics[height=4cm]{lichtspiel.jpg}
\end{myfigure}
```

Figure 1: A standard figure

Figure 2: A tcolorbox figure

N 2015-03-13 **/tcb/blend before title=**⟨*value*⟩ (no default, initially **colon**)

This option formats the title output of **/tcb/new/blend into** → P. 104. Note that this is a common **tcolorbox** option which should be set globally or in the normal option part of **\newtcolorbox** → P. 15.

Feasible values for ⟨*value*⟩ are:

- **colon**: use name/number plus colon.
- **dash**: use name/number plus dash.
- **colon hang**: use name/number plus colon with hanging indent.
- **dash hang**: use name/number plus dash with hanging indent.

```
\newtcolorbox[blend into=figures]{myfigure}[2][]{float=htb,capture=hbox,
  blend before title=dash hang,title={#2},every float=\centering,#1}

\begin{myfigure}{A tcolorbox figure with quite a long title}
  \includegraphics[height=5cm]{lichtspiel.jpg}
\end{myfigure}
```

Figure 3 – A tcolorbox figure with quite a long title

N 2015-03-13 **/tcb/blend before title code=**⟨*code*⟩ (no default)

This option formats the title output of /tcb/new/blend into → P. 104. The ⟨*code*⟩ takes one parameter, the name/number. Use this, if /tcb/blend before title → P. 105 is not flexible enough.

```
\newtcolorbox[blend into=figures]{myfigure}[2][]{float=htb,capture=hbox,
  blend before title code={\fbox{##1}\ },title={#2},every float=\centering,#1}

\begin{myfigure}{A tcolorbox figure}
  \includegraphics[height=6cm]{lichtspiel.jpg}
\end{myfigure}
```

Figure 4 A tcolorbox figure

5.2 Lists of tcolorboxes

For figures and tables, LaTeX provides the `\listoffigures` and `\listoftables` commands to create lists of these numbered entities. Also, a `tcolorbox` can be part of such a kind of list.

1. Assign a list ⟨*name*⟩ by the *initialization* option `/tcb/new/list inside`.
2. Optionally, a new ⟨*type*⟩ for list entries may be assigned by the *initialization* option `/tcb/new/list type`.
3. List entries a generated automatically within each new `tcolorbox` using the above initialization.
   - If `/tcb/list entry` → P. 95 is set, the entry is generated with it.
   - Otherwise, if `/tcb/title` → P. 18 is set, the entry is generated with it.
   - Otherwise, the entry is generated with the current number and the environment name.
4. The generated list is displayed by `\tcblistof`.

`/tcb/new/list inside`=⟨*name*⟩ (no default, initially unset)

Assigns a list or contents file to the generated `tcolorbox`es. Entries to this list are saved to a file which gets the ⟨*name*⟩ as file name extension. The list is referenced by this name in `\tcblistof`. For example:

```
list inside=exam
```

See Section 15.9 from page 320 for a complete example.

`/tcb/new/list type`=⟨*type*⟩ (no default, initially `tcolorbox`)

Optionally, some ⟨*type*⟩ can be assigned to the list entries. For a new ⟨*type*⟩, a macro `\l@`⟨*type*⟩ has to exist which controls the format of the list entry. The default type is defined by

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

This is identical to the `\l@section` setting of LaTeX. `\l@tcolorbox` can be redefined or a new ⟨*type*⟩ can be assigned.

`\tcblistof`[⟨*macro*⟩]{⟨*name*⟩}{⟨*title text*⟩}

Displays the generated list of `tcolorbox`es with the given ⟨*name*⟩. The heading is generated by ⟨*macro*⟩{⟨*title text*⟩} where `\section` is the default setting for ⟨*macro*⟩.
To display the list inside a subsection, use for example:

```
\tcblistof[\subsection]{exam}{List of Exercises}
```

The result of the example is found as Subsection 15.10 on page 323.

The core of the list is generated by `\@starttoc{`⟨*name*⟩`}` which can be wrapped into an own macro.

6 Side by Side

A *side by side* box is a special `tcolorbox` → P. 12 where the upper and lower part of the box are set side by side. All boxes of this kind are unbreakable.

> **!** Further side by side options for code examples are `/tcb/listing side text` → P. 303, `/tcb/text side listing` → P. 303, `/tcb/listing outside text` → P. 303, and `/tcb/text outside listing` → P. 303.

6.1 Basic Settings

`/tcb/sidebyside=true|false` (default `true`, initially `false`)

Normally, the upper part and the lower part of the box have their positions as their names suggest. If `sidebyside` is set to `true`, the upper part is drawn *left-handed* and the lower part is drawn *right-handed*. Both parts are drawn together with the geometry settings of the upper part but the space is divided horizontally according to the following options. Colors, fonts, and box content additions are used individually. The resulting box is unbreakable.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title,sidebyside]
  This is the upper (\textit{left-handed}) part.
\tcblower
  This is the lower (\textit{right-handed}) part.
\end{tcolorbox}
```

My title

This is the upper (*left-handed*) part.

This is the lower (*right-handed*) part.

```
% \usepackage{lipsum}
% \tcbuselibrary{skins}
\begin{tcolorbox}[bicolor,sidebyside,righthand width=3cm,
    sharp corners,boxrule=.4pt,colback=green!5,colbacklower=green!50!black!50]
  \lipsum[2]
\tcblower
  \includegraphics[width=\linewidth]{goldshade}%
\end{tcolorbox}
```

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

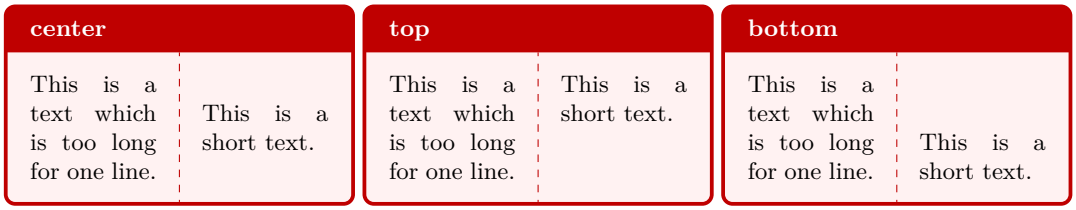
U 2015-02-06 **/tcb/sidebyside align=**⟨*alignment*⟩ (no default, initially **center**)

Sets the vertical ⟨*alignment*⟩ for the left-handed and right-handed part.

Feasible values for ⟨*alignment*⟩ are:

- **center**: identical to **minipage** option **c**.
- **top**: identical to **minipage** option **t** (aligns the top lines of the left-handed and right-handed side according to their baselines).
- **bottom**: identical to **minipage** option **b** (aligns the bottom lines of the left-handed and right-handed side according to their baselines).
- **center seam**: aligns the center of the left-handed and right-handed side.
- **top seam**: aligns the very top seam of the left-handed and right-handed side.
- **bottom seam**: aligns the very bottom seam of the left-handed and right-handed side.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,nobeforeafter,
  left=2mm,right=2mm,sidebyside,sidebyside gap=6mm,width=(\linewidth-2mm)/3}

\begin{tcolorbox}[adjusted title=center,sidebyside align=center]
This is a text which is too long for one line.
\tcblower
This is a short text.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=top,sidebyside align=top]
This is a text which is too long for one line.
\tcblower
This is a short text.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=bottom,sidebyside align=bottom]
This is a text which is too long for one line.
\tcblower
This is a short text.
\end{tcolorbox}
```



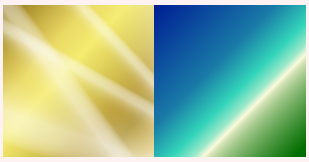
center, **top**, and **bottom** are identical to the known corresponding **minipage** options. While this is the preferred approach for text content, the result for boxed content like tables or images may not be as expected.

For such content, one may use **center seam**, **top seam**, and **bottom seam**. For example, **top seam** aligns the very top seam of the left-handed and right-handed side.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  size=small,righthand width=4cm,sidebyside,sidebyside gap=6mm,lower separated=false}

\begin{tcolorbox}[adjusted title=center seam,sidebyside align=center seam]
  This is my description text for the pictures displayed on the right-handed side.
  \tcblower
  \includegraphics[width=\linewidth/2]{goldshade}%
  \includegraphics[width=\linewidth/2]{blueshade}
\end{tcolorbox}

\begin{tcolorbox}[adjusted title=top seam,sidebyside align=top seam]
  This is my description text for the pictures displayed on the right-handed side.
  \tcblower
  \includegraphics[width=\linewidth/2]{goldshade}%
  \includegraphics[width=\linewidth/2]{blueshade}
\end{tcolorbox}

\begin{tcolorbox}[adjusted title=bottom seam,sidebyside align=bottom seam]
  This is my description text for the pictures displayed on the right-handed side.
  \tcblower
  \includegraphics[width=\linewidth/2]{goldshade}%
  \includegraphics[width=\linewidth/2]{blueshade}
\end{tcolorbox}
```

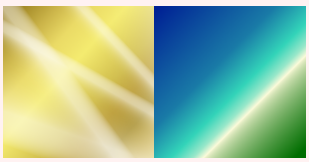
center seam

This is my description text for the pictures displayed on the right-handed side.



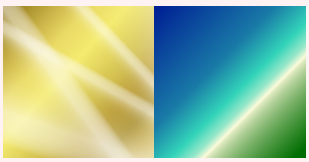
top seam

This is my description text for the pictures displayed on the right-handed side.



bottom seam

This is my description text for the pictures displayed on the right-handed side.



/tcb/sidebyside gap=⟨*length*⟩ (no default, initially 10mm)

Sets the horizontal distance between the left-handed and right-handed part to ⟨*length*⟩.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,nobeforeafter,
  sidebyside,width=(\linewidth-2mm)/2}

\begin{tcolorbox}[adjusted title=Wide gap,sidebyside gap=30mm]
This is a text which is too long for one line.
\tcblower
This is a short text.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=Narrow gap,sidebyside gap=1mm]
This is a text which is too long for one line.
\tcblower
This is a short text.
\end{tcolorbox}
```

Wide gap

This is a text which is too long for one line.

This is a short text.

Narrow gap

This is a text which is too long for one line.

This is a short text.

/tcb/lefthand width=⟨*length*⟩ (no default, initially unset)

Sets the width of the left-handed part to the given ⟨*length*⟩.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title,sidebyside,lefthand width=3cm]
This is the upper (\textit{left-handed}) part.
\tcblower
This is the lower (\textit{right-handed}) part.
\end{tcolorbox}
```

My title

This is the upper (*left-handed*) part.

This is the lower (*right-handed*) part.

/tcb/righthand width=⟨*length*⟩ (no default, initially unset)

Sets the width of the right-handed part to the given ⟨*length*⟩.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title,sidebyside,righthand width=3cm]
This is the upper (\textit{left-handed}) part.
\tcblower
This is the lower (\textit{right-handed}) part.
\end{tcolorbox}
```

My title

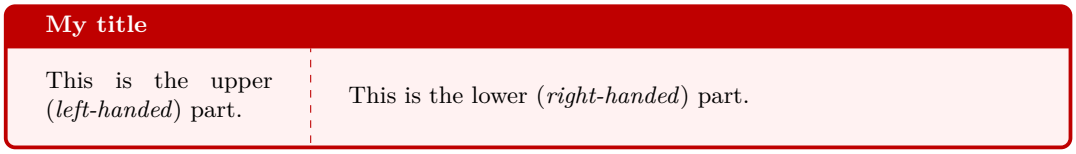
This is the upper (*left-handed*) part.

This is the lower (*right-handed*) part.

/tcb/lefthand ratio=⟨*fraction*⟩ (no default, initially 0.5)

Sets the width of the left-handed part to the given ⟨*fraction*⟩ of the available space. ⟨*fraction*⟩ is a value between 0 and 1.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title,sidebyside,lefthand ratio=0.25]
This is the upper (\textit{left-handed}) part.
\tcblower
This is the lower (\textit{right-handed}) part.
\end{tcolorbox}
```



/tcb/righthand ratio=⟨*fraction*⟩ (no default, initially 0.5)

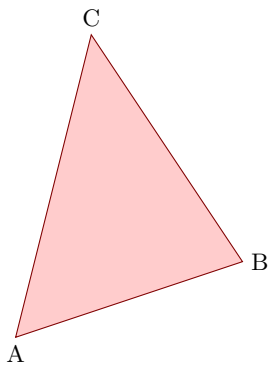
Sets the width of the right-handed part to the given ⟨*fraction*⟩ of the available space. ⟨*fraction*⟩ is a value between 0 and 1.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title,sidebyside,righthand ratio=0.25]
This is the upper (\textit{left-handed}) part.
\tcblower
This is the lower (\textit{right-handed}) part.
\end{tcolorbox}
```

If one side of a side-by-side box should be adapted to the width of its content, this width has to be computed beforehand. The following example uses a savebox `\mysavebox` to store the picture to determine its width. A more convenient way to handle this task is to use the methods from Section 6.2 on page 114.

```
% \tcbuselibrary{skins,xparse}
% \usepackage{lipsum}
% \newsavebox\mysavebox  % preamble
\DeclareTotalTColorBox{\mysidebox}{ O{} +m +m }{
  bicolor,colback=white,colbacklower=yellow!10,
  fonttitle=\bfseries,center title,
  sidebyside,
  code={\sbox{\mysavebox}{#2}},
  lefthand width=\wd\mysavebox,
  drop lifted shadow,
  #1
}
{\usebox{\mysavebox}\tcblower#3}

\mysidebox[title=The Triangle]{%
  \begin{tikzpicture}
    \path[fill=red!20,draw=red!50!black]
      (0,0) node[below]{A} -- (3,1) node[right]{B}
      -- (1,4) node[above]{C} -- cycle;
  \end{tikzpicture}%
}{%
  \lipsum[1]
}
```

The Triangle



Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

6.2 Advanced Settings from the LIB xparse Library

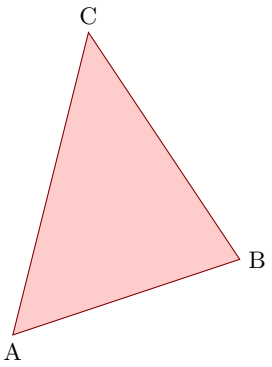
> ! All following macros and options need the LIB xparse library to be loaded, see Section 21 on page 402.

N 2015-11-20 `\tcbsidebyside[⟨options⟩]{⟨left-handed content⟩}{⟨right-handed content⟩}`

Creates a colored box using more or less arbitrary ⟨*options*⟩ for a `tcolorbox` → P. 12. The `/tcb/sidebyside` → P. 108 option is set to `true` and the ⟨*left-handed content*⟩ and ⟨*right-handed content*⟩ is filled into the box appropriately. The resulting box is unbreakable. `\tcbsidebyside` is not only a shortcut for using a normal `tcolorbox` → P. 12 with `/tcb/sidebyside` → P. 108, but allows setting further options like `/tcb/sidebyside adapt` → P. 115 and `/tcb/sidebyside switch` → P. 117.

```
% \tcbuselibrary{skins,xparse}
% \usepackage{lipsum}
\tcbsidebyside[title=The Triangle,
  sidebyside adapt=left,
  bicolor,colback=white,colbacklower=yellow!10,
  fonttitle=\bfseries,center title,drop lifted shadow,
]{%
  \begin{tikzpicture}
    \path[fill=red!20,draw=red!50!black]
      (0,0) node[below]{A} -- (3,1) node[right]{B}
      -- (1,4) node[above]{C} -- cycle;
  \end{tikzpicture}%
}{%
  \lipsum[1]
}
```

The Triangle



Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

N 2015-11-20 **/tcb/sidebyside adapt=**⟨*side(s)*⟩ (no default, initially **none**)

The option allows the left-handed and/or right-handed side to determine the dimensions of the box. This option is only valid inside \tcbsidebyside → P. 114.

Feasible values for ⟨*side(s)*⟩ are:

- **none**: no measurement of left-handed and right-handed side.
- **left**: the actual width of the left-handed content is used to set /tcb/lefthand width → P. 111.
- **right**: the actual width of the right-handed content is used to set /tcb/righthand width → P. 111.
- **both**: the actual width of the left-handed and right-handed content is used to set /tcb/lefthand width → P. 111, /tcb/righthand width → P. 111, and the overall /tcb/width → P. 34.

```
% \tcbuselibrary{skins,xparse}
\tcbsidebyside[sidebyside adapt=left,
  title=Very important table,
  beamer,colframe=blue!50!black,colback=blue!10,
  lower separated=false,sidebyside gap=5mm
]{%
  \begin{tabular}{|l|c|r|}\hline
    left & center & right\\\hline
    A & B & C\\\hline
    D & E & F\\\hline
  \end{tabular}
}{%
  This table contains the most important figures for
  all future actions. You may notice that B follows A,
  C follows B, and so on.
}
```

Very important table

left	center	right
A	B	C
D	E	F

This table contains the most important figures for all future actions. You may notice that B follows A, C follows B, and so on.

```
% \tcbuselibrary{skins,xparse}
\tcbsidebyside[sidebyside adapt=right,
  blanker,sidebyside gap=5mm
]{%
  \lipsum[2]
}{%
\begin{tikzpicture}
  \path[fill=yellow,draw=yellow!75!red] (0,0) circle (1cm);
  \fill[red] (45:5mm) circle (1mm);
  \fill[red] (135:5mm) circle (1mm);
  \draw[line width=1mm,red] (215:5mm) arc (215:325:5mm);
\end{tikzpicture}
}
```

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
% \tcbuselibrary{skins,xparse}
\tcbsidebyside[sidebyside adapt=both,
  enhanced,center,
  title=Both sides adapted,
  attach boxed title to top center={yshift=-2mm},
  coltitle=black,boxed title style={colback=red!25},
  segmentation style=solid,colback=red!5,colframe=red!50
]{%
  \begin{tabular}{|l|c|r|}\hline
    left & center & right\\\hline
    A & B & C\\\hline
    D & E & F\\\hline
  \end{tabular}
}{%
\begin{tikzpicture}
  \path[fill=yellow,draw=yellow!75!red] (0,0) circle (1cm);
  \fill[red] (45:5mm) circle (1mm);
  \fill[red] (135:5mm) circle (1mm);
  \draw[line width=1mm,red] (215:5mm) arc (215:325:5mm);
\end{tikzpicture}
}
```

Both sides adapted

left	center	right
A	B	C
D	E	F

N 2015-11-20 **/tcb/sidebyside switch=true|false** (default `true`, initially `false`)

If set to `true`, the ⟨*left-handed content*⟩ and ⟨*right-handed content*⟩ of `\tcbsidebyside` → P. 114 are switched. Obviously, this option is only valid inside `\tcbsidebyside` → P. 114.

The side switching can be made even/odd page sensitive, if used inside `/tcb/if odd page` → P. 95.

```
% \tcbuselibrary{skins,xparse}
\tcbsidebyside{Left}{Right}

\tcbsidebyside[sidebyside switch]{Left}{Right}

\tcbsidebyside[title=Very important table,
  if odd page={sidebyside switch,sidebyside adapt=right,flushright title}%
              {sidebyside adapt=left},
  beamer,colframe=blue!50!black,colback=blue!10,
  lower separated=false,sidebyside gap=5mm
]{%
  \begin{tabular}{|l|c|r|}\hline
    left & center & right\\\hline
    A & B & C\\\hline
    D & E & F\\\hline
  \end{tabular}
}{%
  This table contains the most important figures for
  all future actions. You may notice that B follows A,
  C follows B, and so on.
}
```

Left | Right

Right | Left

Very important table

This table contains the most important figures for all future actions. You may notice that B follows A, C follows B, and so on.

left	center	right
A	B	C
D	E	F

7 Saving and Loading of Verbatim Texts

The following macros are slightly modified versions of the original macros from the known packages `moreverb` and `verbatim`. They are used implicitly inside of a `tcolorbox` environment, but they can be used outside also.

`\begin{tcbverbatimwrite}{`⟨*file name*⟩`}`
⟨*environment content*⟩
`\end{tcbverbatimwrite}`

Saves the ⟨*environment content*⟩ to a file named by ⟨*file name*⟩. TEX macros inside the environment are not expanded.

```
\begin{tcbverbatimwrite}{\jobname_verbexp.tex}
  This text is saved \textit{as is}.
\end{tcbverbatimwrite}

Now, we are using the file:\par
\input{\jobname_verbexp.tex}
```

Now, we are using the file:
This text is saved *as is*.

This environment may be used inside an own environment. Note, that inside the environment definition `\tcbverbatimwrite` has to be used instead of `\begin{tcbverbatimwrite}` and `\endtcbverbatimwrite` instead of `\end{tcbverbatimwrite}`.

```
\newenvironment{myverbatim}{%
  \begingroup\tcbverbatimwrite{\jobname_myverb.tex}}%
  {\endtcbverbatimwrite\endgroup}

\begin{myverbatim}
  This is the text which is saved by my own environment.
\end{myverbatim}

Now, we are using the file:\par
\input{\jobname_myverb.tex}
```

Now, we are using the file:
This is the text which is saved by my own environment.

`\begin{tcbwritetemp}`
⟨*environment content*⟩
`\end{tcbwritetemp}`

Has the same function as `tcbverbatimwrite`, but uses the key value of `tempfile` for the file name.

```
\begin{tcbwritetemp}
  This text is saved \textit{as is}.
\end{tcbwritetemp}

Now, we are using the file:\par
\tcbusetemp
```

Now, we are using the file:
This text is saved *as is*.

`\tcbusetemp`

Loads the current temporary file which was saved by `tcbwritetemp`.

8 Recording

The package provides some macros and options to take *records* during compilation. This is done by LaTeX file operations to save some data to a file for later usage. The main application scenario is depicted in Section 8.3 on the next page where information about example solutions is recorded and read again in Section 8.4 on page 123.

8.1 Makros

N 2014-11-28 `\tcbstartrecording[`⟨*file name*⟩`]`

Opens a file denoted by ⟨*file name*⟩ for writing the records. The default file name is `\jobname.records`. See Section 8.3 on the next page for an example application.

N 2014-11-28 `\tcbrecord{`⟨*content*⟩`}`

Records any ⟨*content*⟩ to the record file. `\tcbrecord` is implemented as `\immediate\write`. `\tcbstartrecording` has to be called before; otherwise, `\tcbrecord` is silently ignored.

```
\tcbrecord{\string\solution{\thetcbcounter}{solutions/exercise-\thetcbcounter.tex}}
```

N 2014-11-28 `\tcbstoprecording`

Closes the current record file which was opened by `\tcbstartrecording` before.

N 2014-11-28 `\tcbinputrecords[`⟨*file name*⟩`]`

Opens a file denoted by ⟨*file name*⟩ for reading the records via `\input`. The default file name is the name of the last used record file for saving. `\tcbstoprecording` has to be called before.

8.2 Options

N 2014-11-28 `/tcb/record=`⟨*content*⟩ (style, no default)

Records any ⟨*content*⟩ to the record file, see `\tcbrecord`. This key can be used several times to write several lines.

```
record={\string\solution{\thetcbcounter}{solutions/exercise-\thetcbcounter.tex}}
```

N 2014-11-28 `/tcb/no recording`

Disables `\tcbrecord` and `/tcb/record` inside the current group.

8.3 Example: Exercises

The following application example creates exercises and their corresponding solutions. Each pair is generated inside a single `tcolorbox` where the solution is given below `\tcblower` → P. 12. For every example, the solution part is saved by `/tcb/savelowerto` → P. 24 to a file. The saving is recorded using `/tcb/record` → P. 119. To enlighten the possibilities, the second exercise has no solution. Finally, the solutions are input in Section 8.4 on page 123.

Definition in the preamble:

```
% \tcbuselibrary{skins,xparse}

\NewTColorBox[auto counter,number within=section]{exercise}{+O{}}{%
  enhanced,colframe=green!20!black,colback=yellow!10!white,coltitle=green!40!black,
  fonttitle=\bfseries,
  underlay={\begin{tcbclipinterior}
    \shade[inner color=green!80!yellow,outer color=yellow!10!white]
      (interior.north west) circle (2cm);
    \draw[help lines,step=5mm,yellow!80!black,shift={(interior.north west)}]
      (interior.south west) grid (interior.north east);
    \end{tcbclipinterior}},
  title={Exercise~\thetcbcounter:},
  label={exercise@\thetcbcounter},
  attach title to upper=\quad,
  after upper={\par\hfill\textcolor{green!40!black}%
    {\itshape Solution on page~\pageref{solution@\thetcbcounter}}},
  lowerbox=ignored,
  savelowerto=solutions/exercise-\thetcbcounter.tex,
  record={\string\solution{\thetcbcounter}{solutions/exercise-\thetcbcounter.tex}},
  #1
}

\NewTotalTColorBox{\solution}{mm}{%
  enhanced,colframe=red!20!black,colback=yellow!10!white,coltitle=red!40!black,
  fonttitle=\bfseries,
  underlay={\begin{tcbclipinterior}
    \shade[inner color=red!50!yellow,outer color=yellow!10!white]
      (interior.north west) circle (2cm);
    \draw[help lines,step=5mm,yellow!80!black,shift={(interior.north west)}]
      (interior.south west) grid (interior.north east);
    \end{tcbclipinterior}},
  title={Solution of Exercise~\ref{exercise@#1} on page~\pageref{exercise@#1}:},
  phantomlabel={solution@#1},
  attach title to upper=\par,
}{\input{#2}}

\tcbset{no solution/.style={no recording,after upper=}}
```

```
\tcbstartrecording

\begin{exercise}
  Compute the derivative of the following function:
  \begin{equation*}
    f(x)=\sin((\sin x)^2)
  \end{equation*}
\tcblower
  The derivative is:
  \begin{align*}
    f'(x) &= \left( \sin((\sin x)^2) \right)'
    =\cos((\sin x)^2) 2\sin x \cos x.
  \end{align*}
```

```
\end{exercise}

\begin{exercise}[no solution]
  It holds:
  \begin{equation*}
    \frac{d}{dx}\left(\ln|x|\right) = \frac{1}{x}.
  \end{equation*}
\end{exercise}

\begin{exercise}
  Compute the derivative of the following function:
  \begin{equation*}
    f(x)=(\sin(\sin x))^2
  \end{equation*}
\tcblower
  The derivative is:
  \begin{align*}
    f'(x) &= \left( (\sin(\sin x))^2 \right)'
    =2\sin(\sin x)\cos(\sin x)\cos x.
  \end{align*}
\end{exercise}

\begin{exercise}
  Compute the derivative of the following function:
  \begin{equation*}
    f(x)=\sqrt{x^3-6x^2+2x}
  \end{equation*}
\tcblower
  The derivative is:
  \begin{align*}
    f'(x) &= \left( \sqrt{x^3-6x^2+2x} \right)'
    = \frac{3x^2-12x+2}{2\sqrt{x^3-6x^2+2x}}.
  \end{align*}
\end{exercise}

\begin{exercise}
  Compute the derivative of the following function:
  \begin{equation*}
    f(x)=\left(\frac{2+3x}{1-2x}\right)^3
  \end{equation*}
\tcblower
  The derivative is:
  \begin{align*}
    f'(x) &= \left( \left(\frac{2+3x}{1-2x}\right)^3 \right)'
    = 3 \left(\frac{2+3x}{1-2x}\right)^2 \frac{(1-2x)3-(2+3x)(-2)}{(1-2x)^2}
    = \frac{21(2+3x)^2}{(1-2x)^4}.
  \end{align*}
\end{exercise}

\begin{exercise}
  Compute the derivative of the following function:
  \begin{equation*}
    f(x)=\frac{\cos x}{(\tan 2x)^2}
  \end{equation*}
\tcblower
  The derivative is:
  \begin{align*}
    f'(x) &= \left( \frac{\cos x}{(\tan 2x)^2} \right)'
    = \left( \frac{\cos x (\cos 2x)^2}{(\sin 2x)^2} \right)'\\
    &= \frac{(\sin 2x)^2 [(-\sin x)(\cos 2x)^2+(\cos x)4\cos 2x (-\sin 2x)]
       - \cos x (\cos 2x)^2 4\sin 2x \cos 2x}{(\sin 2x)^4}\\
    &= -\frac{\cos(2x) [\sin x \sin 2x \cos 2x+ 4\cos x(\sin 2x)^2
       + 4 \cos x (\cos 2x)^2]}{(\sin 2x)^3}\\
```

```
    &= -\frac{\cos(2x) [\sin x \sin 2x \cos 2x+ 4\cos x]}{(\sin 2x)^3}.
  \end{align*}
\end{exercise}

\begin{exercise}
  Compute the derivative of the following function:
  \begin{equation*}
    f(x)=\cos((2x^2+3)^3)
  \end{equation*}
\tcblower
  The derivative is:
  \begin{align*}
    f'(x) &= \left( \cos((2x^2+3)^3) \right)'
    =-\sin((2x^2+3)^3) 3(2x^2+3)^2 2\cdot 2x\\
    &=-12x(2x^2+3)^2\sin((2x^2+3)^3).
  \end{align*}
\end{exercise}

\begin{exercise}
  Compute the derivative of the following function:
  \begin{equation*}
    f(x)=(x^2+1)\sqrt{x^4+1}
  \end{equation*}
\tcblower
  The derivative is:
  \begin{align*}
    f'(x) &= \left( (x^2+1)\sqrt{x^4+1} \right)'
    = 2x\sqrt{x^4+1} + \frac{2x^3(x^2+1)}{\sqrt{x^4+1}}.
  \end{align*}
\end{exercise}

\tcbstoprecording
```

Exercise 8.1: Compute the derivative of the following function:

$$f(x) = \sin((\sin x)^2)$$

Solution on page 123

Exercise 8.2: It holds:

$$\frac{d}{dx}(\ln \mathtt{x}) = \frac{1}{x}.$$

Exercise 8.3: Compute the derivative of the following function:

$$f(x) = (\sin(\sin x))^2$$

Solution on page 123

Exercise 8.4: Compute the derivative of the following function:

$$f(x) = \sqrt{x^3 - 6x^2 + 2x}$$

Solution on page 123

Exercise 8.5: Compute the derivative of the following function:

$$f(x) = \left(\frac{2+3x}{1-2x}\right)^3$$

Solution on page 124

Exercise 8.6: Compute the derivative of the following function:

$$f(x) = \frac{\cos x}{(\tan 2x)^2}$$

Solution on page 124

Exercise 8.7: Compute the derivative of the following function:

$$f(x) = \cos((2x^2+3)^3)$$

Solution on page 124

Exercise 8.8: Compute the derivative of the following function:

$$f(x) = (x^2+1)\sqrt{x^4+1}$$

Solution on page 124

8.4 Example: Solutions

This concludes the example given in Section 8.3 on page 120. Now, the saved and recorded solutions are included.

```
\tcbinputrecords
```

Solution of Exercise 8.1 on page 122:
The derivative is:

$$f'(x) = \Big(\sin((\sin x)^2)\Big)' = \cos((\sin x)^2)2\sin x\cos x.$$

Solution of Exercise 8.3 on page 122:
The derivative is:

$$f'(x) = \Big((\sin(\sin x))^2\Big)' = 2\sin(\sin x)\cos(\sin x)\cos x.$$

Solution of Exercise 8.4 on page 122:
The derivative is:

$$f'(x) = \Big(\sqrt{x^3-6x^2+2x}\Big)' = \frac{3x^2-12x+2}{2\sqrt{x^3-6x^2+2x}}.$$

Solution of Exercise 8.5 on page 123:
The derivative is:

$$f'(x) = \left(\left(\frac{2+3x}{1-2x}\right)^3\right)' = 3\left(\frac{2+3x}{1-2x}\right)^2 \frac{(1-2x)3-(2+3x)(-2)}{(1-2x)^2} = \frac{21(2+3x)^2}{(1-2x)^4}.$$

Solution of Exercise 8.6 on page 123:
The derivative is:

$$\begin{aligned} f'(x) &= \left(\frac{\cos x}{(\tan 2x)^2}\right)' = \left(\frac{\cos x(\cos 2x)^2}{(\sin 2x)^2}\right)' \\ &= \frac{(\sin 2x)^2[(-\sin x)(\cos 2x)^2 + (\cos x)4\cos 2x(-\sin 2x)] - \cos x(\cos 2x)^2 4\sin 2x\cos 2x}{(\sin 2x)^4} \\ &= -\frac{\cos(2x)[\sin x \sin 2x \cos 2x + 4\cos x(\sin 2x)^2 + 4\cos x(\cos 2x)^2]}{(\sin 2x)^3} \\ &= -\frac{\cos(2x)[\sin x \sin 2x \cos 2x + 4\cos x]}{(\sin 2x)^3}. \end{aligned}$$

Solution of Exercise 8.7 on page 123:
The derivative is:

$$\begin{aligned} f'(x) &= \left(\cos((2x^2+3)^3)\right)' = -\sin((2x^2+3)^3)3(2x^2+3)^2 2\cdot 2x \\ &= -12x(2x^2+3)^2\sin((2x^2+3)^3). \end{aligned}$$

Solution of Exercise 8.8 on page 123:
The derivative is:

$$f'(x) = \left((x^2+1)\sqrt{x^4+1}\right)' = 2x\sqrt{x^4+1} + \frac{2x^3(x^2+1)}{\sqrt{x^4+1}}.$$

9 Technical Overview and Customization

This section provides a technical overview of the skin concept of `tcolorbox`. For most applications of `tcolorbox`, one will not need to know the bells and whistles described herein. You may proceed to Section 10 on page 139 where the customization options for most users are documented.

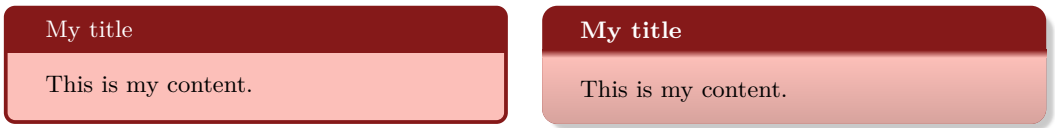
The following explanations also cover options and settings from the LIB skins library, see Section 10 on page 139.

9.1 Skins and Drawing Engines

From a technical point of view, a *skin* is a style definition for the appearance of a `tcolorbox`. The core package provides some additional option keys for skins but only two skins called `standard`→P. 193 and `standard jigsaw`→P. 194. The LIB skins library adds several more skins. To change to a skin, only one option from the core package has to be set.

`/tcb/skin`=⟨*name*⟩ (style, no default, initially `standard`)
Sets the current skin to ⟨*name*⟩. This is a style definition which sets all the following keys, i. e. for many use cases there is nothing more to do.

```
\tcbset{colback=Salmon!50!white,colframe=FireBrick!75!black,
  width=(\linewidth-8mm)/2,before=,after=\hfill,equal height group=ske}

\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}
\begin{tcolorbox}[skin=beamer,beamer,adjusted title=My title]
  This is my content.
\end{tcolorbox}
```



`/tcb/skin first`=⟨*name*⟩ (style, no default, initially `standard`)
If the box is set to be `/tcb/breakable`→P. 351 and *is* broken actually, then the skin for the *first* part of the break sequence is set to ⟨*name*⟩, see Subsection 17.7 on page 362. Typically, this key is set by a `/tcb/skin`.

`/tcb/skin middle`=⟨*name*⟩ (style, no default, initially `standard`)
If the box is set to be `/tcb/breakable`→P. 351 and *is* broken actually, then the skin for the *middle* parts (if any) of the break sequence is set to ⟨*name*⟩, see Subsection 17.7 on page 362. Typically, this key is set by a `/tcb/skin`.

`/tcb/skin last`=⟨*name*⟩ (style, no default, initially `standard`)
If the box is set to be `/tcb/breakable`→P. 351 and *is* broken actually, then the skin for the *last* part of the break sequence is set to ⟨*name*⟩, see Subsection 17.7 on page 362. Typically, this key is set by a `/tcb/skin`.

`/tcb/graphical environment`=⟨*name*⟩ (no default, initially `pgfpicture`)
Sets the graphical environment for the `tcolorbox` to ⟨*name*⟩. Feasible values are `pgfpicture` and `tikzpicture` or environments which inherit from one of these two. This key is set by a `/tcb/skin` and may seldom be used directly.

The skin of a `tcolorbox` is drawn by up to four *engines*. Afterwards, the text content is drawn which is not part of a skin. The four steps are:

1. The *frame* of the box, drawn by `/tcb/frame engine`.
2. The *interior* of the box. The interior of a box with title is drawn differently from a box without title. `/tcb/interior titled engine` or `/tcb/interior engine` → P. 127 is used to draw the interior.
3. The *segmentation* (line) of the box, if there is a lower part; drawn by `/tcb/segmentation engine` → P. 127.
4. The *title area* of the box, if there is a title and `/tcb/title filled` → P. 27 is set to `true`; drawn by `/tcb/title engine` → P. 127.

`/tcb/frame engine`=⟨*name*⟩ (no default, initially `standard`)

Sets the *frame* drawing engine for a box to ⟨*name*⟩. Typically, this key is set by a `/tcb/skin` → P. 125. Feasible values for ⟨*name*⟩ are:

- `standard`: the original code from the core package,
- `path`: a `tikz` path which is controlled by `/tcb/frame style` → P. 139,
- `pathjigsaw`: a `tikz` path which is controlled by `/tcb/frame style` → P. 139,
- `pathfirst`: a `tikz` path which is controlled by `/tcb/frame style` → P. 139,
- `pathfirstjigsaw`: a `tikz` path which is controlled by `/tcb/frame style` → P. 139,
- `pathmiddle`: a `tikz` path which is controlled by `/tcb/frame style` → P. 139,
- `pathmiddlejigsaw`: a `tikz` path which is controlled by `/tcb/frame style` → P. 139,
- `pathlast`: a `tikz` path which is controlled by `/tcb/frame style` → P. 139,
- `pathlastjigsaw`: a `tikz` path which is controlled by `/tcb/frame style` → P. 139,
- `freelance`: deprecated.
- `spartan`: a quite spartan code.
- `empty`: draw nothing.

`/tcb/interior titled engine`=⟨*name*⟩ (no default, initially `standard`)

Sets the *interior* drawing engine for a titled box to ⟨*name*⟩. Typically, this key is set by a `/tcb/skin` → P. 125. Feasible values for ⟨*name*⟩ are:

- `standard`: the original code from the core package,
- `path`: a `tikz` path which is controlled by `/tcb/interior style` → P. 140,
- `pathfirst`: a `tikz` path which is controlled by `/tcb/interior style` → P. 140,
- `pathmiddle`: a `tikz` path which is controlled by `/tcb/interior style` → P. 140,
- `pathlast`: a `tikz` path which is controlled by `/tcb/interior style` → P. 140,
- `freelance`: deprecated.
- `spartan`: a quite spartan code.
- `empty`: draw nothing.

`/tcb/interior engine=`⟨*name*⟩ (no default, initially `standard`)

Sets the *interior* drawing engine for an untitled box to ⟨*name*⟩. Typically, this key is set by a `/tcb/skin` → P. 125. Feasible values for ⟨*name*⟩ are:

- `standard`: the original code from the core package,
- `path`: a `tikz` path which is controlled by `/tcb/interior style` → P. 140,
- `pathfirst`: a `tikz` path which is controlled by `/tcb/interior style` → P. 140,
- `pathmiddle`: a `tikz` path which is controlled by `/tcb/interior style` → P. 140,
- `pathlast`: a `tikz` path which is controlled by `/tcb/interior style` → P. 140,
- `freelance`: deprecated.
- `spartan`: a quite spartan code.
- `empty`: draw nothing.

`/tcb/segmentation engine=`⟨*name*⟩ (no default, initially `standard`)

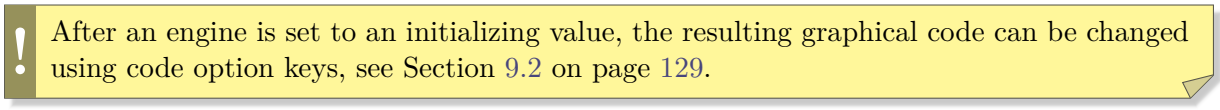
Sets the *segmentation* (line) drawing engine for a box to ⟨*name*⟩. Typically, this key is set by a `/tcb/skin` → P. 125. Feasible values for ⟨*name*⟩ are:

- `standard`: the original code from the core package,
- `path`: a `tikz` path which is controlled by `/tcb/segmentation style` → P. 142,
- `freelance`: deprecated.
- `spartan`: a quite spartan code.
- `empty`: draw nothing.

`/tcb/title engine=`⟨*name*⟩ (no default, initially `standard`)

Sets the *title area* drawing engine for a titled box to ⟨*name*⟩. Typically, this key is set by a `/tcb/skin` → P. 125. Feasible values for ⟨*name*⟩ are:

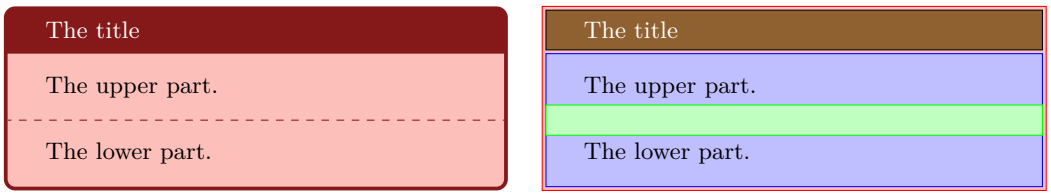
- `standard`: the original code from the core package,
- `path`: a `tikz` path which is controlled by `/tcb/title style` → P. 142,
- `pathfirst`: a `tikz` path which is controlled by `/tcb/title style` → P. 142,
- `pathmiddle`: a `tikz` path which is controlled by `/tcb/title style` → P. 142,
- `pathlast`: a `tikz` path which is controlled by `/tcb/title style` → P. 142,
- `freelance`: deprecated.
- `spartan`: a quite spartan code.
- `empty`: draw nothing.



`/tcb/geometry nodes=true|false` (default `true`, initially `false`)

If set to `true`, up to four `tikz` nodes are defined for a `tcolorbox` which are named `frame`, `interior`, `segmentation`, and `title`. These nodes describe the boundaries of the equally named parts of a `tcolorbox`. They are used by most engines based on Ti*k*Z. Typically, this key is set automatically by a `/tcb/skin` → P. 125.

```
\tcbset{colback=Salmon!50!white,colframe=FireBrick!75!black,
  width=(\linewidth-8mm)/2,before=,after=\hfill,equal height group=geon}

\begin{tcolorbox}[adjusted title=The title]
  The upper part.\tcblower The lower part.
\end{tcolorbox}
\begin{tcolorbox}[enhanced,adjusted title=The title,
  frame code={\path[draw=red,fill=red!25]
      (frame.south west) rectangle (frame.north east);},
  interior titled code={\path[draw=blue,fill=blue!25]
      (interior.south west) rectangle (interior.north east);},
  segmentation code={\path[draw=green,fill=green!25]
      (segmentation.south west) rectangle (segmentation.north east);},
  title code={\path[draw=black,fill=brown!75!black]
      (title.south west) rectangle (title.north east);}]
  The upper part.\tcblower The lower part.
\end{tcolorbox}
```



9.2 Code Option Keys

The following code options are applicable for all skins. The used ⟨*graphical code*⟩ can be any `pgf` code. For all skins with exception of `standard` → P. 193 and `standard jigsaw` → P. 194, the ⟨*graphical code*⟩ can also be any Ti*k*Z code.

`/tcb/frame code=`⟨*graphical code*⟩ (code, default from `standard`)

The given ⟨*graphical code*⟩ is used for drawing the *frame* of the box.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[enhanced,frame code={
  \foreach \n in {north east,north west,south east,south west}
  {\path [fill=red!75!black] (interior.\n) circle (3mm); }; }]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

`/tcb/frame empty` (style, no value)

This is a shortcut for setting `/tcb/frame code` to empty. This option removes the drawing of the frame. Alternatively, use `/tcb/frame hidden` → P. 140.

`/tcb/interior titled code=`⟨*graphical code*⟩ (code, default from `standard`)

The given ⟨*graphical code*⟩ is used for drawing the *interior* of the box, if the box comes with a title.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,interior titled code={
  \path[draw=red!5!white,line width=5mm,line cap=round]
    ([xshift=3mm,yshift=-3mm]interior.north west)
    --([xshift=-3mm,yshift=3mm]interior.south east)
    ([xshift=3mm,yshift=3mm]interior.south west)
    --([xshift=-3mm,yshift=-3mm]interior.north east);}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

This is the lower part.

`/tcb/interior titled empty` (style, no value)

This is a shortcut for setting `/tcb/interior titled code` to empty. This option removes the drawing of the untitled interior. Alternatively, use `/tcb/interior hidden` → P. 141.

`/tcb/interior code=`⟨*graphical code*⟩ (code, default from `standard`)

The given ⟨*graphical code*⟩ is used for drawing the *interior* of the box, if the box is without a title.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[enhanced,interior code={
  \path[draw=red!5!white,line width=5mm,line cap=round]
    ([xshift=3mm,yshift=-3mm]interior.north west)
    --([xshift=-3mm,yshift=3mm]interior.south east)
    ([xshift=3mm,yshift=3mm]interior.south west)
    --([xshift=-3mm,yshift=-3mm]interior.north east);}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

`/tcb/interior empty` (style, no value)

This is a shortcut for setting `/tcb/interior code` to empty. This option removes the drawing of the interior. Alternatively, use `/tcb/interior hidden` → P. 141.

`/tcb/segmentation code=`⟨*graphical code*⟩ (code, default from `standard`)

The given ⟨*graphical code*⟩ is used for drawing the *segmentation* area of the box.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,segmentation code={
  \path[top color=red!5!white,bottom color=red!5!white,middle color=blue]
  (segmentation.south west) rectangle (segmentation.north east);}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

This is the lower part.

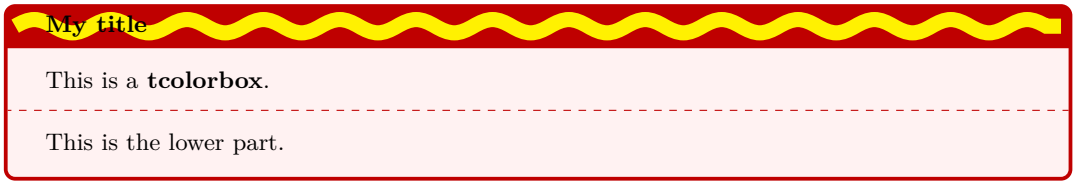
`/tcb/segmentation empty` (style, no value)

This is a shortcut for setting `/tcb/segmentation code` to empty. This option removes the drawing of the segmentation line. Alternatively, use `/tcb/segmentation hidden` → P. 142.

`/tcb/title code=`⟨*graphical code*⟩ (code, default from `standard`)

The given ⟨*graphical code*⟩ is used for drawing the *title* area of the box.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  coltitle=black}

\begin{tcolorbox}[enhanced,title=My title,title code={
  \path[draw=yellow,solid,decorate,line width=2mm,
    decoration={coil,aspect=0,segment length=10.1mm}]
    ([xshift=1mm]title.west) -- ([xshift=-1mm]title.east);}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/title empty` (style, no value)

This is a shortcut for setting `/tcb/title code` to empty. This option removes the drawing of the title area. Alternatively, use `/tcb/title hidden` → P. 143.

9.3 Subskins

A subskin is a new /tcb/skin → P. 125 based on an existing skin which is extended or changed.

> **!** Never use geometry settings or bounding box options inside a subskin definition! If one skin is replaced by another skin, the overall bounding box should stay constant. Especially, if a skin is used for a breakable box, unpredictable and unpleasant results could arise otherwise. If you want to change the geometry also, use an additional style. See the skin beamer → P. 217 and the style /tcb/beamer → P. 217 as pattern.

\tcbsubskin{⟨*name*⟩}{⟨*base skin*⟩}{⟨*options*⟩}

Creates a new skin ⟨*name*⟩ which inherits all properties of an existing ⟨*base skin*⟩ plus the given ⟨*options*⟩. The new skin ⟨*name*⟩ can be used as value for the keys /tcb/skin → P. 125, /tcb/skin first → P. 125, /tcb/skin middle → P. 125, and /tcb/skin last → P. 125. As ⟨*base skin*⟩, one can take standard → P. 193, empty → P. 226, enhanced → P. 195, or any skin from the LIB skins library, see Section 10 on page 139.

```
% \tcbuselibrary{skins}
\tcbsubskin{mycross}{empty}{frame code={%
  \draw[red,line width=5pt] (frame.south west)--(frame.north east);
  \draw[red,line width=5pt] (frame.north west)--(frame.south east);},
  skin first=mycross,skin middle=mycross,skin last=mycross }

\begin{tcolorbox}[skin=mycross]
  This is my content.
\end{tcolorbox}
```

/tcb/skin first is subskin of={⟨*base skin*⟩}{⟨*options*⟩} (no default, initially unset)

Creates a new unnamed skin which inherits all properties of an existing ⟨*base skin*⟩ plus the given ⟨*options*⟩. This skin is set as /tcb/skin first → P. 125.
See a detailed example on page 232.

/tcb/skin middle is subskin of={⟨*base skin*⟩}{⟨*options*⟩} (no default, initially unset)

Creates a new unnamed skin which inherits all properties of an existing ⟨*base skin*⟩ plus the given ⟨*options*⟩. This skin is set as /tcb/skin middle → P. 125.
See a detailed example on page 232.

/tcb/skin last is subskin of={⟨*base skin*⟩}{⟨*options*⟩} (no default, initially unset)

Creates a new unnamed skin which inherits all properties of an existing ⟨*base skin*⟩ plus the given ⟨*options*⟩. This skin is set as /tcb/skin last → P. 125.
See a detailed example on page 232.

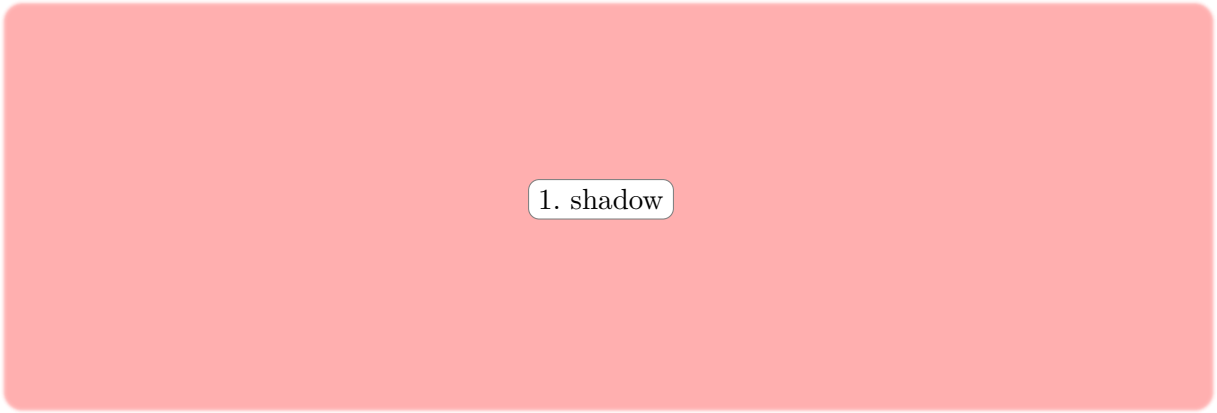
9.4 Drawing Scheme

Depending on the complexity of a `tcolorbox` definition, the resulting box is drawn in a more or less complex series of steps.

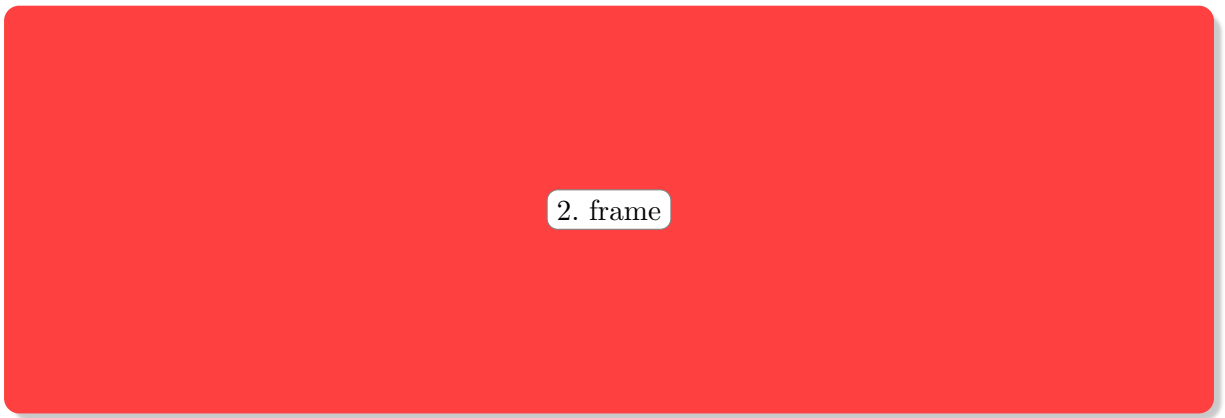
To document and demonstrate these drawing steps, we consider the following box definition:

```
\newtcolorbox{testbox}[1][]{enhanced,title=Test Box,
  boxrule=1mm,titlerule=0.5mm,colframe=blue!50!black,
  interior style={top color=blue!20!green!50!white,bottom color=blue!20!yellow!50!white},
  colbacktitle=blue!50!green!90!white,segmentation style={solid},
  fonttitle=\bfseries,drop fuzzy shadow,borderline={0.3mm}{0.35mm}{yellow!50!white},
  underlay={\path[fill image opacity=0.15,fill image scale=0.9,
    fill stretch picture={\draw[blue,line width=2mm] circle (1);}]
    (interior.south west) rectangle (interior.north east);},
  watermark text={Watermark},watermark color={green!20!white},
  finish={\begin{tcbclipframe}
    \path[bottom color=black,top color=black!50!white,opacity=0.1]
      (frame.south west) -- (frame.south east) -- (frame.north east) -- cycle;
    \path[top color=white,bottom color=black!50!white,opacity=0.1]
      (frame.south west) -- (frame.north east) -- (frame.north west) -- cycle;
    \end{tcbclipframe}},#1}
```

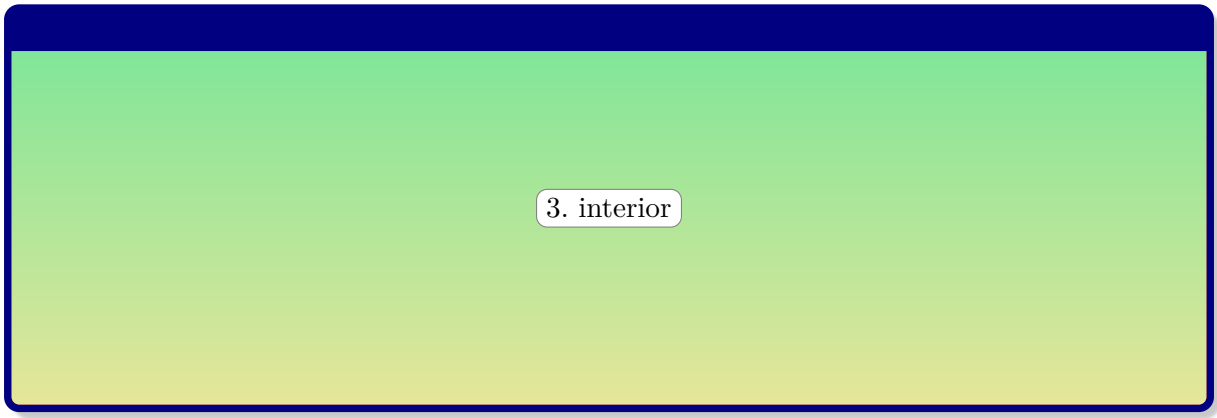
For this definition, we get the maximal number of drawing steps:



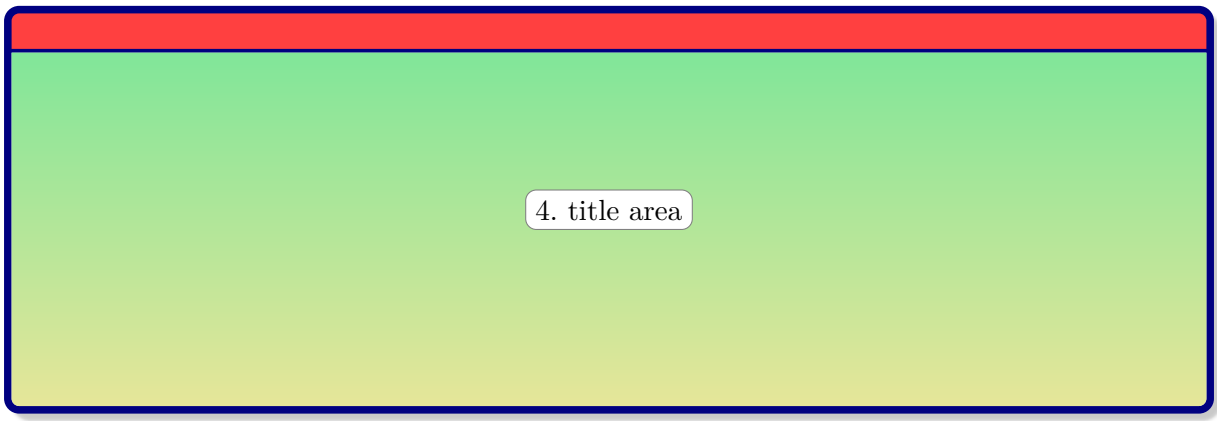
- Section 10.6 on page 173.



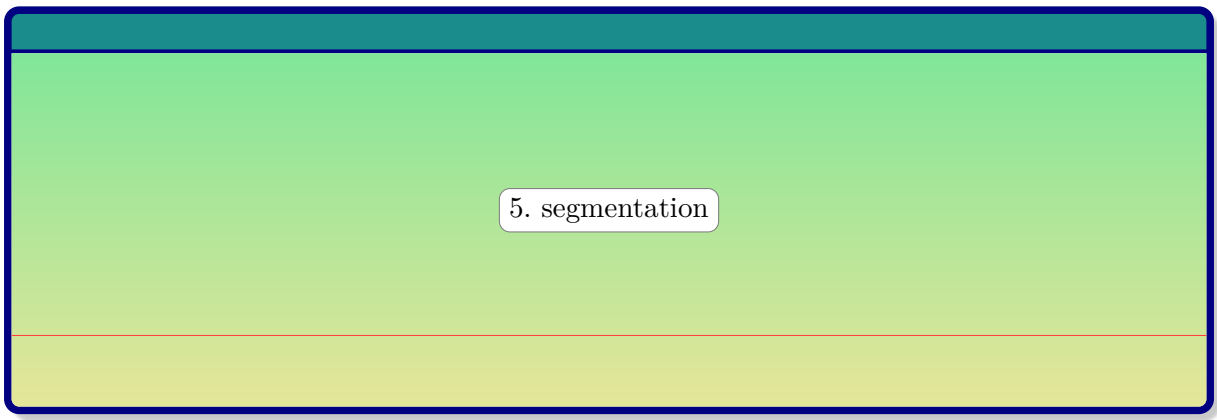
- `/tcb/colframe` → P. 27, `/tcb/opacityframe` → P. 50
- `/tcb/frame code` → P. 129
- `/tcb/frame style` → P. 139



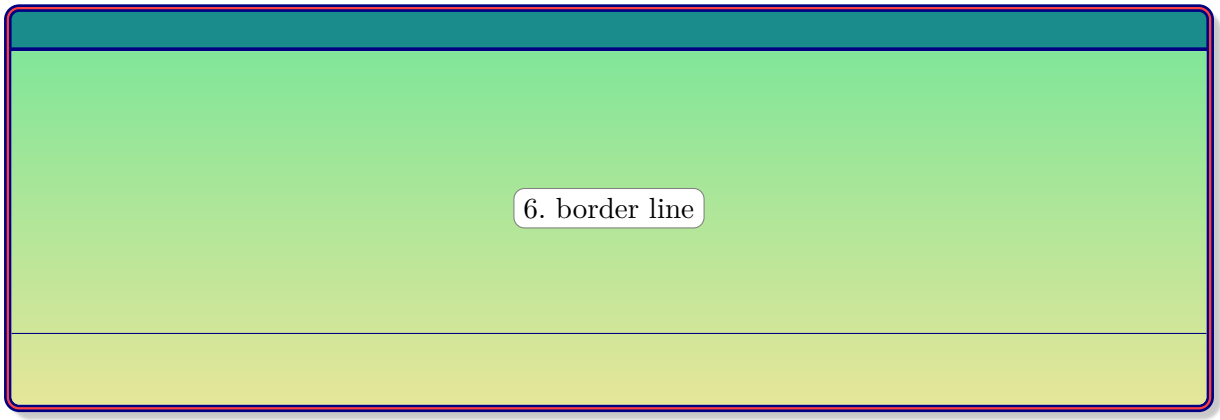
- /tcb/colback → P. 27, /tcb/opacityback → P. 50
- /tcb/interior code → P. 130, /tcb/interior titled code → P. 129
- /tcb/interior style → P. 140



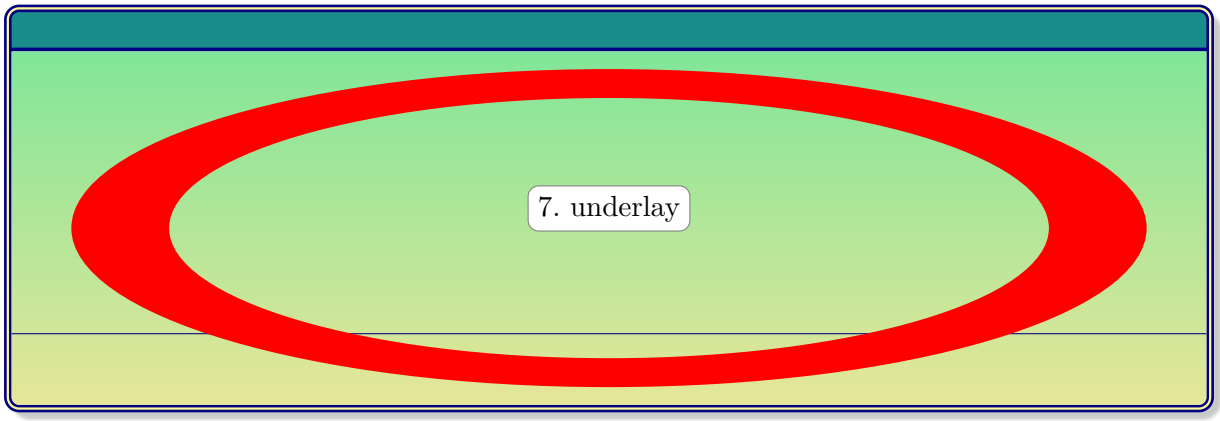
- /tcb/colbacktitle → P. 27, /tcb/opacitybacktitle → P. 50
- /tcb/title code → P. 131
- /tcb/title style → P. 142



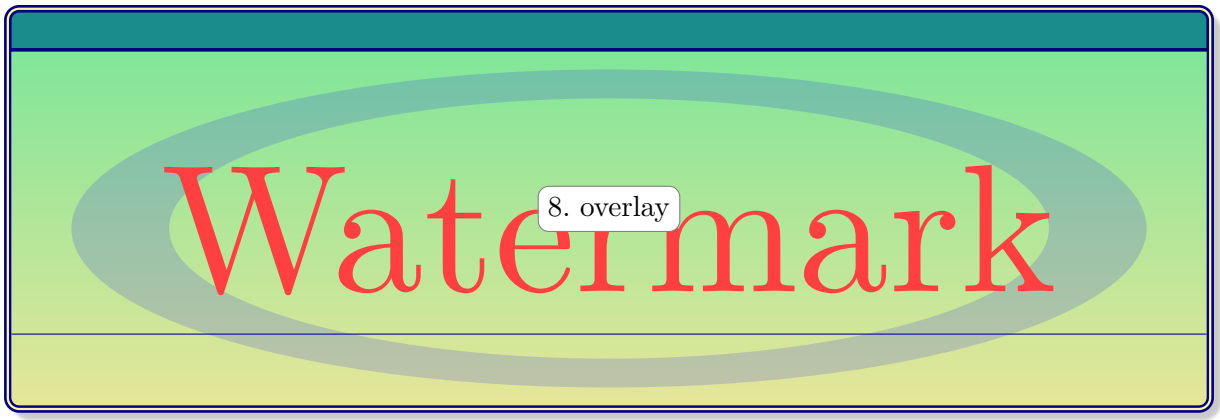
- /tcb/lower separated → P. 25
- /tcb/segmentation code → P. 130
- /tcb/segmentation style → P. 142



- Section 10.5 on page 168



- Section 10.2 on page 146
- Section 10.8 on page 186



- Section 4.12 on page 69
- Section 10.3 on page 156

Test Box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

9. text content

Lower part

- `/tcb/colupper` → P. 28, `/tcb/collower` → P. 28, `/tcb/coltitle` → P. 28
- `/tcb/fontupper` → P. 29, `/tcb/fontlower` → P. 29, `/tcb/fonttitle` → P. 29
- `/tcb/opacityupper` → P. 51, `/tcb/opacitylower` → P. 51, `/tcb/opacitytitle` → P. 51

Test Box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

10. finish

Lower part

- Section 10.9 on page 188

All together, the box is drawn:

```
% \usepackage{lipsum}
\begin{testbox}
\lipsum[2]
\tcblower
Lower part
\end{testbox}
```

Test Box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Lower part

9.5 Useful Properties

The following macros describe certain *properties* which may be used for the drawing scheme, see Section 9.4 on page 133. Sometimes, they are even available inside the box content. All of them are considered to be *read-only* and should never be redefined by the user.

N 2016-02-16 **\tcbheightspace**

If the height of a `tcolorbox` is not the natural height, the space difference between the forced and the natural size is hold by `\tcbheightspace`. This macro is not usable inside the box content, but for skins or inside `/tcb/underlay` → P. 186, `/tcb/overlay` → P. 69, etc. If such a space information is needed inside the box content, see `/tcb/space to` → P. 57 instead.

```
% \tcbuselibrary{skins}
\newtcolorbox{testbox}[2][]{enhanced,size=fbox,
  colframe=blue!75!black,colback=white,height=#2,
  underlay={\node[above,inner sep=3pt] at (interior.south){%
    \includegraphics[width=\tcbtextwidth,height=\tcbheightspace-3pt]{goldshade.png}};
    },
  #1}
\begin{testbox}{3cm}
  This is my box. The space is filled with a picture.
\end{testbox}
\begin{testbox}{2cm}
  This is my box. The space is filled with a picture.
\end{testbox}
```

This is my box. The space is filled with a picture.

This is my box. The space is filled with a picture.

N 2016-02-16 **\tcbtextwidth**

This property describes the box content width.

- If there also is a lower part, it describes the width of the upper part.
- For `/tcb/sidebyside` → P. 108 boxes, it describes the combined text width plus segmentation.
- This property can be used inside the box content text with exception of `/tcb/fit` → P. 382 boxes.
- `\tcbtextwidth` can be used for all box types for skins or inside `/tcb/underlay` → P. 186, `/tcb/overlay` → P. 69, etc.

```
\begin{tcolorbox}[colframe=blue!75!black]
  Inside a box: \tcbtextwidth\ (=\the\linewidth).
\end{tcolorbox}
```

Inside a box: 370.74823pt (=370.74823pt).

N 2016-02-16 **\tcbtextheight**

This property describes the designated box content height. If the box is larger than the natural height, the actual content will be smaller than \tcbtextheight.

- For boxes with a fixed /tcb/height → P. 52, this property can be used inside the box content text. For other boxes, it denotes 0pt inside the box content.
- \tcbtextheight can be used for all box types for skins or inside /tcb/underlay → P. 186, /tcb/overlay → P. 69, etc.

```
% \tcbuselibrary{skins}
\begin{tcolorbox}[enhanced,colframe=blue!75!black,
    underlay={\node[left,red] at (frame.east) {Here: \tcbtextheight};}]
  Inside a box with natural height: \tcbtextheight.
\end{tcolorbox}
\begin{tcolorbox}[enhanced,colframe=blue!75!black,height=1cm,
    underlay={\node[left,red] at (frame.east) {Here: \tcbtextheight};}]
  Inside a box with fixed height: \tcbtextheight.
\end{tcolorbox}
```

Inside a box with natural height: 0pt. Here: 7.95pt

Inside a box with fixed height: 8.5359pt. Here: 8.5359pt

10 Library LIB skins

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{skins}
```

This also loads the package `tikz` [20]. Typically but not necessarily, the following skins use `tikz` instead of `pgf`.

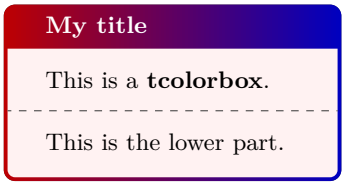
10.1 Style Option Keys

The following style options are applicable for all skins which use engines of type `path`, `pathfirst`, `pathmiddle`, or `pathlast`. Especially, the skin `enhanced` → P. 195 supports *all* of them and `standard` → P. 193 *none*.

`/tcb/frame style=`⟨`tikz` *keys*⟩ (style, no default)

The ⟨`tikz` *keys*⟩ are used inside the `tikz` path command for drawing the *frame* of the box. This option is available if the `/tcb/frame engine` → P. 126 is set to `path`, `pathfirst`, `pathmiddle`, or `pathlast`. It is *not* available for `standard`.

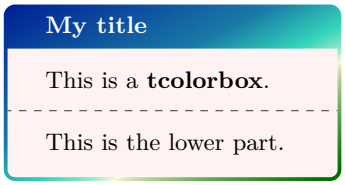
```
\tcbset{colback=red!5!white,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  frame style={left color=red!75!black,
               right color=blue!75!black}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/frame style image=`⟨*file name*⟩ (no default, initially unset)

Fills the frame with an external image referenced by ⟨*file name*⟩. For advanced features like blending of a picture with the background, use `/tcb/frame style` together with `/tikz/fill stretch image` → P. 245.

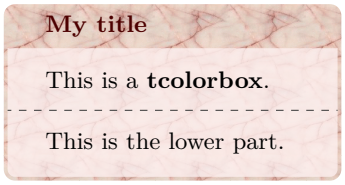
```
\tcbset{colback=red!5!white,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  frame style image=blueshade.png]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/frame style tile={⟨graphics options⟩}{⟨file name⟩}` (no default, initially unset)

Fills the frame with a tile pattern based on an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. For advanced features like blending of a picture with the background, use `/tcb/frame style` → P. 139 together with `/tikz/fill tile image` → P. 249.

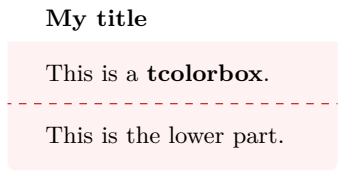
```
\tcbset{colback=red!5!white,coltitle=red!30!black,
  opacityback=0.75,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  frame style tile={width=1cm}{pink_marble.png}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/frame hidden` (style, no value)

This is a shortcut for `frame style={draw=none,fill=none}`. Depending on the skin, this option switches off the drawing of the frame. Alternatively, use `/tcb/frame empty` → P. 129.

```
\tcbset{colback=red!5!white,colframe=red!75!black,
  fonttitle=\bfseries,coltitle=black}

\begin{tcolorbox}[enhanced,title=My title,
  frame hidden]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

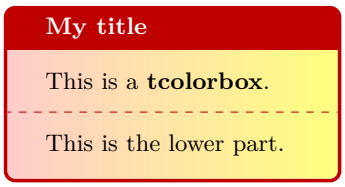


`/tcb/interior style=`⟨`tikz` *keys*⟩ (style, no default)

The ⟨`tikz` *keys*⟩ are used inside the `tikz` path command for drawing the *interior* of the box. They are used for the titled and for the untitled version as well.

This option is available if the `/tcb/interior titled engine` → P. 126 or `/tcb/interior engine` → P. 127 is set to `path`, `pathfirst`, `pathmiddle`, or `pathlast`. It is *not* available for `standard`.

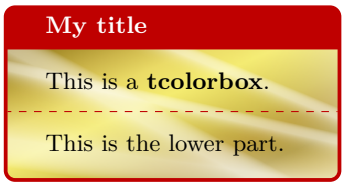
```
\tcbset{colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  interior style={left color=red!20!white,
                  right color=yellow!50!white}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/interior style image=`⟨*file name*⟩ (no default, initially unset)

Fills the interior with an external image referenced by ⟨*file name*⟩. For advanced features like blending of a picture with the background, use `/tcb/interior style` → P. 140 together with `/tikz/fill stretch image` → P. 245.

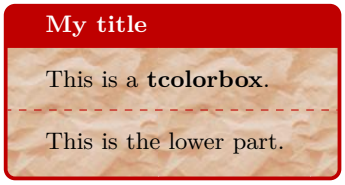
```
\tcbset{colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  interior style image=goldshade.png]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/interior style tile={`⟨*graphics options*⟩`}{`⟨*file name*⟩`}` (no default, initially unset)

Fills the interior with a tile pattern based on an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. For advanced features like blending of a picture with the background, use `/tcb/interior style` → P. 140 together with `/tikz/fill tile image` → P. 249.

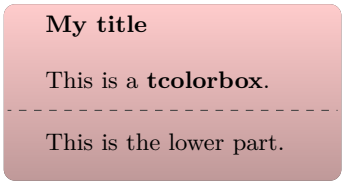
```
\tcbset{colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  interior style tile={width=2cm}{crinklepaper.png}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/interior hidden` (style, no value)

This is a shortcut for `interior style={draw=none,fill=none}`. Depending on the skin, this option switches off the drawing of the interior. Alternatively, use `/tcb/interior empty` → P. 130 and/or `/tcb/interior titled empty` → P. 129.

```
\tcbset{frame style={top color=red!20!white,
  bottom color=red!20!white!75!black},
  fonttitle=\bfseries,coltitle=black}

\begin{tcolorbox}[enhanced,title=My title,
  interior hidden]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

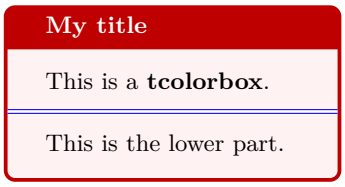


`/tcb/segmentation style=`⟨`tikz` *keys*⟩ (style, no default)

The ⟨`tikz` *keys*⟩ are used inside the `tikz` path command for drawing the *segmentation* line of the box.

This option is available if the `/tcb/segmentation engine` → P. 127 is set to `path`. It is *not* available for `standard`.

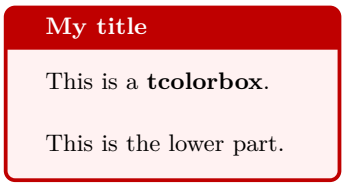
```
\tcbset{colback=red!5!white,colframe=red!75!black,
  fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  segmentation style={double=white,draw=blue,
                  double distance=1pt,solid}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/segmentation hidden` (style, no value)

This is a shortcut for `segmentation style={draw=none,fill=none}`. Depending on the skin, this option switches off the drawing of the segmentation line. See also `/tcb/lower separated` → P. 25 which has the same effect for most skins. Alternatively, use `/tcb/segmentation empty` → P. 130.

```
\tcbset{colback=red!5!white,colframe=red!75!black,
  fonttitle=\bfseries}

\begin{tcolorbox}[title=My title,
  enhanced,segmentation hidden]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

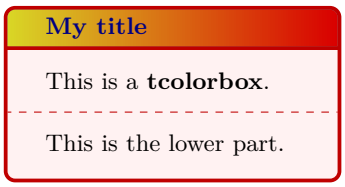


`/tcb/title style=`⟨`tikz` *keys*⟩ (style, no default)

The ⟨`tikz` *keys*⟩ are used inside the `tikz` path command for drawing the *title area* of the box.

This option is available if the `/tcb/title engine` → P. 127 is set to `path`, `pathfirst`, `pathmiddle`, or `pathlast`. It is *not* available for `standard`.

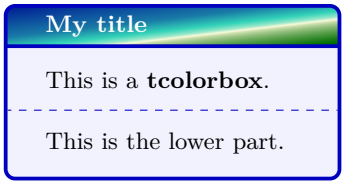
```
\tcbset{colback=red!5!white,colframe=red!75!black,
  coltitle=blue!50!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  title style={left color=blue!15!yellow,
               right color=red!85!black}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/title style image=`⟨*file name*⟩ (no default, initially unset)

Fills the title area with an external image referenced by ⟨*file name*⟩. For advanced features like blending of a picture with the background, use `/tcb/title style` → P. 142 together with `/tikz/fill stretch image` → P. 245.

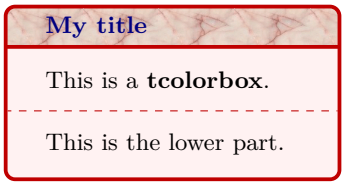
```
\tcbset{colback=blue!5!white,colframe=blue!75!black,
  fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  title style image=blueshade.png]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/title style tile={`⟨*graphics options*⟩`}{`⟨*file name*⟩`}` (no default, initially unset)

Fills the title area with a tile pattern based on an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. For advanced features like blending of a picture with the background, use `/tcb/title style` → P. 142 together with `/tikz/fill tile image` → P. 249.

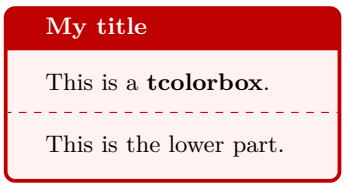
```
\tcbset{colback=red!5!white,colframe=red!75!black,
  coltitle=blue!50!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  title style tile={width=1cm}{pink_marble.png}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/title hidden` (style, no value)

This is a shortcut for `title style={draw=none,fill=none}`. Depending on the skin, this option switches off the drawing of the title background. See also `/tcb/title filled` → P. 27 for a similar effect. Alternatively, use `/tcb/title empty` → P. 131.

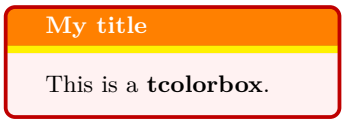
```
\tcbset{colback=red!5!white,colframe=red!75!black,
  fonttitle=\bfseries}

\begin{tcolorbox}[title=My title,
  enhanced,title hidden]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



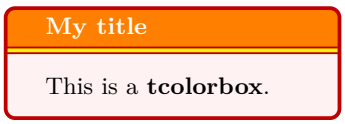
N 2015-01-14 **/tcb/titlerule style=⟨tikz *keys*⟩** (style, no default)

The ⟨tikz *keys*⟩ are used to draw a title rule, i.e. a rule below the optional title. The width of the rule is controlled by /tcb/titlerule →P. 36. It may be set directly to a smaller width to create mixed effects with the standard rule. This option is implemented as an /tcb/underlay →P. 186. Thus, it is not available for standard →P. 193 and standard jigsaw →P. 194, but for all other skins, e.g. enhanced →P. 195. As an underlay, this option can be used multiple times and is removed by /tcb/no underlay →P. 186.

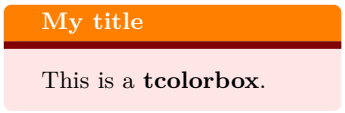
```
\begin{tcolorbox}[enhanced,
  colback=red!5!white,colframe=red!75!black,
  colbacktitle=red!50!yellow,fonttitle=\bfseries,
  title=My title,
  titlerule=1mm,
  titlerule style=yellow  ]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



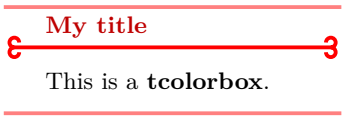

```
\begin{tcolorbox}[enhanced,
  colback=red!5!white,colframe=red!75!black,
  colbacktitle=red!50!yellow,fonttitle=\bfseries,
  title=My title,
  titlerule=1mm,
  titlerule style={yellow,line width=0.5mm}  ]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```




```
\begin{tcolorbox}[enhanced,
  colback=red!10!white,colframe=red!75!black,
  colbacktitle=red!50!yellow,fonttitle=\bfseries,
  frame hidden,
  title=My title,
  boxrule=0pt,titlerule=1mm,
  titlerule style=red!50!black  ]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```




```
%\usetikzlibrary{arrows.meta}
\begin{tcolorbox}[empty,
  coltitle=red!75!black,fonttitle=\bfseries,
  borderline horizontal={0.5mm}{0pt}{red!50!white},
  title=My title,
  titlerule style={red,
    arrows = {Hooks[arc=270]-Hooks[arc=270]}} ]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

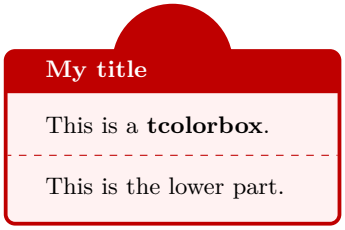


The combined Ti*k*Z style applied to frame, interior, and title background can used by authors in customizing code.

/tikz/tcb fill frame (style, no value)

This is a Ti*k*Z style which is finally applied to the *frame* of the box.

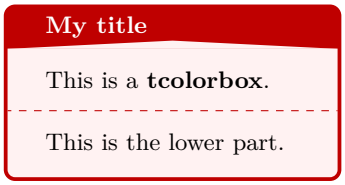
```
% \tcbuselibrary{hooks}
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries,
  frame code app={\path[tcb fill frame]
    ([yshift=-2mm]frame.north)
        circle (8mm); }  }

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



/tikz/tcb fill interior (style, no value)

This is a Ti*k*Z style which is finally applied to the *interior* of the box.

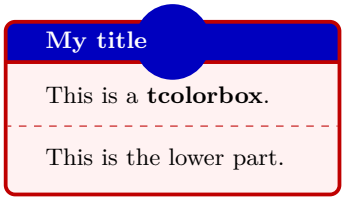
```
% \tcbuselibrary{hooks}
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries,
  interior titled code app={\path[tcb fill interior]
      ([yshift=-0.1pt]interior.north east)
    --([yshift=3pt]interior.north)
    --([yshift=-0.1pt]interior.north west)
    --cycle;}  }

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



/tikz/tcb fill title (style, no value)

This is a Ti*k*Z style which is finally applied to the *title area* of the box.

```
% \tcbuselibrary{hooks}
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries,
  colbacktitle=blue!75!black,
  title code app={\path[tcb fill title]
    (title) circle (5mm); }  }

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



10.2 Boxed Title Option Keys

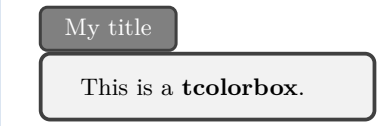
10.2.1 Boxed Title Placement

The following options place the title text into an own \tcbox→P. 14. This boxed title can be customized independently from the main box using /tcb/boxed title style→P. 150. The placement can be influenced by ⟨*boxtitle options*⟩.

/tcb/attach boxed title to top left={⟨*boxtitle options*⟩} (style, default empty)

The title is boxed with a \tcbox→P. 14 and attached to the top left corner of the main box.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top left]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



/tcb/attach boxed title to top center={⟨*boxtitle options*⟩} (style, default empty)

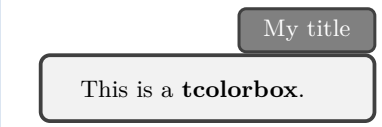
The title is boxed with a \tcbox→P. 14 and attached to the top of the main box.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top center]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

/tcb/attach boxed title to top right={⟨*boxtitle options*⟩} (style, default empty)

The title is boxed with a \tcbox→P. 14 and attached to the top right corner of the main box.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top right]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



/tcb/attach boxed title to bottom left={⟨*boxtitle options*⟩} (style, default empty)

The title is boxed with a \tcbox→P. 14 and attached to the bottom left corner of the main box.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to bottom left]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

/tcb/attach boxed title to bottom center={⟨*boxtitle options*⟩} (style, default empty)

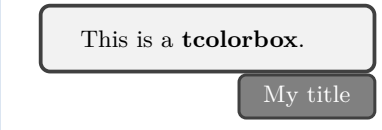
The title is boxed with a \tcbox→P. 14 and attached to the bottom of the main box.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to bottom center]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

/tcb/attach boxed title to bottom right={⟨*boxtitle options*⟩} (style, default empty)

The title is boxed with a \tcbox→P. 14 and attached to the bottom right corner of the main box.

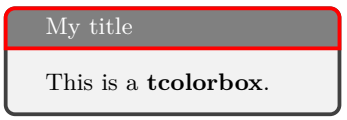
```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to bottom right]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



N 2016-02-26 **/tcb/attach boxed title to top={⟨*boxtitle options*⟩}** (style, default empty)

This is a convenient style to mimic a standard title. It uses /tcb/attach boxed title to top center → P. 146, /tcb/minipage boxed title → P. 154, and sizes the boxed title to match the base box.

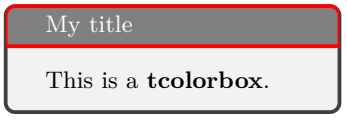
```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top,
  boxed title style={colframe=red}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



N 2016-02-26 **/tcb/attach boxed title to top*={⟨*boxtitle options*⟩}** (style, default empty)

In contrast to /tcb/attach boxed title to top, this style uses smaller left and right rules to avoid previewer glitches. Typically, one would not use different colors for the frame as in the example below.

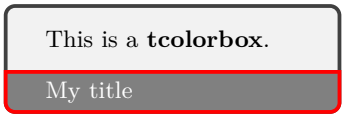
```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top*,
  boxed title style={colframe=red}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



N 2016-02-26 **/tcb/attach boxed title to bottom={⟨*boxtitle options*⟩}** (style, default empty)

This is a convenient style to produce a standard-like title at the bottom of the box. It uses /tcb/attach boxed title to bottom center → P. 146, /tcb/minipage boxed title → P. 154, and sizes the boxed title to match the base box.

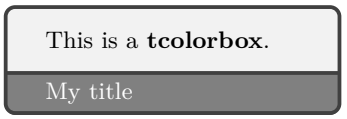
```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to bottom,
  boxed title style={colframe=red}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



N 2016-02-26 **/tcb/attach boxed title to bottom*={⟨*boxtitle options*⟩}** (style, default empty)

In contrast to /tcb/attach boxed title to top, this style uses smaller left and right rules to avoid previewer glitches.

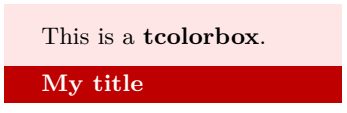
```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to bottom*]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



N 2016-02-26 **/tcb/flip title={⟨*options*⟩}** (style, default empty)

This style combines /tcb/attach boxed title to bottom* with /tcb/boxed title style → P. 150. The ⟨*options*⟩ are given to /tcb/boxed title style → P. 150.

```
\begin{tcolorbox}[tile,flip title={sharp corners},
  title=My title,colback=red!10,
  colbacktitle=red!75!black]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



10.2.2 Options for the Boxed Title Placement

The ⟨*boxtitle options*⟩ of the keys described above are shift values. The dimensions of the boxed title are stored into two macros **\tcboxedtitleheight** and **\tcboxedtitlewidth**. These macros can be used inside the following ⟨*boxtitle options*⟩:

/tcb/boxtitle/xshift=⟨*length*⟩ (no default, initially 0pt)

The boxed title is shifted by ⟨*length*⟩ in the horizontal direction.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top left={xshift=-2mm},
  boxed title style={size=small,colback=blue}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

/tcb/boxtitle/yshift=⟨*length*⟩ (no default, initially 0pt)

The boxed title is shifted by ⟨*length*⟩ in the vertical direction.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top center=
    {yshift=-\tcboxedtitleheight/2},
  boxed title style={size=small,colback=blue}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

/tcb/boxtitle/yshifttext=⟨*length*⟩ (no default, initially 0pt)

The text inside the main box by ⟨*length*⟩ to give room for e. g. a sunken title.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top center=
    {yshift=-3mm,yshifttext=-1mm},
  boxed title style={size=small,colback=blue}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

/tcb/boxtitle/yshift*=⟨*length*⟩ (no default, initially 0pt)

Sets /tcb/boxtitle/yshift and /tcb/boxtitle/yshifttext the same time. /tcb/boxtitle/yshifttext is only set if necessary.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top center={yshift*=-3mm},
  boxed title style={size=small,colback=blue}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

! The bounding box of the resulting total **tcolorbox** is adapted automatically to the *vertical* dimensions of the boxed title. Possible horizontal enlargements are *not* automatically computed.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top left=
    {xshift=-2mm,yshift=-2mm},
  boxed title style={size=small,colback=blue},
  show bounding box]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

10.2.3 Options for the Boxed Title Box

! The boxed title options are implemented as an underlay, see Section 10.8 on page 186. Therefore, a boxed title is not drawn, if a skin does not support underlays like `standard`→P. 193. Still, the room for the boxed titles gets reserved in these cases.

! A TikZ node `title` is produced by a boxed title which can be used inside `/tcb/frame code`→P. 129, `/tcb/interior code`→P. 130, underlays, overlays, and finishes.

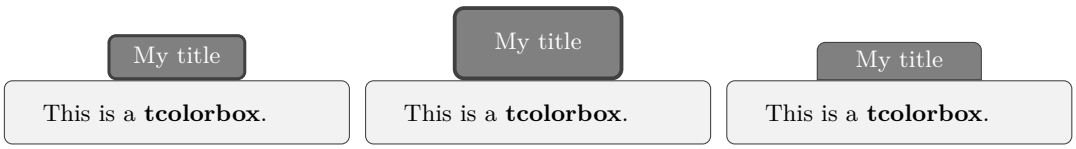
! A boxed title is almost always the first underlay. The only exceptions are underlays defined by `/tcb/underlay boxed title`→P. 187 which are drawn before. Additionally, underlays defined by `/tcb/underlay boxed title`→P. 187 are only drawn, if a boxed title is actually set. They are ignored, if there is no boxed title.

N 2016-02-26 `/tcb/boxed title size=`⟨*size*⟩ (no default, initially `title`)

This setting defines the basic size for the title box. Further settings can be applied using `/tcb/boxed title style`→P. 150. Feasible values for ⟨*size*⟩ are:

- `title`: Sets the size according to `/tcb/size`→P. 43`=title`.
- `standard`: No size setting. Typically, this is identical to `/tcb/size`→P. 43`=normal`.
- `copy`: The size values for a title of the base box are copied for the title box.

```
% \tcbuselibrary{raster}
\begin{tcbraster}[raster columns=3,enhanced,boxrule=0.4pt,
    title=My title,attach boxed title to top center]
  \begin{tcolorbox}[boxed title size=title]
    This is a \textbf{tcolorbox}.
  \end{tcolorbox}
  \begin{tcolorbox}[boxed title size=standard]
    This is a \textbf{tcolorbox}.
  \end{tcolorbox}
  \begin{tcolorbox}[boxed title size=copy]
    This is a \textbf{tcolorbox}.
  \end{tcolorbox}
\end{tcbraster}
```



U 2016-02-26 **/tcb/boxed title style=⟨*options*⟩** (style, initially empty)

By default, a boxed title is dimensioned with /tcb/size →P. 43=title and inherits the /tcb/skin →P. 125 and /tcb/colframe →P. 27 of the main box. Also, the /tcb/colback →P. 27 is inherited from the main /tcb/colbacktitle →P. 27. Font and color of the title text are set as usual. All other ⟨*options*⟩ are set by the /tcb/boxed title style key. Since a boxed title is set by \tcbox →P. 14, all tcolorbox options are applicable here. If /tcb/boxed title style is used several times, the ⟨*options*⟩ are is appended.

```
\begin{tcolorbox}[enhanced,title=My title,
  fonttitle=\bfseries,coltitle=green!25!black,
  attach boxed title to top center=
    {yshift=-2mm,yshifttext=-1mm},
  boxed title style={colframe=green!75!black,
    colback=yellow!50!green}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

```
\begin{tcolorbox}[enhanced,title=My title,
  colframe=red!50!black,colback=red!10!white,
  arc=1mm,colbacktitle=red!10!white,
  fonttitle=\bfseries,coltitle=red!50!black,
  attach boxed title to top left=
    {xshift=3.2mm,yshift=-0.50mm},
  boxed title style={skin=enhancedfirst jigsaw,
    size=small,arc=1mm,bottom=-1mm,
    interior style={fill=none,
      top color=red!30!white,
      bottom color=red!20!white}}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

```
\begin{tcolorbox}[enhanced,title=My title,
  colframe=blue!50!black,colback=blue!10!white,colbacktitle=blue!5!yellow!10!white,
  fonttitle=\bfseries,coltitle=black,attach boxed title to top center=
    {yshift=-0.25mm-\tcboxedtitleheight/2,yshifttext=2mm-\tcboxedtitleheight/2},
  boxed title style={boxrule=0.5mm,
    frame code={ \path[tcb fill frame] ([xshift=-4mm]frame.west)
    -- (frame.north west) -- (frame.north east) -- ([xshift=4mm]frame.east)
    -- (frame.south east) -- (frame.south west) -- cycle; },
    interior code={ \path[tcb fill interior] ([xshift=-2mm]interior.west)
    -- (interior.north west) -- (interior.north east)
    -- ([xshift=2mm]interior.east) -- (interior.south east) -- (interior.south west)
    -- cycle;}  }]
  \lipsum[2]
\end{tcolorbox}
```

My title

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
% \usepackage{varwidth}
\newtcolorbox{mybox}[2][]{enhanced,skin=enhancedlast jigsaw,
  attach boxed title to top left={xshift=-4mm,yshift=-0.5mm},
  fonttitle=\bfseries\sffamily,varwidth boxed title=0.7\linewidth,
  colbacktitle=blue!45!white,colframe=red!50!black,
  interior style={top color=blue!10!white,bottom color=red!10!white},
  boxed title style={empty,arc=0pt,outer arc=0pt,boxrule=0pt},
  underlay boxed title={
    \fill[blue!45!white] (title.north west) -- (title.north east)
      -- +(\tcboxedtitleheight-1mm,-\tcboxedtitleheight+1mm)
      -- ([xshift=4mm,yshift=0.5mm]frame.north east) -- +(0mm,-1mm)
      -- (title.south west) -- cycle;
    \fill[blue!45!white!50!black] ([yshift=-0.5mm]frame.north west)
      -- +(-0.4,0) -- +(0,-0.3) -- cycle;
    \fill[blue!45!white!50!black] ([yshift=-0.5mm]frame.north east)
      -- +(0,-0.3) -- +(0.4,0) -- cycle;  },
  title={#2},#1}

\begin{mybox}{My title}
  \lipsum[2]
\end{mybox}
```

My title

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
% \usepackage{varwidth}
\newtcolorbox{mybox}[2][]{enhanced,
  attach boxed title to top left={xshift=1cm,yshift=-2mm},
  fonttitle=\bfseries,varwidth boxed title=0.7\linewidth,
  colbacktitle=green!45!white,coltitle=green!10!black,colframe=green!50!black,
  interior style={top color=yellow!10!white,bottom color=green!10!white},
  boxed title style={boxrule=0.75mm,colframe=white,
    borderline={0.1mm}{0mm}{green!50!black},
    borderline={0.1mm}{0.75mm}{green!50!black},
    interior style={top color=green!10!white,bottom color=green!10!white,
      middle color=green!50!white},
    drop fuzzy shadow},
  title={#2},#1}

\begin{mybox}{My title}
  \lipsum[2]
\end{mybox}
```

My title

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
\newtcolorbox{flipbox}[2][]{
  enhanced,colframe=blue!50!black,colback=yellow!5,fonttitle=\bfseries,
  flip title={interior hidden},title={#2},#1}

\begin{flipbox}{My title}
\lipsum[2]
\end{flipbox}
```

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

My title

```
% \usepackage{varwidth}
\newtcolorbox{mybox}[2][]{skin=enhancedlast jigsaw,interior hidden,
  boxsep=0pt,top=0pt,colframe=red,coltitle=red!50!black,
  fonttitle=\bfseries\sffamily,
  attach boxed title to bottom center,
  boxed title style={empty,boxrule=0.5mm},
  varwidth boxed title=0.5\linewidth,
  underlay boxed title={
    \draw[white,line width=0.5mm]
      ([xshift=0.3mm-\tcboxedtitleheight*2,yshift=0.3mm]title.north west)
      --([xshift=-0.3mm+\tcboxedtitleheight*2,yshift=0.3mm]title.north east);
    \path[draw=red,top color=white,bottom color=red!50!white,line width=0.5mm]
    ([xshift=0.25mm-\tcboxedtitleheight*2,yshift=0.25mm]title.north west)
    cos +(\tcboxedtitleheight,-\tcboxedtitleheight/2)
    sin +(\tcboxedtitleheight,-\tcboxedtitleheight/2)
    -- ([xshift=0.25mm,yshift=0.25mm]title.south west)
    -- ([yshift=0.25mm]title.south east)
    cos +(\tcboxedtitleheight,\tcboxedtitleheight/2)
    sin +(\tcboxedtitleheight,\tcboxedtitleheight/2); },
  title={#2},#1}

\begin{mybox}{My title}
  \lipsum[2]
\end{mybox}
```

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

My title

```
% \usepackage{varwidth}
\newtcolorbox{mybox}[2][]{enhanced,
  before skip=2mm,after skip=2mm,
  colback=black!5,colframe=black!50,boxrule=0.2mm,
  attach boxed title to top left={xshift=1cm,yshift*=1mm-\tcboxedtitleheight},
  varwidth boxed title*=-3cm,
  boxed title style={frame code={
    \path[fill=tcbcol@back!30!black]
      ([yshift=-1mm,xshift=-1mm]frame.north west)
        arc[start angle=0,end angle=180,radius=1mm]
      ([yshift=-1mm,xshift=1mm]frame.north east)
        arc[start angle=180,end angle=0,radius=1mm];
    \path[left color=tcbcol@back!60!black,right color=tcbcol@back!60!black,
      middle color=tcbcol@back!80!black]
      ([xshift=-2mm]frame.north west) -- ([xshift=2mm]frame.north east)
      [rounded corners=1mm]-- ([xshift=1mm,yshift=-1mm]frame.north east)
      -- (frame.south east) -- (frame.south west)
      -- ([xshift=-1mm,yshift=-1mm]frame.north west)
      [sharp corners]-- cycle;
    },interior engine=empty,
  },
  fonttitle=\bfseries,
  title={#2},#1}

\begin{mybox}[colbacktitle=green]{My title}
\lipsum[2]
\end{mybox}
\begin{mybox}[colbacktitle=red]{My title}
\lipsum[3]
\end{mybox}
```

My title

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

My title

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

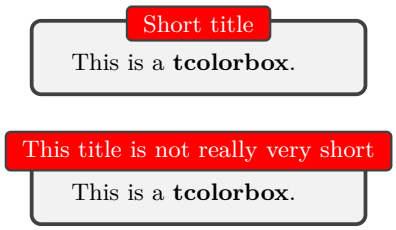
N 2016-02-26 **/tcb/no boxed title style** (style, initially set)

Removes all options which were set by /tcb/boxed title style → P. 150.

/tcb/hbox boxed title (no value, initially set)

The title text content is captured with a horizontal box. Especially, there are no linebreak possible.

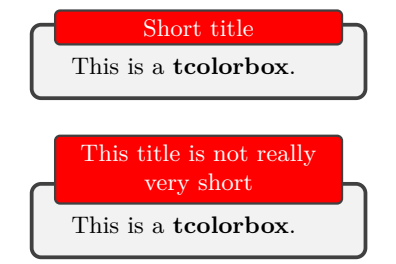
```
\newtcolorbox{mybox}[1]{hbox boxed title,
  enhanced,attach boxed title to top center=
    {yshift=-3mm,yshifttext=-1mm},
  boxed title style={size=small,colback=red},
  title={#1}}

\begin{mybox}{Short title}
  This is a \textbf{tcolorbox}.
\end{mybox}\bigskip

\begin{mybox}{This title is not really very short}
  This is a \textbf{tcolorbox}.
\end{mybox}
```



/tcb/minipage boxed title=⟨*length*⟩ (initially unset)

The title text content is captured with a minipage with a width of ⟨*length*⟩. By default, the resulting boxed title is somewhat smaller than the main box.

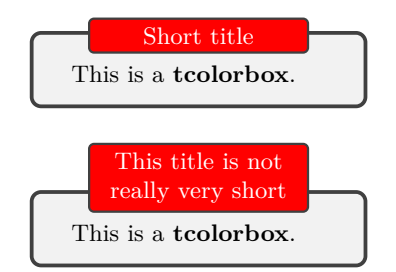
```
\newtcolorbox{mybox}[1]{minipage boxed title,
  enhanced,attach boxed title to top center=
    {yshift=-3mm,yshifttext=-1mm},
  boxed title style={size=small,colback=red},
  center title,title={#1}}

\begin{mybox}{Short title}
  This is a \textbf{tcolorbox}.
\end{mybox}\bigskip

\begin{mybox}{This title is not really very short}
  This is a \textbf{tcolorbox}.
\end{mybox}
```



/tcb/minipage boxed title*=⟨*length*⟩ (initially unset)

The title text content is captured with a minipage with a width of main box width plus ⟨*length*⟩. By default, the resulting boxed title is somewhat smaller than the main box.

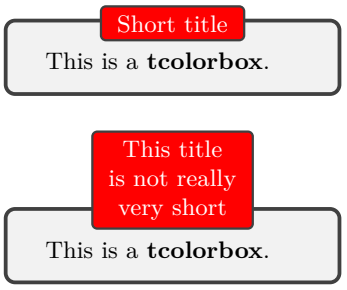
```
\newtcolorbox{mybox}[1]{minipage boxed title*=-2cm,
  enhanced,attach boxed title to top center=
    {yshift=-3mm,yshifttext=-1mm},
  boxed title style={size=small,colback=red},
  center title,title={#1}}

\begin{mybox}{Short title}
  This is a \textbf{tcolorbox}.
\end{mybox}\bigskip

\begin{mybox}{This title is not really very short}
  This is a \textbf{tcolorbox}.
\end{mybox}
```



`/tcb/tikznode boxed title=`⟨*options*⟩ (initially unset)

The title text content is captured with a Ti*k*Z node with given Ti*k*Z ⟨*options*⟩. The text is centered by default

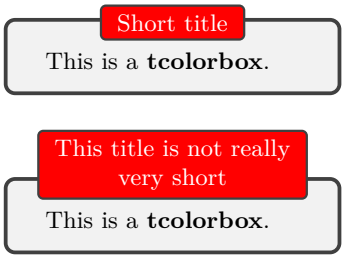
```
\newtcolorbox{mybox}[1]{tikznode boxed title,
  enhanced,attach boxed title to top center=
    {yshift=-3mm,yshifttext=-1mm},
  boxed title style={size=small,colback=red},
  title={#1}}

\begin{mybox}{Short title}
  This is a \textbf{tcolorbox}.
\end{mybox}\bigskip

\begin{mybox}{This title\\is not really\\very short}
  This is a \textbf{tcolorbox}.
\end{mybox}
```



`/tcb/varwidth boxed title=`⟨*length*⟩ (initially unset)

The title text content is captured with a `varwidth` environment with a width of ⟨*length*⟩. This style needs the `varwidth` package [1] to be loaded manually. By default, the resulting boxed title is somewhat smaller than the main box.

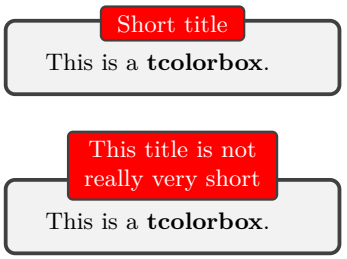
```
% \usepackage{varwidth}
\newtcolorbox{mybox}[1]{varwidth boxed title,
  enhanced,attach boxed title to top center=
    {yshift=-3mm,yshifttext=-1mm},
  boxed title style={size=small,colback=red},
  center title,title={#1}}

\begin{mybox}{Short title}
  This is a \textbf{tcolorbox}.
\end{mybox}\bigskip

\begin{mybox}{This title is not really very short}
  This is a \textbf{tcolorbox}.
\end{mybox}
```



`/tcb/varwidth boxed title*=`⟨*length*⟩ (initially unset)

The title text content is captured with a `varwidth` environment with a width of main box width plus ⟨*length*⟩. This style needs the `varwidth` package [1] to be loaded manually. By default, the resulting boxed title is somewhat smaller than the main box.

```
% \usepackage{varwidth}
\newtcolorbox{mybox}[1]{varwidth boxed title*=-2cm,
  enhanced,attach boxed title to top center=
    {yshift=-3mm,yshifttext=-1mm},
  boxed title style={size=small,colback=red},
  center title,title={#1}}

\begin{mybox}{Short title}
  This is a \textbf{tcolorbox}.
\end{mybox}\bigskip

\begin{mybox}{This title is not really very short}
  This is a \textbf{tcolorbox}.
\end{mybox}
```



10.3 Watermark Option Keys

The following watermark options are applicable for all skins which use `tikzpicture` as `/tcb/graphical environment`→P. 125. Therefore, the skin `standard`→P. 193 does not support these watermarks, but all other skins, e. g. `enhanced`→P. 195.

> ! The watermark options rely on the more general overlay options described in Section 4.12 from page 69. Therefore, *watermarks* and *overlays* cannot be used mixed. But a mixture is possible with the LIB hooks library, see Section 20.

`/tcb/watermark text=`⟨*text*⟩ (no default, initially unset)

Writes some ⟨*text*⟩ in the center of the interior region of a `tcolorbox`. This ⟨*text*⟩ is written *after* the frame and interior are drawn and *before* the text content is drawn. It is zoomed or stretched according the values of `/tcb/watermark zoom`→P. 159 or `/tcb/watermark stretch`→P. 161.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,watermark text=My Watermark]
\lipsum[1]
\tcblower
\lipsum[2]
\end{tcolorbox}
```

My title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

My Watermark

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/watermark text on=`⟨*part*⟩ `is` ⟨*text*⟩ (no default, initially unset)

This option writes some ⟨*text*⟩ in the center of the interior region of a `tcolorbox` as described for `/tcb/watermark text`. But this is done only for boxes named ⟨*part*⟩ of a break sequence, see `/tcb/breakable`→P. 351.

Feasible values for ⟨*part*⟩ are:

- `broken`: all broken box parts,
- `unbroken`: unbroken boxes only,
- `first`: first parts of a break sequence,
- `middle`: middle parts of a break sequence,
- `last`: last parts of a break sequence,
- `unbroken and first`: unbroken boxes and first parts of a break sequence,
- `middle and last`: middle and last parts of a break sequence.
- `first and middle`: first and middle parts of a break sequence.

`/tcb/watermark graphics`=⟨*file name*⟩ (no default, initially unset)

Draws an external picture referenced by ⟨*file name*⟩ in the center of the interior region of a `tcolorbox`. The picture is drawn *after* the frame and interior are drawn and *before* the text content is drawn. It is zoomed or stretched according the values of `/tcb/watermark zoom` → P. 159 or `/tcb/watermark stretch` → P. 161.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,watermark graphics=Basilica_5.png,
  watermark opacity=0.15]
\lipsum[1-2]
\tcblower
This example uses a public domain picture from\\
\url{http://commons.wikimedia.org/wiki/File:Basilica_5.png}
\end{tcolorbox}
```

My title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

This example uses a public domain picture from
http://commons.wikimedia.org/wiki/File:Basilica_5.png

`/tcb/watermark graphics on`=⟨*part*⟩ `is` ⟨*file name*⟩ (no default, initially unset)

This option draws a picture referenced by ⟨*file name*⟩ in the center of the interior region of a `tcolorbox` as described for `/tcb/watermark graphics`. But this is done only for boxes named ⟨*part*⟩ of a break sequence, see `/tcb/breakable` → P. 351.

Feasible values for ⟨*part*⟩ are:

- `broken`: all broken box parts,
- `unbroken`: unbroken boxes only,
- `first`: first parts of a break sequence,
- `middle`: middle parts of a break sequence,
- `last`: last parts of a break sequence,
- `unbroken and first`: unbroken boxes and first parts of a break sequence,
- `middle and last`: middle and last parts of a break sequence.

`/tcb/watermark tikz=`⟨*graphical code*⟩ (no default, initially unset)

Draws the given `tikz` ⟨*graphical code*⟩ in the center of the interior region of a `tcolorbox`. The code is executed *after* the frame and interior are drawn and *before* the text content is drawn. The result is zoomed or stretched according the values of `/tcb/watermark zoom` → P. 159 or `/tcb/watermark stretch` → P. 161.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  watermark tikz={\draw[line width=2mm] circle (1cm)
    node{\fontfamily{ptm}\fontseries{b}\fontsize{20mm}{20mm}\selectfont ?};}]
\lipsum[1]
\tcblower
\lipsum[2]
\end{tcolorbox}
```

My title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/watermark tikz on=`⟨*part*⟩ `is` ⟨*graphical code*⟩ (no default, initially unset)

This option draws the given `tikz` ⟨*graphical code*⟩ in the center of the interior region of a `tcolorbox` as described for `/tcb/watermark tikz`. But this is done only for boxes named ⟨*part*⟩ of a break sequence, see `/tcb/breakable` → P. 351.

Feasible values for ⟨*part*⟩ are:

- `broken`: all broken box parts,
- `unbroken`: unbroken boxes only,
- `first`: first parts of a break sequence,
- `middle`: middle parts of a break sequence,
- `last`: last parts of a break sequence,
- `unbroken and first`: unbroken boxes and first parts of a break sequence,
- `middle and last`: middle and last parts of a break sequence.

`/tcb/no watermark` (style, no default, initially set)

Removes the watermark if set before. This is an alias for `/tcb/no overlay` → P. 70.

`/tcb/watermark opacity=`⟨*fraction*⟩ (no default, initially `1.00`)

Sets the opacity value ∈ [0, 1] for a watermark.

```
\tcbset{enhanced,colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  watermark text=Watermark,nobeforeafter,width=(\linewidth-2mm)/2}

\begin{tcolorbox}[title=Opacity 1.00,watermark opacity=1.00]
\lipsum[2]
\end{tcolorbox}\hfill%
\begin{tcolorbox}[title=Opacity 0.50,watermark opacity=0.50]
\lipsum[2]
\end{tcolorbox}%
```

Opacity 1.00

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Opacity 0.50

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/watermark zoom=`⟨*fraction*⟩ (no default, initially `0.75`)

Sets the zoom value for a watermark. The zoom respects the aspect ratio. The value 1.0 means to fill the whole box until the watermark touches the frame.

```
\tcbset{enhanced,colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  watermark text=Watermark,nobeforeafter,width=(\linewidth-2mm)/2}

\begin{tcolorbox}[title=Zoom 1.0,watermark zoom=1.0]
\lipsum[2]
\end{tcolorbox}\hfill%
\begin{tcolorbox}[title=Zoom 0.5,watermark zoom=0.5]
\lipsum[2]
\end{tcolorbox}%
```

Zoom 1.0

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Zoom 0.5

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/watermark shrink`=⟨*fraction*⟩ (no default, initially unset)

Identically to `/tcb/watermark zoom` → P. 159, but the watermark never gets enlarged. Thus, the watermark keeps its original size or is shrunk.

`/tcb/watermark overzoom`=⟨*fraction*⟩ (no default, initially unset)

Sets the overzoom value for a watermark. The overzoom respects the aspect ratio. The value 1.0 means to fill the whole box until the watermark touches all four sides of the frame.

```
\tcbset{enhanced,colback=white,colframe=blue!50!black,fonttitle=\bfseries,
  watermark opacity=0.5,
  watermark graphics=lichtspiel.jpg,nobeforeafter,width=(\linewidth-2mm)/2}

\begin{tcolorbox}[title=Zoom 1.0,watermark zoom=1.0]
\lipsum[1]
\end{tcolorbox}\hfill%
\begin{tcolorbox}[title=Overzoom 1.0,watermark overzoom=1.0]
\lipsum[1]
\end{tcolorbox}%
```

Zoom 1.0

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Overzoom 1.0

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

! If a `/tcb/watermark overzoom` value of `1.0` is used in connection with invisible top and bottom rules which still have a thickness greater than `0pt`, the space of these invisible rules may not be covered by the watermark. For example, this situation may occur during the breaking of `/tcb/enhanced` → P. 195 boxes. To avoid this optical glitch, just set `/tcb/pad at break` → P. 355 to any desired value.

`/tcb/watermark stretch=`⟨*fraction*⟩ (no default, initially unset)

Sets the stretch value for a watermark. The stretch value is applied to width and height in relation to the box dimensions. It does not respect the aspect ratio. The value 1.0 means to fill the whole box.

```
\tcbset{enhanced,colback=white,colframe=blue!50!black,fonttitle=\bfseries,
  watermark graphics=lichtspiel.jpg,watermark opacity=0.5,
  nobeforeafter,width=(\linewidth-2mm)/2}

\begin{tcolorbox}[title=Stretch 1.00,watermark stretch=1.00]
\lipsum[2]
\end{tcolorbox}\hfill%
\begin{tcolorbox}[title=Stretch 0.50,watermark stretch=0.50]
\lipsum[2]
\end{tcolorbox}%
```

Stretch 1.00

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Stretch 0.50

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/watermark color=`⟨*color*⟩ (no default, initially mixed background and frame color)

Sets the color for the watermark.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,watermark text=My Watermark,
  watermark color=yellow!50!red]
\lipsum[1]
\end{tcolorbox}
```

My title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

`/tcb/clip watermark=true|false` (default `true`, initially `true`)

Sets the watermark to be clipped to the interior area.

```
\tcbset{enhanced,colback=white,colframe=blue!50!white,fonttitle=\bfseries,
  watermark opacity=0.5,watermark stretch=1.00,arc=3mm,
  watermark graphics=lichtspiel.jpg}

\begin{tcolorbox}[title=Clip (default),clip watermark]
\lipsum[1]
\end{tcolorbox}

\begin{tcolorbox}[title=No clip,clip watermark=false]
\lipsum[1]
\end{tcolorbox}%
```

Clip (default)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

No clip

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

10.4 Clip Environments

The following clip environments are applicable for all skins which use engines of type `path`, `pathfirst`, `pathmiddle`, or `pathlast`. Especially, the skin `enhanced` → P. 195 supports *all* of them and `standard` → P. 193 *none*. The typical area of application is inside overlay code, see Section 4.12 from page 69.

```
\begin{tcbclipframe}
  ⟨environment content⟩
\end{tcbclipframe}
```

Defines a `Tikz` scope which clips to the frame area path.

```
\makeatletter
\newtcolorbox{picturebox}[2][]{%
  enhanced,frame hidden,interior hidden,fonttitle=\bfseries,
  overlay={\begin{tcbclipframe}\node at (frame)
    {\includegraphics[width=\tcb@width,height=\tcb@height]{#2}};\end{tcbclipframe}%
    \begin{tcbclipinterior}\fill[white,opacity=0.75]
    (frame.south west) rectangle (frame.north east);\end{tcbclipinterior}},#1}
\makeatother

\begin{picturebox}[title=My Picture Box]{lichtspiel.jpg}
\lipsum[1]
\end{picturebox}
```

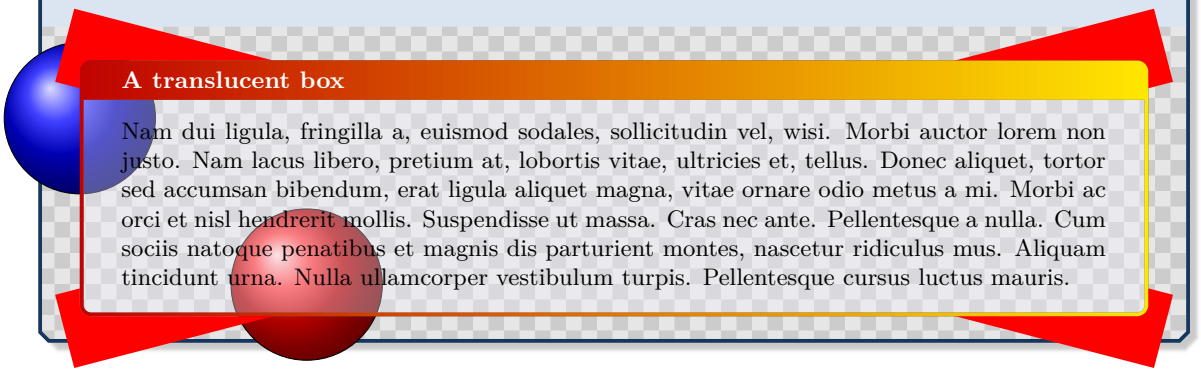
My Picture Box

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

```
\begin{tcbinvclipframe}
 ⟨environment content⟩
\end{tcbinvclipframe}
```

Defines a Tikz scope which clips to the *outside* of the frame area path.

```
\tcbset{enhanced jigsaw,fonttitle=\bfseries,opacityback=0.35,colback=blue!5!white,
  frame style={left color=red!75!black,right color=red!10!yellow}}

\begin{tikzpicture}% draw two balls
  \path[use as bounding box] (0,0.8) rectangle +(0.1,0.1);
  \shadedraw [shading=ball] (0,0) circle (1cm);
  \shadedraw [ball color=red] (3,-2.2) circle (1cm);
\end{tikzpicture}

\begin{tcolorbox}[title=A translucent box,
  overlay={\begin{tcbinvclipframe}
    \draw[red,line width=1cm] ([xshift=-2mm,yshift=2mm]frame.north west)
      --([xshift=2mm,yshift=-2mm]frame.south east);
    \draw[red,line width=1cm] ([xshift=-2mm,yshift=-2mm]frame.south west)
      --([xshift=2mm,yshift=2mm]frame.north east);
  \end{tcbinvclipframe}}]
  \lipsum[2]
\end{tcolorbox}
```



```
\begin{tcbclipinterior}
 ⟨environment content⟩
\end{tcbclipinterior}
```

Defines a `Tikz` scope which clips to the interior area path.

```
\begin{tcolorbox}[enhanced,title=My Title,
  overlay={\begin{tcbclipinterior}
    \draw[red,line width=1cm] (interior.north west)--(interior.south east);
    \draw[red,line width=1cm] (interior.south west)--(interior.north east);
  \end{tcbclipinterior}}]
\lipsum[1]
\end{tcolorbox}
```

My Title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

```
\begin{tcbcliptitle}
 ⟨environment content⟩
\end{tcbcliptitle}
```

Defines a `Tikz` scope which clips to the title area path.

```
\begin{tcolorbox}[enhanced,title=My Title,colframe=blue,colback=yellow!10!white,
  overlay={\begin{tcbcliptitle}\node at (title)
  {\includegraphics[width=\linewidth]{lichtspiel.jpg}};\end{tcbcliptitle}}]
\lipsum[1]
\end{tcolorbox}
```

My Title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

`/tcb/clip title=true|false` (default `true`, initially `false`)

Sets the title to be clipped to the title area.

```
\tcbset{enhanced,width=5cm,colframe=red!50!white,coltitle=black,
  colbacktitle=yellow!50!white}

\begin{tcolorbox}[title=\mbox{This is a title which is unbreakable and far too long}]
This is a tcolorbox.
\end{tcolorbox}

\begin{tcolorbox}[title=\mbox{This is a title which is unbreakable and far too long},
  clip title]
This is a tcolorbox.
\end{tcolorbox}
```

This is a title which is unbreakable and far too long

This is a tcolorbox.

This is a title which is unbreak

This is a tcolorbox.

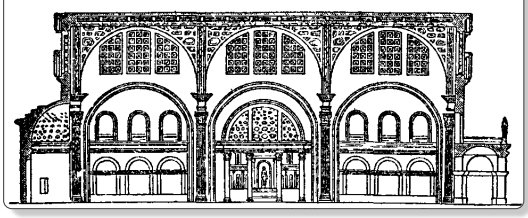
`/tcb/clip upper=true|false` (default `true`, initially `false`)

Sets the upper part to be clipped to the interior area.

```
\newcommand{\mygraphics}[2][]{%
  \tcbox[enhanced,boxsep=0pt,top=0pt,bottom=0pt,left=0pt,
    right=0pt,boxrule=0.4pt,drop fuzzy shadow,clip upper,
    colback=black!75!white,toptitle=2pt,bottomtitle=2pt,nobeforeafter,
    center title,fonttitle=\small\sffamily,title=\detokenize{#2}]
  {\includegraphics[width=\the\dimexpr(\linewidth-4mm)/2\relax]{#2}}}

\mygraphics{lichtspiel.jpg}\hfill
\mygraphics{Basilica_5.png}
```

lichtspiel.jpg

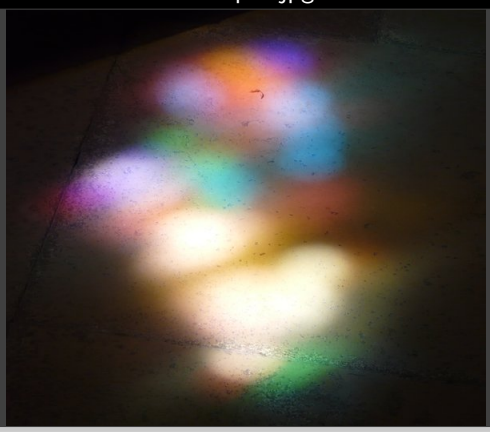
Basilica_5.png



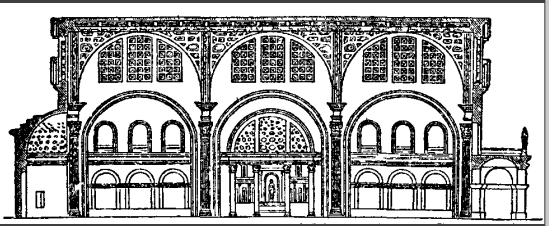
The example for /tcb/clip upper → P. 166 sizes the box according to the dimensions of the picture. To do it the other way around, the watermark options provide an easy solution.

```
\newcommand{\mygraphics}[2][]{%
  \tcbox[enhanced,capture=minipage,boxsep=0pt,top=0pt,bottom=0pt,left=0pt,
    right=0pt,boxrule=0.4pt,drop fuzzy shadow,nobeforeafter,
    colback=black!75!white,toptitle=2pt,bottomtitle=2pt,
    center title,fonttitle=\small\sffamily,title=\detokenize{#2},
    width=(\linewidth-4mm)/2,height=6cm,colbacktitle={black},
    watermark zoom=1.0,watermark graphics={#2}]{}}

\mygraphics{lichtspiel.jpg}\hfill
\mygraphics{Basilica_5.png}
```

lichtspiel.jpg



Basilica_5.png



/tcb/clip lower=true|false (default true, initially false)

Sets the lower part to be clipped to the interior area.

```
\tcbset{enhanced,width=5cm,colframe=red!50!black,text and listing}

\begin{tcblisting}{}
Donau\-dampf\-schiff\-fahrts\-ka\-pi\-t\"ans\-m\"ut\-zen\-fran\-sen
\end{tcblisting}

\begin{tcblisting}{clip lower}
Donau\-dampf\-schiff\-fahrts\-ka\-pi\-t\"ans\-m\"ut\-zen\-fran\-sen
\end{tcblisting}
```

Donaudampfschifffahrtska-
pitänsmützenfransen

```
Donau\-dampf\-schiff\-fahrts\-ka\-pi\-t\"ans\-m\"ut\-zen\-fran\-sen
```

Donaudampfschifffahrtska-
pitänsmützenfransen

```
Donau\-dampf\-schiff\-fa
```

10.5 Border Line Option Keys

The following borderline options are applicable for most skins which use `tikzpicture` as `/tcb/graphical environment`→ P. 125. Therefore, the skin `standard`→ P. 193 does not support these border lines, but most other skins, e. g. `enhanced`→ P. 195.

The borderlines are independent from the normal `tcolorbox` rules. They may be used with or without the `/tcb/segmentation engine`→ P. 127.

The borderlines are stackable, i. e. several different border lines can be used on the same `tcolorbox`. They are drawn *after* the box frame and box interior and *before* overlays or watermarks.

> **!** Technically, the normal `tcolorbox` rules result from a TikZ *filling* process. The border lines are created by a TikZ *drawing* process. This can be used to apply different effects.

`/tcb/borderline={`⟨*width*⟩`}{`⟨*offset*⟩`}{`⟨*options*⟩`}` (no default, initially unset)

Adds a new borderline to the stack of border lines. This border line is drawn with the given ⟨*width*⟩ and gets an ⟨*offset*⟩ computed from the frame outline. A positive ⟨*offset*⟩ value moves the borderline inside the `tcolorbox` and a negative ⟨*offset*⟩ value moves it outside without changing the bounding box.

The border line is drawn along a TikZ path with the given TikZ ⟨*options*⟩. Note that the TikZ `line width` option should not be used here.

The border lines adapt to the rounded corners of the `tcolorbox`. An inside borderline will switch to sharp corners if necessary, an outside borderline will always be rounded except for `/tcb/sharp corners`→ P. 47.

```
\begin{tcolorbox}[enhanced,title=Rounded corners,fonttitle=\bfseries,boxsep=5pt,
  arc=8pt,
  borderline={0.5pt}{0pt}{red},
  borderline={0.5pt}{5pt}{blue,dotted},
  borderline={0.5pt}{-5pt}{green} ]
This is a tcolorbox.
\end{tcolorbox}
\bigskip
\begin{tcolorbox}[enhanced,title=Sharp corners,fonttitle=\bfseries,boxsep=5pt,
  arc=8pt,sharp corners=downhill,
  borderline={0.5pt}{0pt}{red},
  borderline={0.5pt}{5pt}{blue,dotted},
  borderline={0.5pt}{-5pt}{green} ]
This is a tcolorbox.
\end{tcolorbox}
```

Rounded corners

This is a tcolorbox.

Sharp corners

This is a tcolorbox.

```
% \usepackage{lipsum}
\begin{tcolorbox}[enhanced,arc=3mm,boxrule=1.5mm,boxsep=1.5mm,
  colback=yellow!20!white,
  colframe=blue,
  borderline={1mm}{1mm}{white},
  borderline={1mm}{2mm}{red} ]
  \lipsum[1]
\end{tcolorbox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

```
% \usepackage{lipsum}
\begin{tcolorbox}[enhanced,arc=3mm,boxrule=1.5mm,
  frame hidden,colback=blue!10!white,
  borderline={1mm}{0mm}{blue,dotted} ]
  \lipsum[2]
\end{tcolorbox}
```

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
% \usepackage{lipsum}
\begin{tcolorbox}[enhanced,skin=enhancedmiddle,
  frame hidden,interior hidden,top=0mm,bottom=0mm,boxsep=0mm,
  borderline={0.75mm}{0mm}{red},
  borderline={0.75mm}{0.75mm}{red!50!yellow},
  borderline={0.75mm}{1.5mm}{yellow}, ]
  \lipsum[3]
\end{tcolorbox}
```

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

```
% \usepackage{lipsum}
\newtcolorbox{mygreenbox}[2][]{%
  enhanced,width=\linewidth-6pt,
  enlarge top by=3pt,enlarge bottom by=3pt,
  enlarge left by=3pt,enlarge right by=3pt,
  title={#2},frame hidden,boxrule=0pt,top=1mm,bottom=1mm,
  colframe=green!30!black, colbacktitle=green!50!yellow,
  coltitle=black, colback=green!25!white,
  borderline={0.5pt}{-0.5pt}{green!75!blue},
  borderline={1pt}{-3pt}{green!50!blue},#1}

\begin{mygreenbox}{My title}
  \lipsum[4]
\end{mygreenbox}
```

My title

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

`/tcb/no borderline` (no default, initially set)

Removes all borderlines if set before.

`/tcb/show bounding box=`⟨*color*⟩ (default `red`, initially unset)

Displays the bounding box borderline of a `tcolorbox`. Its intended use is debugging and fine tuning. It should not be part of a final document. The optional ⟨*color*⟩ is the base color for the bounding box borderline.

```
\tcbset{enhanced,nobeforeafter,width=4cm,fonttitle=\bfseries}

\begin{tcolorbox}[show bounding box,title=Normal]
This is a tcolorbox.
\end{tcolorbox}%
\begin{tcolorbox}[show bounding box=blue,title=Shadow,drop fuzzy shadow]
This is a tcolorbox.
\end{tcolorbox}%
\begin{tcolorbox}[show bounding box=green,title=Enlarged,drop fuzzy shadow,
  enlarge by=2mm]
This is a tcolorbox.
\end{tcolorbox}
```

Normal

This is a tcolorbox.

Shadow

This is a tcolorbox.

Enlarged

This is a tcolorbox.

The following *partial* borderlines act slightly different from the complete borderlines described before. They ignore rounded corner settings, their length is not modified by their ⟨*offset*⟩, they ignore skin settings but adapt to breakable boxes.

N 2014-10-20 **/tcb/borderline north={⟨*width*⟩}{⟨*offset*⟩}{⟨*options*⟩}** (no default, initially unset)

Adds a new borderline with the given ⟨*width*⟩ to the north of the `tcolorbox`. A positive ⟨*offset*⟩ value moves the borderline inside the `tcolorbox` and a negative ⟨*offset*⟩ value moves it outside without changing the bounding box.

```
\begin{tcolorbox}[enhanced,
  borderline north={2pt}{-2pt}{red}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

N 2014-10-20 **/tcb/borderline south={⟨*width*⟩}{⟨*offset*⟩}{⟨*options*⟩}** (no default, initially unset)

Adds a new borderline with the given ⟨*width*⟩ to the south of the `tcolorbox`. A positive ⟨*offset*⟩ value moves the borderline inside the `tcolorbox` and a negative ⟨*offset*⟩ value moves it outside without changing the bounding box.

```
\begin{tcolorbox}[enhanced,
  borderline south={2pt}{-2pt}{red}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

N 2014-10-20 **/tcb/borderline east={⟨*width*⟩}{⟨*offset*⟩}{⟨*options*⟩}** (no default, initially unset)

Adds a new borderline with the given ⟨*width*⟩ to the east of the `tcolorbox`. A positive ⟨*offset*⟩ value moves the borderline inside the `tcolorbox` and a negative ⟨*offset*⟩ value moves it outside without changing the bounding box.

```
\begin{tcolorbox}[enhanced,
  borderline east={2pt}{-2pt}{red}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

N 2014-10-20 **/tcb/borderline west={⟨*width*⟩}{⟨*offset*⟩}{⟨*options*⟩}** (no default, initially unset)

Adds a new borderline with the given ⟨*width*⟩ to the west of the `tcolorbox`. A positive ⟨*offset*⟩ value moves the borderline inside the `tcolorbox` and a negative ⟨*offset*⟩ value moves it outside without changing the bounding box.

```
\begin{tcolorbox}[enhanced,
  borderline west={2pt}{-2pt}{red}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

N 2014-10-20 **/tcb/borderline horizontal={⟨*width*⟩}{⟨*offset*⟩}{⟨*options*⟩}** (no default, initially unset)

Adds a new borderline with the given ⟨*width*⟩ to the north and south of the `tcolorbox`. A positive ⟨*offset*⟩ value moves the borderlines inside the `tcolorbox` and a negative ⟨*offset*⟩ value moves them outside without changing the bounding box.

```
\begin{tcolorbox}[blanker,top=3mm,bottom=3mm,
   borderline horizontal={2pt}{0pt}{red}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

N 2014-10-20 **/tcb/borderline vertical={⟨*width*⟩}{⟨*offset*⟩}{⟨*options*⟩}** (no default, initially unset)

Adds a new borderline with the given ⟨*width*⟩ to the east and west of the `tcolorbox`. A positive ⟨*offset*⟩ value moves the borderlines inside the `tcolorbox` and a negative ⟨*offset*⟩ value moves them outside without changing the bounding box.

```
\begin{tcolorbox}[blanker,left=3mm,right=3mm,
   borderline vertical={2pt}{0pt}{red}]
  This is a \textbf{tcolorbox}.\\
  My second line.
\end{tcolorbox}
```

This is a **tcolorbox**.
My second line.

```
\begin{tcolorbox}[enhanced,colback=yellow!10!white,boxrule=0pt,frame hidden,
  borderline north={1mm}{-2mm}{red},
  borderline south={1mm}{-2mm}{blue},
  borderline west={1mm}{-2mm}{green},
  borderline east={1mm}{-2mm}{yellow}]
\lipsum[1]
\end{tcolorbox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

10.6 Shadow Option Keys

The following shadow options are applicable for most skins which use `tikzpicture` as `/tcb/graphical environment` → P. 125. Therefore, the skin `standard` → P. 193 does not support these shadows, but most other skins, e. g. `enhanced` → P. 195.

The shadows are stackable, i. e. several different shadows can be used on the same `tcolorbox`. They are drawn *before* the box frame is drawn.

`/tcb/no shadow` (no default)

Removes all shadows if set before.

10.6.1 Common Shadows and Halos

`/tcb/drop shadow=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows. Optionally, the ⟨*color*⟩ for the shadow can be changed.

```
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[drop shadow]
This is a tcolorbox.
\end{tcolorbox}\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  drop shadow=blue]
This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

Another shadow

This is a tcolorbox.

`/tcb/drop fuzzy shadow=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows. Optionally, the ⟨*color*⟩ for the shadow can be changed.

```
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[drop fuzzy shadow]
This is a tcolorbox.
\end{tcolorbox}\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  drop fuzzy shadow=blue]
This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

Another shadow

This is a tcolorbox.

`/tcb/drop midday shadow=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows. Optionally, the ⟨*color*⟩ for the shadow can be changed.

```
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[drop midday shadow]
This is a tcolorbox.
\end{tcolorbox}\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  drop midday shadow=blue]
This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

Another shadow

This is a tcolorbox.

`/tcb/drop fuzzy midday shadow=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows. Optionally, the ⟨*color*⟩ for the shadow can be changed.

```
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[drop fuzzy midday shadow]
This is a tcolorbox.
\end{tcolorbox}\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  drop fuzzy midday shadow=blue]
This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

Another shadow

This is a tcolorbox.

`/tcb/halo=`⟨*size*⟩ `with` ⟨*color*⟩ (style, default `0.9mm with yellow`)

Adds a new halo shadow with the given ⟨*color*⟩ which overlaps the colorbox an all sides by ⟨*size*⟩.

```
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My own halo,halo]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip\bigskip
\begin{tcolorbox}[title=Another halo,
  halo=2mm with green]
This is a tcolorbox.
\end{tcolorbox}
```

My own halo

This is a tcolorbox.

Another halo

This is a tcolorbox.

`/tcb/fuzzy halo=`⟨*size*⟩ `with` ⟨*color*⟩ (style, default `0.9mm with yellow`)

Adds a new fuzzy halo shadow with the given ⟨*color*⟩ which overlaps the colorbox an all sides by ⟨*size*⟩ plus `0.48mm`.

```
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My own halo,fuzzy halo]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip\bigskip
\begin{tcolorbox}[title=Another halo,
  fuzzy halo=2mm with green]
This is a tcolorbox.
\end{tcolorbox}
```

My own halo

This is a tcolorbox.

Another halo

This is a tcolorbox.

```
\begin{tcolorbox}[blank,enhanced jigsaw,boxsep=2pt,arc=2pt,
  fuzzy halo=2mm with red!50!white,
  fuzzy halo=1mm with white]
\lipsum[1]
\end{tcolorbox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

For all following shadows, the optionally given ⟨*color*⟩ for the shadow can be changed equivalent to the preceding examples.

`/tcb/drop shadow southeast=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows. This shadow is identical to `/tcb/drop shadow` → P. 173.

```
\begin{tcolorbox}[drop shadow southeast,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop shadow south=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows. This shadow is identical to `/tcb/drop midday shadow` → P. 173.

```
\begin{tcolorbox}[drop shadow south,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop shadow southwest=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop shadow southwest,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop shadow west=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop shadow west,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop shadow northwest=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop shadow northwest,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop shadow north=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop shadow north,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop shadow northeast=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop shadow northeast,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop shadow east=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop shadow east,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop fuzzy shadow southeast=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows. This shadow is identical to `/tcb/drop fuzzy shadow` → P. 173.

```
\begin{tcolorbox}[drop fuzzy shadow southeast,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop fuzzy shadow south=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows. This shadow is identical to `/tcb/drop fuzzy midday shadow` → P. 174.

```
\begin{tcolorbox}[drop fuzzy shadow south,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop fuzzy shadow southwest=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop fuzzy shadow southwest,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop fuzzy shadow west=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop fuzzy shadow west,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop fuzzy shadow northwest=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop fuzzy shadow northwest,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop fuzzy shadow north=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop fuzzy shadow north,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop fuzzy shadow northeast=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop fuzzy shadow northeast,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop fuzzy shadow east=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop fuzzy shadow east,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

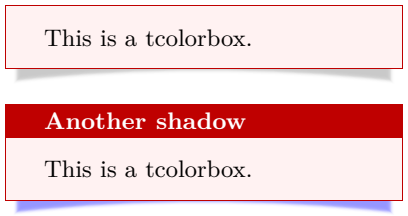
This is a tcolorbox.

10.6.2 Lifted Shadows

`/tcb/drop lifted shadow=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new lifted shadow with standard dimensions to the stack of shadows. Optionally, the ⟨*color*⟩ for the shadow can be changed.

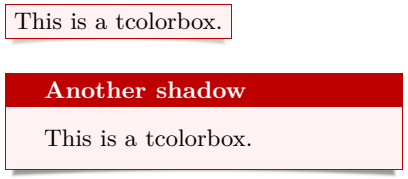
```
\tcbset{enhanced,colback=red!5!white,
  boxrule=0.4pt,sharp corners,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[drop lifted shadow]
This is a tcolorbox.
\end{tcolorbox}\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  drop lifted shadow=blue]
This is a tcolorbox.
\end{tcolorbox}
```



`/tcb/drop small lifted shadow=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new small lifted shadow with standard dimensions to the stack of shadows. Optionally, the ⟨*color*⟩ for the shadow can be changed.

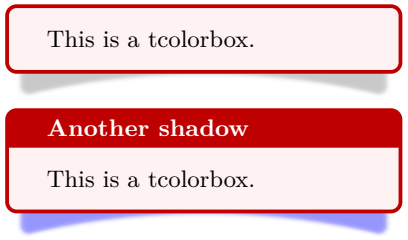
```
\tcbset{enhanced,colback=red!5!white,
  boxrule=0.4pt,sharp corners,
  colframe=red!75!black,fonttitle=\bfseries}

\tcbox[drop small lifted shadow,size=fbox]
  {This is a tcolorbox.}
\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  drop small lifted shadow=black]
This is a tcolorbox.
\end{tcolorbox}
```



`/tcb/drop large lifted shadow=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new large lifted shadow with standard dimensions to the stack of shadows. Optionally, the ⟨*color*⟩ for the shadow can be changed.

```
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[drop large lifted shadow]
This is a tcolorbox.
\end{tcolorbox}\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  drop large lifted shadow=blue]
This is a tcolorbox.
\end{tcolorbox}
```



10.6.3 Generic Shadows

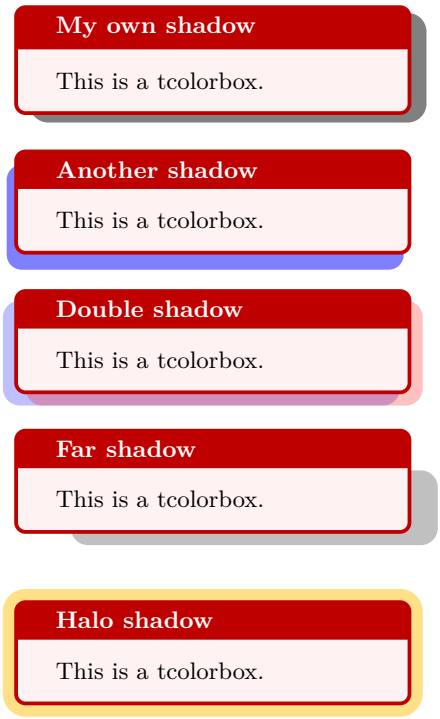
`/tcb/shadow={⟨xshift⟩}{⟨yshift⟩}{⟨offset⟩}{⟨options⟩}` (no default)

Adds a new shadow to the stack of shadows. This shadow follows the outline of the `tcolorbox` but is shifted by ⟨*xshift*⟩ and ⟨*yshift*⟩. The ⟨*offset*⟩ value is a distance value from the frame outline. A positive ⟨*offset*⟩ value shrinks the shadow and a negative ⟨*offset*⟩ value enlarges the shadow. The shadow is filled along a Ti*k*Z path with the given Ti*k*Z ⟨*options*⟩.

The shadows adapt to the rounded corners of the `tcolorbox`. An shrinked shadow will switch to sharp corners if necessary, an enlarged shadow may become more rounded depending on several factors. But `/tcb/sharp corners` → P. 47 have sharp shadows.

> **!** Shadows are not considered for the bounding box computation by default. Large shadows may be overlaped by the following content. But, the bounding box can be adapted if necessary.

```
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My own shadow,
  shadow={2mm}{-1mm}{0mm}{black!50!white}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  shadow={-1mm}{-2mm}{0mm}{fill=blue,
    opacity=0.5}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip
\begin{tcolorbox}[title=Double shadow,
  shadow={-1.5mm}{-1.5mm}{0mm}{fill=blue,
    opacity=0.25},
  shadow={1.5mm}{-1.5mm}{0mm}{fill=red,
    opacity=0.25}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip
\begin{tcolorbox}[title=Far shadow,
  shadow={5.5mm}{-3.5mm}{2mm}{fill=black,
    opacity=0.25}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip\bigskip
\begin{tcolorbox}[title=Halo shadow,
  shadow={0mm}{0mm}{-1.5mm}%
    {fill=yellow!75!red,opacity=0.5}]
This is a tcolorbox.
\end{tcolorbox}
```



`/tcb/fuzzy shadow={⟨xshift⟩}{⟨yshift⟩}{⟨offset⟩}{⟨step⟩}{⟨options⟩}` (no default)

Adds a new fuzzy shadow to the stack of shadows. Actually, this option adds several shadows which appear like a shadow with a fuzzy border. This fuzzy shadow follows the outline of the `tcolorbox` but is shifted by ⟨*xshift*⟩ and ⟨*yshift*⟩. The ⟨*offset*⟩ value is a distance value from the frame outline. A positive ⟨*offset*⟩ value shrinks the shadow and a negative ⟨*offset*⟩ value enlarges the shadow. The `{`⟨*step*⟩`}` value describes a shrink offset used for the combination of the partial shadows. The shadow is filled along a Ti*k*Z path with the given Ti*k*Z ⟨*options*⟩ but any `opacity` value will be ignored.

```
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My own shadow,
  fuzzy shadow={2mm}{-1mm}{0mm}{0.1mm}%
               {black!50!white}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  fuzzy shadow={-1mm}{-2mm}{0mm}{0.2mm}%
               {fill=blue}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip
\begin{tcolorbox}[title=Double shadow,
  fuzzy shadow={-1.5mm}{-1.5mm}{0mm}{0.1mm}%
               {blue},
  fuzzy shadow={1.5mm}{-1.5mm}{0mm}{0.1mm}%
               {red}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip
\begin{tcolorbox}[title=Far shadow,
  fuzzy shadow={5.5mm}{-3.5mm}{0mm}{0.3mm}%
               {black}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip\bigskip
\begin{tcolorbox}[title=Glow shadow,
  fuzzy shadow={0mm}{0mm}{-1.5mm}{0.15mm}%
               {yellow!75!red}]
This is a tcolorbox.
\end{tcolorbox}
```

My own shadow

This is a tcolorbox.

Another shadow

This is a tcolorbox.

Double shadow

This is a tcolorbox.

Far shadow

This is a tcolorbox.

Glow shadow

This is a tcolorbox.

```
\newtcolorbox{mybox}[1][]{enhanced,
  fuzzy shadow={1.0mm}{-1.0mm}{0.12mm}{0mm}{blue!50!white},
  fuzzy shadow={-1.0mm}{-1.0mm}{0.12mm}{0mm}{red!50!white},
  fuzzy shadow={-1.0mm}{1.0mm}{0.12mm}{0mm}{green!50!white},
  fuzzy shadow={1.0mm}{1.0mm}{0.12mm}{0mm}{yellow!50!white},#1
}

\begin{mybox}[title=A multi shadow box]
This is a tcolorbox.
\end{mybox}
```

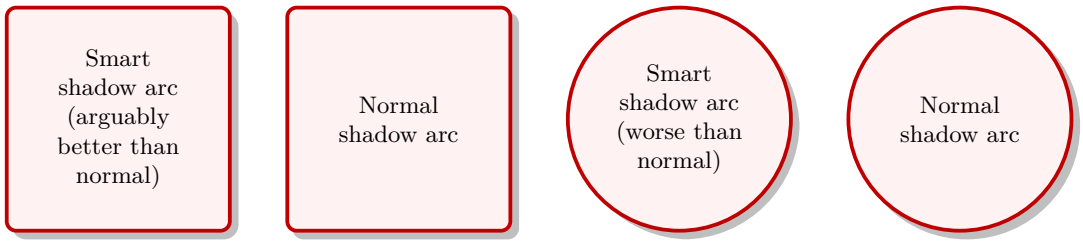
A multi shadow box

This is a tcolorbox.

N 2015-05-05 **/tcb/smart shadow arc=true|false** (default `true`, initially `true`)

If set to `true`, the shadow drawing algorithm tries to do a somewhat smart calculation of the arc for the shadow. The result is pleasing for typical boxes with rounded corners, but gives strange results for circular boxes.

```
\tcbset{enhanced,nobeforeafter,colback=red!5!white,
  colframe=red!75!black,width=3cm,square,halign=center,valign=center
  }

\begin{tcolorbox}[drop shadow]
Smart shadow arc (arguably better than normal)
\end{tcolorbox}
\hfill
\begin{tcolorbox}[smart shadow arc=false, drop shadow]
Normal shadow arc
\end{tcolorbox}
\hfill
\begin{tcolorbox}[circular arc, drop shadow]
Smart shadow arc (worse than normal)
\end{tcolorbox}
\hfill
\begin{tcolorbox}[circular arc, smart shadow arc=false, drop shadow]
Normal shadow arc
\end{tcolorbox}
```



/tcb/lifted shadow=`{`⟨*xshift*⟩`}{`⟨*yshift*⟩`}{`⟨*bend*⟩`}{`⟨*step*⟩`}{`⟨*options*⟩`}` (no default)

Adds a new lifted shadow to the stack of shadows. Actually, this option adds several shadows which appear like a shadow with a fuzzy border. This lifted shadow follows the outline of the `tcolorbox` but is shifted by ⟨*xshift*⟩ and ⟨*yshift*⟩ on the lower left corner and by −⟨*xshift*⟩ and ⟨*yshift*⟩ on the lower right corner. Additionally, there is a ⟨*bend*⟩ in the middle. The `{`⟨*step*⟩`}` value describes a shrink offset used for the combination of the partial shadows. The shadow is filled along a Ti*k*Z path with the given Ti*k*Z ⟨*options*⟩ but any `opacity` value will be ignored.

```
\tcbset{enhanced,colback=red!5!white,
  boxrule=0.1pt,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My own shadow,
  lifted shadow={1mm}{-2mm}{3mm}{0.1mm}%
                {black!50!white}]
This is a tcolorbox.
\end{tcolorbox}
```

10.6.4 Ti*k*Z Shadows

Alternativ to the package shadow options described before, shadows from the «Shadows Library» of Ti*k*Z can be used. Such shadows can be added directly to the frame path using `/tcb/frame style` → P. 139.

```
% \usetikzlibrary{shadows}
\begin{tcolorbox}[enhanced,
  colback=red!5!white,colframe=red!75!black,
  frame style={drop shadow} ]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

```
% \usetikzlibrary{shadows}
\begin{tcolorbox}[enhanced,height=3cm,
  colback=red!5!white,colframe=red!75!black,
  halign=center,valign=center,
  frame style={circular drop shadow} ]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

```
% \usetikzlibrary{shadows}
\begin{tcolorbox}[enhanced,width=2.5cm,
  square,circular arc,
  halign=center,valign=center,
  colback=red!5!white,colframe=red!75!black,
  frame style={circular glow={fill=red}} ]
  tcolorbox
\end{tcolorbox}
```

tcolorbox

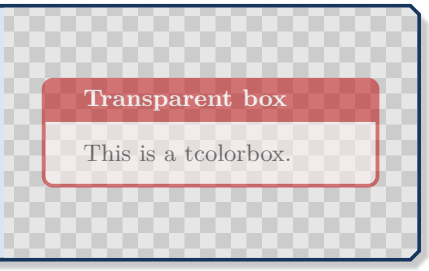
10.7 TikZ Picture Option Keys

The following general options are applicable for skins which use `tikzpicture` as `/tcb/graphical environment` → P. 125. Therefore, the skin `standard` → P. 193 does not support these options, but most other skins, e. g. `enhanced` → P. 195.

`/tcb/tikz=`⟨*tikz option list*⟩ (no default, initially empty)

Adds the given ⟨*tikz option list*⟩ to the main `tikzpicture` environment used to draw the color box, see [20]. If this option is applied a second time, the new ⟨*tikz option list*⟩ is appended to the current option list.

```
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=Transparent box,
  tikz={opacity=0.5,transparency group}]
This is a tcolorbox.
\end{tcolorbox}
```




```
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries,
  fontupper=\bfseries\Huge,
  halign title=center,halign=center}

\begin{tcolorbox}[title=Rotated box,
  tikz={rotate=30}]
Sold!
\end{tcolorbox}
```

`/tcb/tikz reset` (initially set)

Removes all options given by `/tcb/tikz`.

`/tcb/at begin tikz=`⟨*tikz code*⟩ (no default, initially empty)

The given ⟨*tikz code*⟩ is executed at the beginning of the `tikzpicture` environment after the TikZ option `execute at begin picture` was applied. If this option is applied a second time, the new ⟨*tikz code*⟩ is appended to the current code.

`/tcb/at begin tikz reset` (initially set)

Removes all code given by `/tcb/at begin tikz`.

`/tcb/at end tikz=`⟨*tikz code*⟩ (no default, initially empty)

The given ⟨*tikz code*⟩ is executed at the ending of the `tikzpicture` environment before the TikZ option `execute at end picture` was applied. If this option is applied a second time, the new ⟨*tikz code*⟩ is appended to the current code.

`/tcb/at end tikz reset` (initially set)

Removes all code given by `/tcb/at end tikz`.

/tcb/rotate=⟨*angle*⟩ (no default, initially unset)

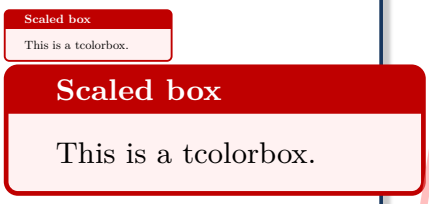
Rotates the tcolorbox by the given ⟨*angle*⟩. Note that this is a Ti*k*Z coordinate transformation i.e. not all graphical elements like shadings will really be rotated.

```
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=Rotated box,rotate=30]
This is a tcolorbox.
\end{tcolorbox}
```

/tcb/scale=⟨*fraction*⟩ (no default, initially unset)

Scales the tcolorbox by the given ⟨*fraction*⟩. Note that this is a Ti*k*Z coordinate transformation i.e. not all graphical elements like line widths will really be scaled.

```
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=Scaled box,scale=0.5]
This is a tcolorbox.
\end{tcolorbox}
\begin{tcolorbox}[title=Scaled box,scale=1.25]
This is a tcolorbox.
\end{tcolorbox}
```



/tcb/remember (style, initially unset)

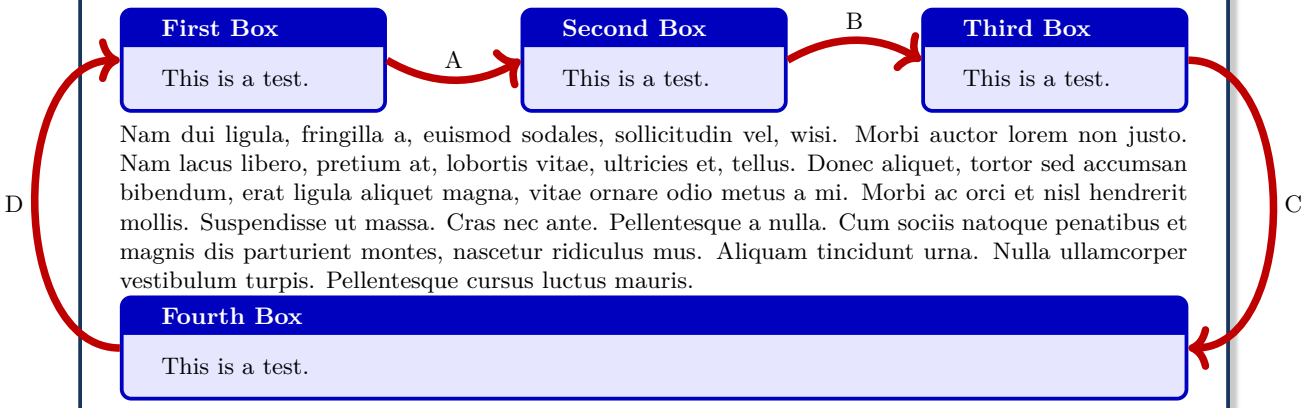
Shortcut for tikz={remember picture}. This allows one to reference nodes in other Ti*k*Z pictures.

```
\begin{tcolorbox}[enhanced,remember,colback=red!5!white,colframe=red!75!black,
  fonttitle=\bfseries,title=The four corners of a paper,
  overlay={\draw[red!50!white,line width=1mm,opacity=0.5,shorten >=3mm]
    (frame.north west) edge[->] (current page.north west)
    (frame.north east) edge[->] (current page.north east)
    (frame.south west) edge[->] (current page.south west)
    (frame.south east) edge[->] (current page.south east);}]
This is a tcolorbox.
\end{tcolorbox}
```

`/tcb/remember as=`⟨*name*⟩ (style, no default, initially unset)

The `frame` node will be remembered by the given ⟨*name*⟩ to be referenced in other Ti*k*Z pictures.

```
% \usepackage{lipsum}
\newtcolorbox{mybox}[1][]{enhanced,colframe=blue!75!black,colback=blue!10!white,
  fonttitle=\bfseries,#1}

\begin{mybox}[title=First Box,nobeforeafter,width=\linewidth/4,remember as=one]
This is a test.
\end{mybox}
\hfill
\begin{mybox}[title=Second Box,nobeforeafter,width=\linewidth/4,remember as=two]
This is a test.
\end{mybox}
\hfill
\begin{mybox}[title=Third Box,nobeforeafter,width=\linewidth/4,remember as=three]
This is a test.
\end{mybox}

\lipsum[2]

\begin{mybox}[title=Fourth Box,remember as=four]
This is a test.
\end{mybox}

\begin{tikzpicture}[overlay,remember picture,line width=1mm,draw=red!75!black]
  \draw[->] (one.east) to[bend right] node[above] {A} (two.west);
  \draw[->] (two.east) to[bend left] node[above] {B} (three.west);
  \draw[->] (three.east) to[bend left=90] node[right] {C} (four.east);
  \draw[->] (four.west) to[bend left=90] node[left] {D} (one.west);
\end{tikzpicture}
```



10.8 Underlay Option Keys

Underlays are quite similar to overlays described in Section 4.12 on page 69. Underlays are drawn *after* the frame and interior are drawn and *before* overlays and the text content is drawn; see Section 9.4 on page 133 for the general drawing scheme.

The differences between underlays and overlays are:

- Underlays are not applicable for the skins `standard` → P. 193 and `standard jigsaw` → P. 194, whereas overlays are applicable also for these skins. The skin `spartan` → P. 236 supports underlays but no overlays.

  > ! If an underlay is used with the `standard` → P. 193 skin, it is silently ignored.

- Underlays are stackable, i. e. several different underlays can be used on the same `tcolorbox`. Overlays are not stackable by default (but with some help of the library LIB hooks).
- Boxed titles are implemented with underlays (Section 10.2 on page 146), watermarks are implemented with overlays (Section 10.3 on page 156).

`/tcb/underlay`=⟨*graphical code*⟩ (no default, initially unset)

Adds ⟨*graphical code*⟩ to the box drawing process. This ⟨*graphical code*⟩ is drawn *after* the frame and interior and *before* the text content.

```
\newtcolorbox{mybox}[1][]{enhanced,colback=red!5!white,
  colbacktitle=red!85!black!50!white,
  colframe=red!75!black,fonttitle=\bfseries,watermark color=yellow!50!white,
  underlay={\begin{tcbclipinterior}
    \draw[red!40!white,line width=1cm] (interior.south west)--(interior.north east);
    \end{tcbclipinterior}},
  attach boxed title to top center={yshift=-2mm},#1}

\begin{mybox}[title=My box,watermark text=My Watermark]
\lipsum[2]
\end{mybox}
```

My box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

My Watermark

`/tcb/no underlay` (style, no default, initially set)

Removes the underlay if set before.

`/tcb/underlay broken`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` →P. 351 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process. `/tcb/underlay` →P. 186 overwrites this key.

`/tcb/underlay unbroken`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` →P. 351 but *is not* broken actually or if the box is set to be `/tcb/unbreakable` →P. 352, then the ⟨*graphical code*⟩ is added to the box drawing process. `/tcb/underlay` →P. 186 overwrites this key.

`/tcb/no underlay unbroken` (style, no default, initially set)

Removes the unbroken underlay if set before.

`/tcb/underlay first`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` →P. 351 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *first* part of the break sequence. `/tcb/underlay` →P. 186 overwrites this key.

`/tcb/no underlay first` (style, no default, initially set)

Removes the first underlay if set before.

`/tcb/underlay middle`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` →P. 351 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *middle* parts (if any) of the break sequence. `/tcb/underlay` →P. 186 overwrites this key.

`/tcb/no underlay middle` (style, no default, initially set)

Removes the middle underlay if set before.

`/tcb/underlay last`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` →P. 351 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *last* part of the break sequence. `/tcb/underlay` →P. 186 overwrites this key.

`/tcb/no underlay last` (style, no default, initially set)

Removes the last underlay if set before.

`/tcb/underlay boxed title`=⟨*graphical code*⟩ (no default, initially unset)

If the box has a *boxed title*, see Section 10.2 on page 146, then the ⟨*graphical code*⟩ is added to the box drawing process *before* the boxed title is drawn.

`/tcb/no underlay boxed title` (style, no default, initially set)

Removes the boxed title underlay if set before.

`/tcb/underlay unbroken and first`=⟨*graphical code*⟩ (no default, initially unset)

This is an abbreviation for setting `/tcb/underlay unbroken` and `/tcb/underlay first` together. `/tcb/underlay` →P. 186 overwrites this key.

`/tcb/underlay middle and last`=⟨*graphical code*⟩ (no default, initially unset)

This is an abbreviation for setting `/tcb/underlay middle` and `/tcb/underlay last` together. `/tcb/underlay` →P. 186 overwrites this key.

`/tcb/underlay unbroken and last`=⟨*graphical code*⟩ (no default, initially unset)

This is an abbreviation for setting `/tcb/underlay unbroken` and `/tcb/underlay last` together. `/tcb/underlay` →P. 186 overwrites this key.

N 2014-09-19 `/tcb/underlay first and middle`=⟨*graphical code*⟩ (no default, initially unset)

This is an abbreviation for setting `/tcb/underlay first` and `/tcb/underlay middle` together. `/tcb/underlay` →P. 186 overwrites this key.

10.9 Finish Option Keys

Finishes are quite similar to underlays described in Section 10.8 on page 186 and overlays described in Section 4.12 on page 69. Finishes are drawn *after* the text content is drawn; see Section 9.4 on page 133 for the general drawing scheme. Therefore, a finish will reduce the readability of the text content.

Finishes are intended for special effects like highlights or glosses or text over text.

- Finishes are only applicable for the skins `enhanced`→ P. 195, `empty`→ P. 226, `freelance`→ P. 239, `bicolor`→ P. 208, `beamer`→ P. 217, and `widget`→ P. 222.

  ! If a finish is used with the `standard`→ P. 193 skin, it is silently ignored.

- Finishes are stackable, i. e. several different finishes can be used on the same `tcolorbox`.

`/tcb/finish`=⟨*graphical code*⟩ (no default, initially unset)

Adds ⟨*graphical code*⟩ to the box drawing process. This ⟨*graphical code*⟩ is drawn *after* the text content.

```
\newtcolorbox{mybox}[1][]{enhanced,colback=red!5!white,
  colbacktitle=red!85!black!50!white,colframe=red!75!black,fonttitle=\bfseries,
  finish={\begin{tcbclipframe}
    \path[bottom color=black,top color=black!50!white,opacity=0.1]
      (frame.south west) -- (frame.south east) -- (frame.north east) -- cycle;
    \path[top color=white,bottom color=black!50!white,opacity=0.1]
      (frame.south west) -- (frame.north east) -- (frame.north west) -- cycle;
    \end{tcbclipframe}},#1}

\begin{mybox}[title=My box]
\lipsum[2]
\end{mybox}
```

My box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
\newtcolorbox{mybox}[1][]{enhanced,colback=red!5!white,
  colbacktitle=red!85!black!50!white,colframe=red!75!black,fonttitle=\bfseries,
  finish={\node[draw,fill=white,fill opacity=0.85,inner sep=5mm,
    rounded corners] at (frame.center) {\Huge\bfseries Finish!};},#1}

\begin{mybox}[title=My box]
\lipsum[2]
\end{mybox}
```

My box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Finish!

`/tcb/no finish` (style, no default, initially set)

Removes the finish if set before.

`/tcb/finish broken`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` → P. 351 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process. `/tcb/finish` → P. 188 overwrites this key.

`/tcb/finish unbroken`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` → P. 351 but *is not* broken actually or if the box is set to be `/tcb/unbreakable` → P. 352, then the ⟨*graphical code*⟩ is added to the box drawing process. `/tcb/finish` → P. 188 overwrites this key.

`/tcb/no finish unbroken` (style, no default, initially set)

Removes the unbroken finish if set before.

`/tcb/finish first`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` → P. 351 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *first* part of the break sequence. `/tcb/finish` → P. 188 overwrites this key.

`/tcb/no finish first` (style, no default, initially set)

Removes the first finish if set before.

`/tcb/finish middle`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` → P. 351 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *middle* parts (if any) of the break sequence. `/tcb/finish` → P. 188 overwrites this key.

`/tcb/no finish middle` (style, no default, initially set)

Removes the middle finish if set before.

`/tcb/finish last`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` → P. 351 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *last* part of the break sequence. `/tcb/finish` → P. 188 overwrites this key.

`/tcb/no finish last` (style, no default, initially set)

Removes the last finish if set before.

`/tcb/finish unbroken and first`=⟨*graphical code*⟩ (no default, initially unset)

This is an abbreviation for setting `/tcb/finish unbroken` and `/tcb/finish first` together. `/tcb/finish` → P. 188 overwrites this key.

`/tcb/finish middle and last`=⟨*graphical code*⟩ (no default, initially unset)

This is an abbreviation for setting `/tcb/finish middle` and `/tcb/finish last` together. `/tcb/finish` → P. 188 overwrites this key.

`/tcb/finish unbroken and last`=⟨*graphical code*⟩ (no default, initially unset)

This is an abbreviation for setting `/tcb/finish unbroken` and `/tcb/finish last` together. `/tcb/finish` → P. 188 overwrites this key.

N 2014-09-19 `/tcb/finish first and middle`=⟨*graphical code*⟩ (no default, initially unset)

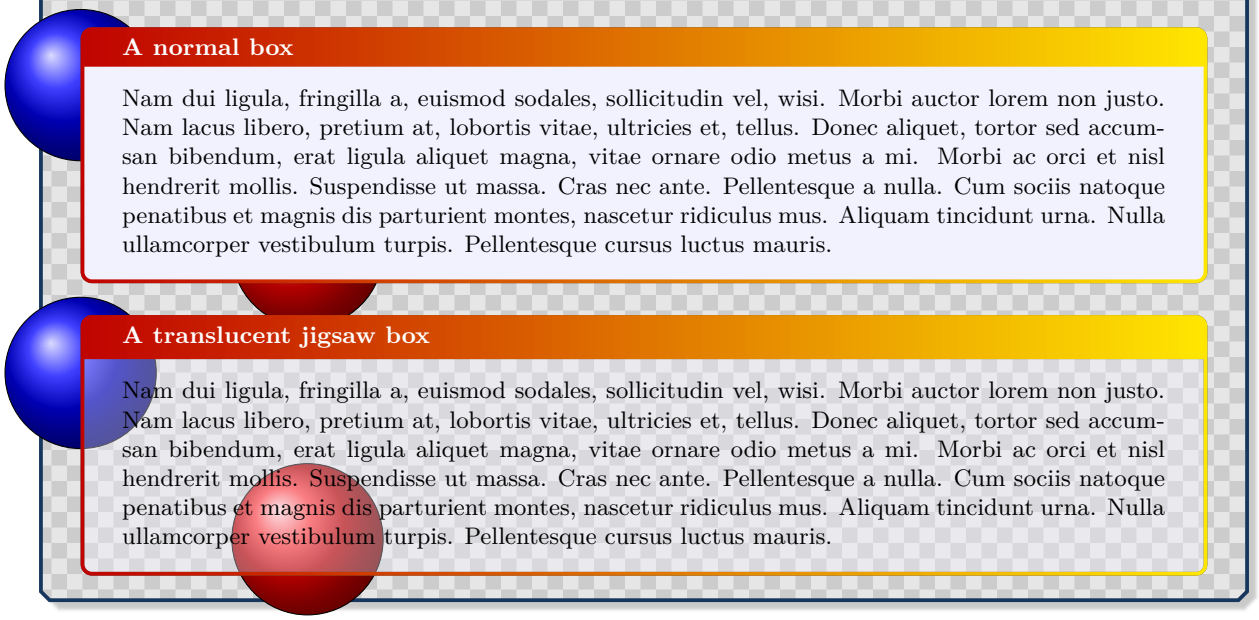
This is an abbreviation for setting `/tcb/finish first` and `/tcb/finish middle` together. `/tcb/finish` → P. 188 overwrites this key.

10.10 Jigsaw Skin Variants

As described in Section 9.1 on page 125, a `tcolorbox` is drawn by up to four *engines*. Typically, the *frame* engine fills the complete box area with color and the other engines fill certain areas with other colors. Finally, only the area which you see as *frame* of the box will display the frame color. For most applications, this is a good approach.

For certain boxes, a more delicate procedure is needed. E.g., if the box should be translucent, an already painted area cannot be made unpainted. Therefore, more elaborate frame engines saw holes into the frame where the interior area and optionally the title area will be painted. The resulting skins are called *jigsaw* skins. For `standard` → P. 193 and `enhanced` → P. 195, there are variants called `standard jigsaw` → P. 194 and `enhanced jigsaw` → P. 202.

```
\newcommand{\ballexample}{\begin{tikzpicture}
  \path[use as bounding box] (0,0.8) rectangle +(0.1,0.1);
  \shadedraw [shading=ball] (0,0) circle (1cm);
  \shadedraw [ball color=red] (3,-2.2) circle (1cm);
\end{tikzpicture}}

\tcbset{enhanced,colback=blue!5!white,
  frame style={left color=red!75!black,right color=red!10!yellow},
  fonttitle=\bfseries }

\ballexample

\begin{tcolorbox}[title=A normal box]
  \lipsum[2]
\end{tcolorbox}

\ballexample

\begin{tcolorbox}[title=A translucent jigsaw box,
  enhanced jigsaw,opacityback=0.35]
  \lipsum[2]
\end{tcolorbox}
```



```
\tcbset{enhanced,colback=red!10!white,coltitle=black,
  frame style={left color=red!75!black,right color=red!10!yellow},
  fonttitle=\bfseries,interior hidden,title hidden}

\begin{tcolorbox}[title=A normal box with hidden interior and title]
  This is a tcolorbox.
\end{tcolorbox}

\begin{tcolorbox}[enhanced jigsaw,
  title=A jigsaw box with hidden interior and title]
  This is a tcolorbox.
\end{tcolorbox}
```

A normal box with hidden interior and title

This is a tcolorbox.

A jigsaw box with hidden interior and title

This is a tcolorbox.

```
\newtcolorbox{mybox}{skin=enhancedmiddle jigsaw,leftrule=5mm,rightrule=5mm,
  boxsep=0mm,top=0mm,bottom=0mm,
  frame style={top color=blue,bottom color=red},interior hidden}

\begin{mybox}
  \lipsum[2]
\end{mybox}
```

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

10.11 Draft Mode

To reduce the compiliation time while drafting a document, the *draft mode* can be applied. Basically, it changes all skins to spartan → P. 236 and sets the /tcb/fit algorithm → P. 388 to squeeze. Especially, when fuzzy shadows are used, the speedup will be considerable high.

> ! It is strongly recommended that the draft mode is *not* used for the final document. Use spartan → P. 236 directly, if you want to stay with it. The draft mode implementation may change in future.

> ! Normally, switching to the draft mode should not alter the geometry of your document. Since overlays are deactivated, any code placed there (e.g. counter changes) is not executed anymore! Also, /tcb/remember as → P. 185 will not have any effect. You may exclude critical code with \tcbinterruptdraftmode / \tcbcontinuedraftmode from converting to draft mode.

\tcbstartdraftmode

Any following tcolorbox code is put into *draft mode*. All skin settings are overruled with spartan → P. 236. Overlays, watermarks, shadows, borderlines, and rounded corners are deactivated for all tcolorbox layers.

\tcbstopdraftmode

The *draft mode* is deactivated for the following code.

\tcbinterruptdraftmode

If the compilation is in *draft mode*, the *draft mode* is deactivated until a following \tcbcontinuedraftmode is detected.

If the compilation is not in *draft mode*, nothing happens and a following \tcbcontinuedraftmode will not start the *draft mode*.

> ! The pair \tcbinterruptdraftmode and \tcbcontinuedraftmode cannot be used nested.

\tcbcontinuedraftmode

Continues the *draft mode* which was suspended by a preceding \tcbinterruptdraftmode. Nothing happens, if there was no draft mode before \tcbinterruptdraftmode.

> ! Code, which is place between \tcbinterruptdraftmode and \tcbcontinuedraftmode is shielded from *draft mode*.

/tcb/draftmode=true|false (default true, initially false)

If set to true, the *draft mode* is started. If set to false, the *draft mode* is stopped.

```
\newtcolorbox{mybeamer}[2][]{beamer,colback=Salmon!50!white,
  colframe=FireBrick!75!black,adjusted title={#2},#1}

\begin{mybeamer}{Beamer box}
This box looks like a box provided by the \texttt{beamer} class.
\end{mybeamer}\par\medskip
\begin{mybeamer}[draftmode]{Beamer box}
This box looks like a box provided by the \texttt{beamer} class.
\end{mybeamer}
```

Beamer box

This box looks like a box provided by the beamer class.

Beamer box

This box looks like a box provided by the beamer class.

10.12 Skin Family 'standard'

Note that the option keys /tcb/frame style → P. 139, /tcb/interior style → P. 140, /tcb/segmentation style → P. 142, and /tcb/title style → P. 142 are not be applicable to the standard skin. Also, watermarks (see Subsection 10.3) are not usable with the standard skin.

/tcb/skin=standard (skin)

This is the standard skin from the core package. All drawing engines are set to type `standard`. The drawing is based on `pgf` commands and does not need the `tikz` package.

Environment and engines for the skin 'standard'

/tcb/graphical environment → P. 125:	pgfpicture
/tcb/frame engine → P. 126:	standard
/tcb/interior titled engine → P. 126:	standard
/tcb/interior engine → P. 127:	standard
/tcb/segmentation engine → P. 127:	standard
/tcb/title engine → P. 127:	standard

/tcb/standard (style, no value)

This is an abbreviation for setting `skin=standard`.

```
\begin{tcbraster}[standard,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacktitle=LimeGreen!75!DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=standard jigsaw (skin)

This is the standard jigsaw skin from the core package. It differs from the skin standard → P. 193 by its frame engine, see Section 10.10 on page 190.

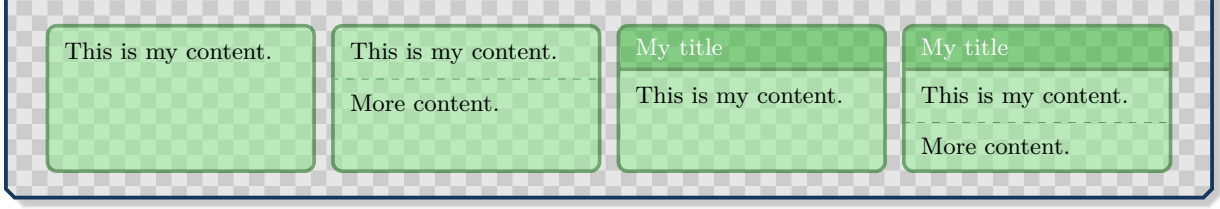
Environment and engines for the skin 'standard jigsaw'

/tcb/graphical environment → P. 125:	pgfpicture
/tcb/frame engine → P. 126:	standardjigsaw
/tcb/interior titled engine → P. 126:	standard
/tcb/interior engine → P. 127:	standard
/tcb/segmentation engine → P. 127:	standard
/tcb/title engine → P. 127:	standard

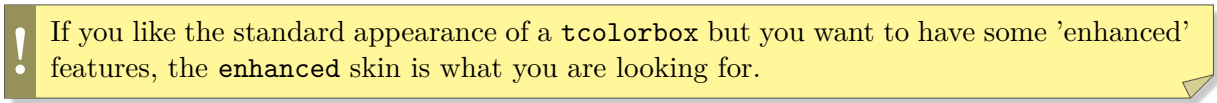
/tcb/standard jigsaw (style, no value)

This is an abbreviation for setting skin=standard jigsaw.

```
\begin{tcbraster}[standard jigsaw,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacktitle=LimeGreen!75!DarkGreen,
    opacityframe=0.5,opacityback=0.5,opacitybacktitle=0.5,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```



10.13 Skin Family 'enhanced'


If you like the standard appearance of a `tcolorbox` but you want to have some 'enhanced' features, the `enhanced` skin is what you are looking for.

`/tcb/skin=enhanced` (skin)

This skin translates the drawing commands of the core package into `tikz` path commands. Therefore, it allows all `tikz` high level options for these paths and has more flexibility compared to the `standard` → P. 193 skin. You pay for this with some prolonged compilation time. The `tikz` path options can be given with the option keys `/tcb/frame style` → P. 139, `/tcb/interior style` → P. 140, `/tcb/segmentation style` → P. 142, and `/tcb/title style` → P. 142.

Environment and engines for the skin 'enhanced'

`/tcb/graphical environment` → P. 125:	`tikzpicture`
`/tcb/frame engine` → P. 126:	`path`
`/tcb/interior titled engine` → P. 126:	`path`
`/tcb/interior engine` → P. 127:	`path`
`/tcb/segmentation engine` → P. 127:	`path`
`/tcb/title engine` → P. 127:	`path`

`/tcb/enhanced` (style, no value)

This is an abbreviation for setting `skin=enhanced`.

```
\begin{tcbraster}[enhanced,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacktitle=LimeGreen!75!DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

```
% \usetikzlibrary{shadings}          % preamble
\tcbset{skin=enhanced,fonttitle=\bfseries,
  frame style={upper left=blue,upper right=red,lower left=yellow,lower right=green},
  interior style={white,opacity=0.5},
  segmentation style={black,solid,opacity=0.2,line width=1pt}}

\begin{tcolorbox}[title=Nice box in rainbow colors]
  With the 'enhanced' skin, it is quite easy to produce fancy looking effects.
  \tcblower
  Note that this is still a \texttt{tcolorbox}.
\end{tcolorbox}
```

Nice box in rainbow colors

With the 'enhanced' skin, it is quite easy to produce fancy looking effects.

Note that this is still a `tcolorbox`.

```
% \usetikzlibrary{decorations.pathmorphing} % preamble
\tcbset{skin=enhanced,fonttitle=\bfseries,boxrule=1mm,
  frame style={draw=FireBrick,fill=Salmon},drop fuzzy shadow,
  interior style={draw=FireBrick,top color=Salmon!10,bottom color=Salmon!20},
  segmentation style={draw=FireBrick,solid,decorate,
        decoration={coil,aspect=0,segment length=10.1mm}}}

\begin{tcblisting}{title=A listing box with shadow and some specials}
Of course, skins can be used for listings also.
\begin{equation}
  \int\limits_1^2 \frac{1}{x}~dx = \ln(2).
\end{equation}
\end{tcblisting}
```

A listing box with shadow and some specials

```
Of course, skins can be used for listings also.
\begin{equation}
  \int\limits_1^2 \frac{1}{x}~dx = \ln(2).
\end{equation}
```

Of course, skins can be used for listings also.

$$\int_1^2 \frac{1}{x}\,dx = \ln(2). \tag{2}$$

`/tcb/enhanced standard` (style, no value)

For unbreakable boxes, this is identical to using `/tcb/enhanced` → P. 195. But, for breakable boxes, the *break sequence* is identical to the `standard` → P. 193 skin, see Section 17.7 from page 362.

`/tcb/blank` (style, initially unset)

This style relies on the skin `enhanced` → P. 195. All drawing operations are hidden and all margins are set to `0pt`. See `/tcb/blanker` → P. 227 for switching off the drawing engines.

```
\begin{tcolorbox}[blank,watermark text=A blank box]
\lipsum[1]
\end{tcolorbox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

A blank box

\tcbline

Sometimes, a line is only a line. With \tcblower → P. 12 you separate the box content into two functional units. \tcbline draws only a line which looks like the segmentation line between upper and lower part. Furthermore, you can use \tcbline more than just once. \tcbline always uses the path drawing engine. Therefore, the /tcb/segmentation style → P. 142 can be applied.

```
\tcbset{enhanced,colframe=blue!50!black,colback=white}

\begin{tcolorbox}[colupper=red!50!black,collower=green!50!black]
\lipsum[1]
\tcbline
\lipsum[2]
\tcblower
\lipsum[3]
\tcbline
\lipsum[4]
\end{tcolorbox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

\tcbline*

Equivalent to \tcbline, but in a breakable box, \tcbline* is removed if at a page/box break. Also, it is removed at the end of a box.

/tcb/skin=enhancedfirst (skin)

This is a flavor of enhanced → P. 195 which is used as a *first* part in a break sequence for enhanced → P. 195. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'enhancedfirst'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	pathfirst
/tcb/interior titled engine → P. 126:	pathfirst
/tcb/interior engine → P. 127:	pathfirst
/tcb/segmentation engine → P. 127:	path
/tcb/title engine → P. 127:	pathfirst

```
\begin{tcbraster}[skin=enhancedfirst,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacktitle=LimeGreen!75!DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=enhancedmiddle (skin)

This is a flavor of enhanced → P. 195 which is used as a *middle* part in a break sequence for enhanced → P. 195. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'enhancedmiddle'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	pathmiddle
/tcb/interior titled engine → P. 126:	pathmiddle
/tcb/interior engine → P. 127:	pathmiddle
/tcb/segmentation engine → P. 127:	path
/tcb/title engine → P. 127:	pathmiddle

```
\begin{tcbraster}[skin=enhancedmiddle,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacktitle=LimeGreen!75!DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=enhancedlast (skin)

This is a flavor of enhanced → P. 195 which is used as a *last* part in a break sequence for enhanced → P. 195. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'enhancedlast'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	pathlast
/tcb/interior titled engine → P. 126:	pathlast
/tcb/interior engine → P. 127:	pathlast
/tcb/segmentation engine → P. 127:	path
/tcb/title engine → P. 127:	pathlast

```
\begin{tcbraster}[skin=enhancedlast,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacktitle=LimeGreen!75!DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=enhanced jigsaw (skin)

This is the jigsaw variant of skin enhanced → P. 195. It differs by its frame engine, see Section 10.10 on page 190.

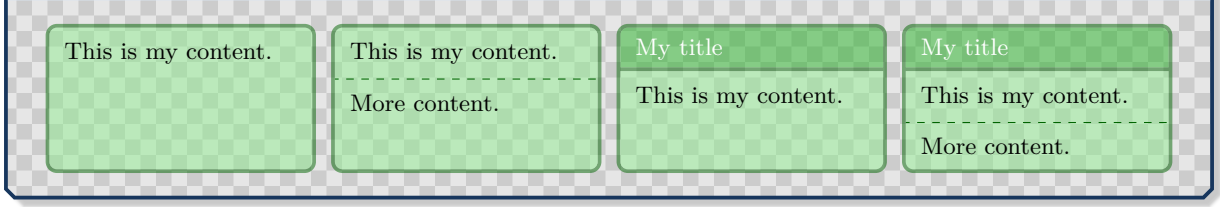
Environment and engines for the skin 'enhanced jigsaw'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	pathjigsaw
/tcb/interior titled engine → P. 126:	path
/tcb/interior engine → P. 127:	path
/tcb/segmentation engine → P. 127:	path
/tcb/title engine → P. 127:	path

/tcb/enhanced jigsaw (style, no value)

This is an abbreviation for setting skin=enhanced jigsaw.

```
\begin{tcbraster}[enhanced jigsaw,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacktitle=LimeGreen!75!DarkGreen,
    opacityframe=0.5,opacityback=0.5,opacitybacktitle=0.5,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```



/tcb/skin=enhancedfirst jigsaw (skin)

This is the jigsaw variant of skin enhancedfirst → P. 199. It differs by its frame engine, see Section 10.10 on page 190.

Environment and engines for the skin 'enhancedfirst jigsaw'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	pathfirstjigsaw
/tcb/interior titled engine → P. 126:	pathfirst
/tcb/interior engine → P. 127:	pathfirst
/tcb/segmentation engine → P. 127:	path
/tcb/title engine → P. 127:	pathfirst

```
\begin{tcbraster}[skin=enhancedfirst jigsaw,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacktitle=LimeGreen!75!DarkGreen,
    opacityframe=0.5,opacityback=0.5,opacitybacktitle=0.5,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=enhancedmiddle jigsaw (skin)

This is the jigsaw variant of skin enhancedmiddle → P. 200. It differs by its frame engine, see Section 10.10 on page 190.

Environment and engines for the skin 'enhancedmiddle jigsaw'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	pathmiddlejigsaw
/tcb/interior titled engine → P. 126:	pathmiddle
/tcb/interior engine → P. 127:	pathmiddle
/tcb/segmentation engine → P. 127:	path
/tcb/title engine → P. 127:	pathmiddle

```
\begin{tcbraster}[skin=enhancedmiddle jigsaw,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacktitle=LimeGreen!75!DarkGreen,
    opacityframe=0.5,opacityback=0.5,opacitybacktitle=0.5,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/marker (style, no value)

This styles relies on the skin enhancedmiddle jigsaw. It is intended to be used as an optical marker like a highlighter pen.

```
\begin{tcolorbox}[marker]
\lipsum[2]
\end{tcolorbox}
```

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

This examples demonstrates the creation of several *text marker* environments based on `enhancedmiddle` → P. 200.

```
\tcbset{textmarker/.style={%
    skin=enhancedmiddle jigsaw,breakable,parbox=false,
    boxrule=0mm,leftrule=5mm,rightrule=5mm,boxsep=0mm,arc=0mm,outer arc=0mm,
    left=3mm,right=3mm,top=1mm,bottom=1mm,toptitle=1mm,bottomtitle=1mm,oversize}}

\newtcolorbox{yellow}{textmarker,colback=yellow!5!white,colframe=yellow}
\newtcolorbox{orange}{textmarker,colback=DarkOrange!5!white,
                      colframe=DarkOrange!75!yellow}
\newtcolorbox{red}{textmarker,colback=red!5!white,colframe=red}
\newtcolorbox{blue}{textmarker,colback=DeepSkyBlue!5!white,colframe=DeepSkyBlue}
\newtcolorbox{green}{textmarker,colback=Chartreuse!5!white,colframe=Chartreuse}
\newtcolorbox{rainbow}{textmarker,interior hidden,
  frame style={top color=blue,bottom color=red,middle color=green}}

\begin{yellow}
  \lipsum[1-3]
\end{yellow}

\begin{orange}
  \lipsum[4]
\end{orange}

\begin{red}
  \lipsum[5]
\end{red}

\begin{green}
  \lipsum[6]
\end{green}

\begin{blue}
  \lipsum[7]
\end{blue}

\begin{rainbow}
  \lipsum[8]
\end{rainbow}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique,

libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

`/tcb/skin=enhancedlast jigsaw` (skin)

This is the jigsaw variant of skin `enhancedlast` → P. 201. It differs by its frame engine, see Section 10.10 on page 190.

Environment and engines for the skin 'enhancedlast'

`/tcb/graphical environment` → P. 125:	`tikzpicture`
`/tcb/frame engine` → P. 126:	`pathlastjigsaw`
`/tcb/interior titled engine` → P. 126:	`pathlast`
`/tcb/interior engine` → P. 127:	`pathlast`
`/tcb/segmentation engine` → P. 127:	`path`
`/tcb/title engine` → P. 127:	`pathlast`

```
\begin{tcbraster}[skin=enhancedlast jigsaw,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacktitle=LimeGreen!75!DarkGreen,
    opacityframe=0.5,opacityback=0.5,opacitybacktitle=0.5,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

10.14 Skin Family 'bicolor'

/tcb/skin=bicolor (skin)

This skin is quite similar to the standard → P. 193 and enhanced → P. 195 skin. But instead of a segmentation line, the optional lower part of the box is filled with a different color or drawn with a different style.

Environment and engines for the skin 'bicolor'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	path
/tcb/interior titled engine → P. 126:	*special*
/tcb/interior engine → P. 127:	*special*
/tcb/segmentation engine → P. 127:	*special*
/tcb/title engine → P. 127:	path

- The most basic usage of this skin is to set the background color of the lower part by /tcb/colbacklower → P. 209 and all other options like for the standard → P. 193 skin.

```
\begin{tcolorbox}[skin=bicolor,title=The title,
    colframe=FireBrick!75!black,colback=Salmon!50!white,colbacklower=Salmon]
  The upper part.
  \tcblower
  The lower part.
\end{tcolorbox}
```

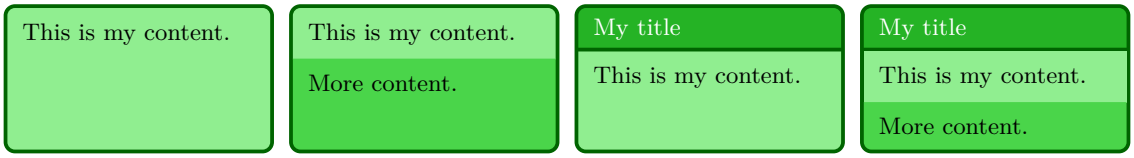
- The more advanced usage of this skin is to apply the /tcb/frame style → P. 139 and the /tcb/interior style → P. 140 like for the enhanced → P. 195 skin. Also, the /tcb/segmentation style → P. 142 can be used, but it is applied to the whole lower part.

```
\begin{tcolorbox}[skin=bicolor,title=The title,
    frame style={top color=FireBrick,
                 bottom color=FireBrick!15!white,draw=black},
    interior style={left color=Salmon,right color=Salmon!50!white},
    segmentation style={right color=Salmon,left color=Salmon!50!white}]
  The upper part.
  \tcblower
  The lower part.
\end{tcolorbox}
```

/tcb/bicolor (style, no value)

This is an abbreviation for setting skin=bicolor.

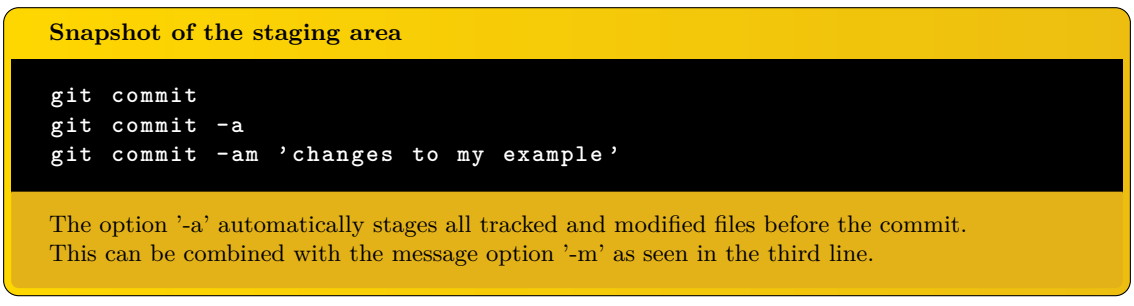
```
\begin{tcbraster}[bicolor,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
    colbacktitle=LimeGreen!75!DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```



/tcb/colbacklower=⟨*color*⟩ (no default, initially `black!15!white`)

Sets the background ⟨*color*⟩ of the lower part. It depends on the skin, if this value is used.

```
\tcbset{gitexample/.style={listing and comment,comment={#1},
  skin=bicolor,boxrule=1mm,fonttitle=\bfseries,coltitle=black,
  frame style={draw=black,left color=Gold,right color=Goldenrod!50!Gold},
  colback=black,colbacklower=Goldenrod!75!Gold,
  colupper=white,collower=black,
  listing options={language={bash},aboveskip=0pt,belowskip=0pt,nolol,
  basicstyle=\ttfamily\bfseries,extendedchars=true}}}

\begin{tcblisting}{title={Snapshot of the staging area},
  gitexample={The option '-a' automatically stages all tracked and modified
              files before the commit.\par
              This can be combined with the message option '-m'
              as seen in the third line.}}
git commit
git commit -a
git commit -am 'changes to my example'
\end{tcblisting}
```



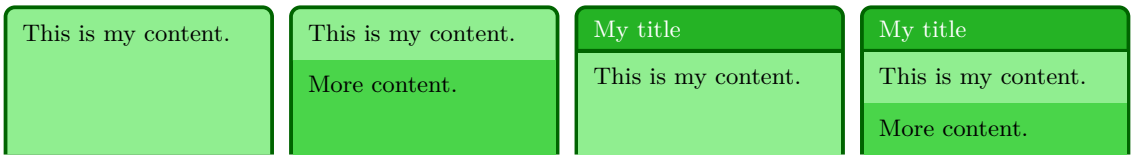
/tcb/skin=bicolorfirst (skin)

This is a flavor of bicolor → P. 208 which is used as a *first* part in a break sequence for bicolor → P. 208. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'bicolorfirst'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	pathfirst
/tcb/interior titled engine → P. 126:	*special*
/tcb/interior engine → P. 127:	*special*
/tcb/segmentation engine → P. 127:	*special*
/tcb/title engine → P. 127:	pathfirst

```
\begin{tcbraster}[skin=bicolorfirst,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
    colbacktitle=LimeGreen!75!DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```



/tcb/skin=bicolormiddle (skin)

This is a flavor of bicolor → P. 208 which is used as a *middle* part in a break sequence for bicolor → P. 208. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'bicolormiddle'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	pathmiddle
/tcb/interior titled engine → P. 126:	*special*
/tcb/interior engine → P. 127:	*special*
/tcb/segmentation engine → P. 127:	*special*
/tcb/title engine → P. 127:	pathmiddle

```
\begin{tcbraster}[skin=bicolormiddle,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
    colbacktitle=LimeGreen!75!DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

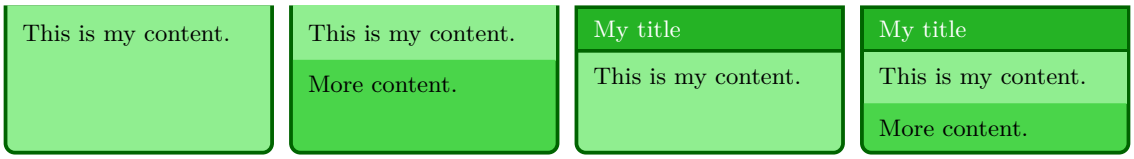
/tcb/skin=bicolorlast (skin)

This is a flavor of bicolor → P. 208 which is used as a *last* part in a break sequence for bicolor → P. 208. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'bicolorlast'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	pathlast
/tcb/interior titled engine → P. 126:	*special*
/tcb/interior engine → P. 127:	*special*
/tcb/segmentation engine → P. 127:	*special*
/tcb/title engine → P. 127:	pathlast

```
\begin{tcbraster}[skin=bicolorlast,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
    colbacktitle=LimeGreen!75!DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```



10.15 Skin Family 'tile'

N 2016-02-25 **/tcb/skin=tile** (skin)

This skin is a variant of skin bicolor → P. 208. Especially, the optional lower part of the box is colored by /tcb/colbacklower → P. 209. The main difference to bicolor → P. 208 is that tile has no frame.

Environment and engines for the skin 'tile'

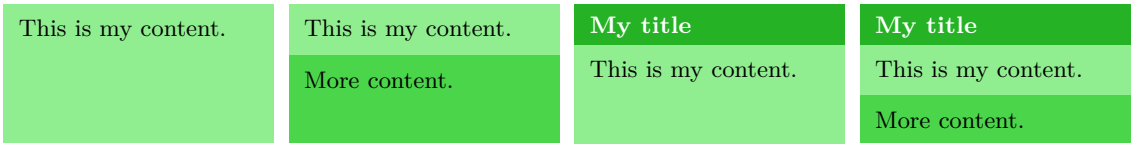
/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	empty
/tcb/interior titled engine → P. 126:	*special*
/tcb/interior engine → P. 127:	*special*
/tcb/segmentation engine → P. 127:	*special*
/tcb/title engine → P. 127:	path

N 2016-02-25 **/tcb/tile** (style, no value)

This is an abbreviation for setting skin=tile.

It also changes the geometry and some style options.

```
\begin{tcbraster}[tile,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
    colbacktitle=LimeGreen!75!DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```



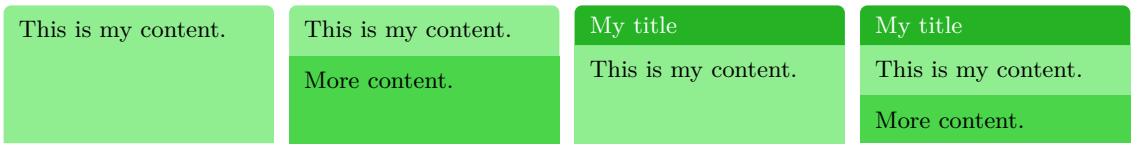
N 2016-02-25 **/tcb/skin=tilefirst** (skin)

This is a flavor of tile → P. 213 which is used as a *first* part in a break sequence for tile → P. 213. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'tilefirst'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	empty
/tcb/interior titled engine → P. 126:	*special*
/tcb/interior engine → P. 127:	*special*
/tcb/segmentation engine → P. 127:	*special*
/tcb/title engine → P. 127:	pathfirst

```
\begin{tcbraster}[skin=tilefirst,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
    colbacktitle=LimeGreen!75!DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm,boxrule=0pt]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```



N 2016-02-25 **/tcb/skin=tilemiddle** (skin)

This is a flavor of tile[→ P. 213] which is used as a *middle* part in a break sequence for tile[→ P. 213]. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'tilemiddle'

/tcb/graphical environment[→ P. 125]:	tikzpicture
/tcb/frame engine[→ P. 126]:	empty
/tcb/interior titled engine[→ P. 126]:	*special*
/tcb/interior engine[→ P. 127]:	*special*
/tcb/segmentation engine[→ P. 127]:	*special*
/tcb/title engine[→ P. 127]:	pathmiddle

```
\begin{tcbraster}[skin=tilemiddle,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
    colbacktitle=LimeGreen!75!DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm,boxrule=0pt]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

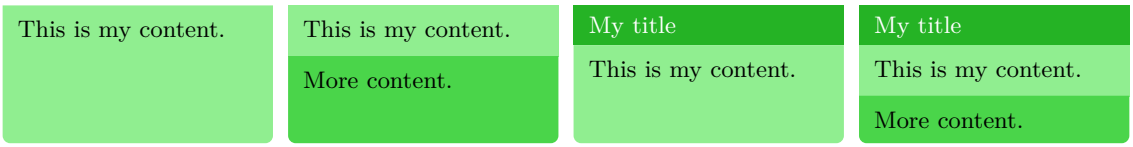
N 2016-02-25 **/tcb/skin=tilelast** (skin)

This is a flavor of tile → P. 213 which is used as a *last* part in a break sequence for tile → P. 213. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'tilelast'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	empty
/tcb/interior titled engine → P. 126:	*special*
/tcb/interior engine → P. 127:	*special*
/tcb/segmentation engine → P. 127:	*special*
/tcb/title engine → P. 127:	pathlast

```
\begin{tcbraster}[skin=tilelast,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
    colbacktitle=LimeGreen!75!DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm,boxrule=0pt]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```



10.16 Skin Family 'beamer'

/tcb/skin=beamer (skin)

This skin resembles boxes known from the `beamer` class and therefore is called 'beamer'. It uses the normal colors from the core package but shades them a little bit. To use this skin, the `tikz` library `shadings` has to be included in the preamble by:

```
\usetikzlibrary{shadings}
```

The appearance of the skin can be controlled by /tcb/frame style → P. 139 and /tcb/interior style → P. 140, if needed. Here, the *segmentation* cannot be controlled by a style.

Environment and engines for the skin 'beamer'

/tcb/graphical environment → P. 125:	`tikzpicture`
/tcb/frame engine → P. 126:	`path`
/tcb/interior titled engine → P. 126:	*special*
/tcb/interior engine → P. 127:	*special*
/tcb/segmentation engine → P. 127:	*special*
/tcb/title engine → P. 127:	`path`

/tcb/beamer (style, no value)

This is an abbreviation for setting `skin=beamer`.

! It also changes the geometry and some style options.

```
\begin{tcbraster}[beamer,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

```
\begin{tcolorbox}[beamer,colback=Salmon!50!white,colframe=FireBrick!75!black,
  adjusted title=A colored box with the 'beamer' skin]
This box looks like a box provided by the \texttt{beamer} class.
\end{tcolorbox}
```

A colored box with the 'beamer' skin

This box looks like a box provided by the `beamer` class.

```
\begin{tcolorbox}[beamer,colframe=blue,colback=black,
  watermark graphics=lichtspiel.jpg,
  coltext=white,watermark opacity=0.75,watermark stretch=1.0,
  title=Beamer Box with background picture]
\lipsum[1]
\end{tcolorbox}
```

Beamer Box with background picture

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

```
\newtcolorbox{myblock}[2][]{%
  beamer,breakable,colback=LightBlue,colframe=DarkBlue,#1,title=#2}%

\begin{myblock}{Beamerish \texttt{block}: \texttt{myblock}}
\lipsum[1]
\end{myblock}
```

Beamerish `block`: `myblock`

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

/tcb/skin=beamerfirst (skin)

This is a flavor of beamer → P. 217 which is used as a *first* part in a break sequence for beamer → P. 217. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'beamerfirst'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	pathfirst
/tcb/interior titled engine → P. 126:	*special*
/tcb/interior engine → P. 127:	*special*
/tcb/segmentation engine → P. 127:	*special*
/tcb/title engine → P. 127:	pathfirst

```
\begin{tcbraster}[beamer,skin=beamerfirst,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

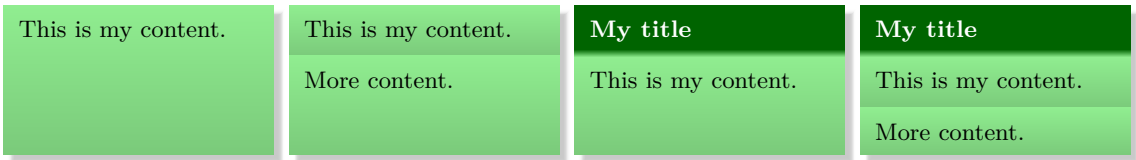
/tcb/skin=beamermiddle (skin)

This is a flavor of beamer → P. 217 which is used as a *middle* part in a break sequence for beamer → P. 217. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'beamermiddle'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	pathmiddle
/tcb/interior titled engine → P. 126:	*special*
/tcb/interior engine → P. 127:	*special*
/tcb/segmentation engine → P. 127:	*special*
/tcb/title engine → P. 127:	pathmiddle

```
\begin{tcbraster}[beamer,skin=beamermiddle,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```



/tcb/skin=beamerlast (skin)

This is a flavor of beamer → P. 217 which is used as a *last* part in a break sequence for beamer → P. 217. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'beamerlast'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	pathlast
/tcb/interior titled engine → P. 126:	*special*
/tcb/interior engine → P. 127:	*special*
/tcb/segmentation engine → P. 127:	*special*
/tcb/title engine → P. 127:	pathlast

```
\begin{tcbraster}[beamer,skin=beamerlast,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

10.17 Skin Family 'widget'

/tcb/skin=widget (skin)

This skin uses the normal colors from the core package but shades them a little bit. To use this skin, the `tikz` library `shadings` has to be included in the preamble by:

```
\usetikzlibrary{shadings}
```

The appearance of the skin can be controlled by /tcb/frame style → P. 139, /tcb/interior style → P. 140, and /tcb/segmentation style → P. 142, if needed.

Environment and engines for the skin 'widget'

/tcb/graphical environment → P. 125:	`tikzpicture`
/tcb/frame engine → P. 126:	`path`
/tcb/interior titled engine → P. 126:	`path`
/tcb/interior engine → P. 127:	`path`
/tcb/segmentation engine → P. 127:	*special*
/tcb/title engine → P. 127:	*special*

/tcb/widget (style, no value)

This is an abbreviation for setting `skin=widget`.

! It also changes the geometry and some style options.

```
\begin{tcbraster}[widget,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

```
\begin{tcolorbox}[widget,colback=Salmon!50!white,colframe=FireBrick!75!black,
  adjusted title=A colored box with the 'widget' skin]
This is my content.
\end{tcolorbox}
```

A colored box with the 'widget' skin

This is my content.

/tcb/skin=widgetfirst (skin)

This is a flavor of widget → P. 222 which is used as a *first* part in a break sequence for widget → P. 222. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'widgetfirst'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	pathfirst
/tcb/interior titled engine → P. 126:	pathfirst
/tcb/interior engine → P. 127:	pathfirst
/tcb/segmentation engine → P. 127:	*special*
/tcb/title engine → P. 127:	*special*

```
\begin{tcbraster}[widget,skin=widgetfirst,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=widgetmiddle (skin)

This is a flavor of widget → P. 222 which is used as a *middle* part in a break sequence for widget → P. 222. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'widgetmiddle'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	pathmiddle
/tcb/interior titled engine → P. 126:	pathmiddle
/tcb/interior engine → P. 127:	pathmiddle
/tcb/segmentation engine → P. 127:	*special*
/tcb/title engine → P. 127:	*special*

```
\begin{tcbraster}[widget,skin=widgetmiddle,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=widgetlast (skin)

This is a flavor of widget → P. 222 which is used as a *last* part in a break sequence for widget → P. 222. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'widgetlast'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	pathlast
/tcb/interior titled engine → P. 126:	pathlast
/tcb/interior engine → P. 127:	pathlast
/tcb/segmentation engine → P. 127:	*special*
/tcb/title engine → P. 127:	*special*

```
\begin{tcbraster}[widget,skin=widgetlast,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

10.18 Skin Family 'empty'

/tcb/skin=empty (skin)

This skin sets all engines to `empty`, i. e. nothing is drawn at all. Therefore, this skin is a good starting point to create a complete new style by yourself.

Environment and engines for the skin 'empty'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	empty
/tcb/interior titled engine → P. 126:	empty
/tcb/interior engine → P. 127:	empty
/tcb/segmentation engine → P. 127:	empty
/tcb/title engine → P. 127:	empty

/tcb/empty (style, no value)

This is an abbreviation for setting `skin=empty`.

```
\begin{tcbraster}[empty,raster equal height,raster columns=4,
    coltitle=Navy,borderline={2pt}{0pt}{black!10!white},
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/blanker (style, initially unset)

This style relies on the skin `empty` → P. 226. All engines are set to empty and all margins are set to `0pt`. In contrast to `/tcb/blank` → P. 197, the graphical paths are not constructed with exception of the geometry nodes.

```
\begin{tcolorbox}[blanker,watermark text=A blank box]
\lipsum[1]
\end{tcolorbox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

A blank box

```
% \tcbuselibrary{fitting}
\newtcboxfit{\mybox}[1]{blanker,width=4cm,height=7cm,top=4pt,
  watermark text=#1}

\begin{tabular}{|c|c|c|}\hline
A & B & C\\\hline
\mybox{A}{\lipsum[1]} & \mybox{B}{\lipsum[2]} & \mybox{C}{\lipsum[3]}\\\hline
\end{tabular}
```

A	B	C
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.	Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.	Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

A blank box

`/tcb/blankest` (style, initially unset)

This style extends `/tcb/blanker` → P. 227. All engines are set to empty and all margins are set to `0pt`. In contrast to `/tcb/blanker` → P. 227, also title, shadow, underlay, overlay, finish and borderline are removed.

```
% \tcbuselibrary{raster}
\begin{tcbraster}[raster columns=3,raster equal height,
    title=Box \thetcbrasternum,
    enhanced,size=small,colframe=red!50!black,colback=red!10!white,
    coltitle=yellow!85!black,
    drop fuzzy shadow,watermark text={Box \thetcbrasternum},
    borderline={.25mm}{-0.5mm}{green!40!black},
    finish={\begin{tcbclipframe}\draw[blue,opacity=0.1,line width=1cm]
      (frame.south west) -- (frame.north east);\end{tcbclipframe}},
    ]
  \begin{tcolorbox}\lipsum[4]\end{tcolorbox}
  \begin{tcolorbox}[blanker]\lipsum[4]\end{tcolorbox}
  \begin{tcolorbox}[blankest]\lipsum[4]\end{tcolorbox}
\end{tcbraster}
```

Box 1

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Box 2

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

/tcb/skin=emptyfirst (skin)

This is a flavor of empty → P. 226 which is used as a *first* part in a break sequence for empty → P. 226. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'emptyfirst'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	empty
/tcb/interior titled engine → P. 126:	empty
/tcb/interior engine → P. 127:	empty
/tcb/segmentation engine → P. 127:	empty
/tcb/title engine → P. 127:	empty

```
\begin{tcbraster}[empty,skin=emptyfirst,raster equal height,raster columns=4,
    coltitle=Navy,borderline={2pt}{0pt}{black!10!white},
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=emptymiddle (skin)

This is a flavor of empty → P. 226 which is used as a *middle* part in a break sequence for empty → P. 226. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'emptymiddle'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	empty
/tcb/interior titled engine → P. 126:	empty
/tcb/interior engine → P. 127:	empty
/tcb/segmentation engine → P. 127:	empty
/tcb/title engine → P. 127:	empty

```
\begin{tcbraster}[empty,skin=emptymiddle,raster equal height,raster columns=4,
    coltitle=Navy,borderline={2pt}{0pt}{black!10!white},
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=emptylast (skin)

This is a flavor of empty → P. 226 which is used as a *last* part in a break sequence for empty → P. 226. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'emptylast'

/tcb/graphical environment → P. 125:	tikzpicture
/tcb/frame engine → P. 126:	empty
/tcb/interior titled engine → P. 126:	empty
/tcb/interior engine → P. 127:	empty
/tcb/segmentation engine → P. 127:	empty
/tcb/title engine → P. 127:	empty

```
\begin{tcbraster}[empty,skin=emptylast,raster equal height,raster columns=4,
    coltitle=Navy,borderline={2pt}{0pt}{black!10!white},
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

This example demonstrates a breakable customized box. Here, we define an environment `freebox`. The first application of `freebox` produces an unbroken `tcolorbox`. The box is drawn by the code given by `/tcb/frame code` → P. 129 and `/tcb/interior code` → P. 130. The second application of `freebox` is broken into several parts which are drawn by the codes given by `/tcb/skin first is subskin of` → P. 132, `/tcb/skin middle is subskin of` → P. 132, and `/tcb/skin last is subskin of` → P. 132.

```
% Preamble:
%\usepackage{tikz,lipsum}
%\tcbuselibrary{skins,breakable}
\tikzset{coltria/.style={fill=red!15!white}}

\newtcolorbox{freebox}[1][]{empty,breakable,leftrule=5mm,left=2mm,
  frame style={fill,top color=red!75!black,bottom color=red!75!black,middle color=red},
  colback=yellow!50!white,
  watermark color=red!50!yellow!75!white,
  watermark text on=unbroken is unbroken box,
  watermark text on=first is first part,
  watermark text on=middle is middle part,
  watermark text on=last is last part,
  % code for unbroken boxes:
  frame code={\path[tcb fill frame] (frame.south west)--(frame.north west)
    --([xshift=-5mm]frame.north east)--([yshift=-5mm]frame.north east)
    --([yshift=5mm]frame.south east)--([xshift=-5mm]frame.south east)--cycle; },
  interior code={\path[tcb fill interior] (interior.south west)--(interior.north west)
    --([xshift=-4.8mm]interior.north east)--([yshift=-4.8mm]interior.north east)
    --([yshift=4.8mm]interior.south east)--([xshift=-4.8mm]interior.south east)
    --cycle; },
  % code for the first part of a break sequence:
  skin first is subskin of={emptyfirst}{%
    frame code={\path[tcb fill frame] (frame.south west)--(frame.north west)
      --([xshift=-5mm]frame.north east)--([yshift=-5mm]frame.north east)
      --(frame.south east)--cycle;
      \path[coltria] ([xshift=2.5mm,yshift=1mm]frame.south west) -- +(120:2mm)
      -- +(60:2mm)-- cycle; },
    interior code={\path[tcb fill interior] (interior.south west|-frame.south)
      --(interior.north west)--([xshift=-4.8mm]interior.north east)
      --([yshift=-4.8mm]interior.north east)--(interior.south east|-frame.south)
      --cycle; },
  },%
  % code for the middle part of a break sequence:
  skin middle is subskin of={emptymiddle}{%
    frame code={\path[tcb fill frame] (frame.south west)--(frame.north west)
      --(frame.north east)--(frame.south east)--cycle;
      \path[coltria] ([xshift=2.5mm,yshift=-1mm]frame.north west) -- +(240:2mm)
        -- +(300:2mm) -- cycle;
      \path[coltria] ([xshift=2.5mm,yshift=1mm]frame.south west) -- +(120:2mm)
        -- +(60:2mm) -- cycle;
      },
    interior code={\path[tcb fill interior] (interior.south west|-frame.south)
      --(interior.north west|-frame.north)--(interior.north east|-frame.north)
      --(interior.south east|-frame.south)--cycle; },
    },
  % code for the last part of a break sequence:
  skin last is subskin of={emptylast}{%
    frame code={\path[tcb fill frame] (frame.south west)--(frame.north west)
      --(frame.north east)--([yshift=5mm]frame.south east)
      --([xshift=-5mm]frame.south east)--cycle;
      \path[coltria] ([xshift=2.5mm,yshift=-1mm]frame.north west) -- +(240:2mm)
      -- +(300:2mm) -- cycle;
      },
    interior code={\path[tcb fill interior] (interior.south west)
```

```
      --(interior.north west|-frame.north)--(interior.north east|-frame.north)
      --([yshift=4.8mm]interior.south east)--([xshift=-4.8mm]interior.south east)
      --cycle; },
    },
  #1}

\begin{freebox}
\lipsum[1]
\end{freebox}

\begin{freebox}
\lipsum[1-12]
\end{freebox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

unbroken box

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

first part

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim.

Vestibulum pellentesque felis eu massa.
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.
Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.
Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.
Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.
Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.
Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus. Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et,

middle part

lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

last part

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

10.19 Skin 'spartan'

/tcb/skin=spartan (skin)

This skin is quite ... spartan. It supports no rounded corners, no overlays, no shadows, no borderlines, and no finishes. The only exception are underlays. One cannot do very fancy things with this skin, but it compiles very fast. Therefore, the `spartan` skin is used for the draft mode, see Section 10.11 on page 192. Nevertheless, it can be used as a normal skin.

Environment and engines for the skin 'spartan'

/tcb/graphical environment →P. 125:	tikzpicture
/tcb/frame engine →P. 126:	spartan
/tcb/interior titled engine →P. 126:	spartan
/tcb/interior engine →P. 127:	spartan
/tcb/segmentation engine →P. 127:	spartan
/tcb/title engine →P. 127:	spartan

/tcb/spartan (style, no value)

This is an abbreviation for setting `skin=spartan`.

```
\begin{tcbraster}[spartan,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacktitle=LimeGreen!75!DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

10.20 Skin 'draft'

/tcb/skin=draft (skin)

This skin is intended to be used while drafting new geometric settings for a `tcolorbox`.

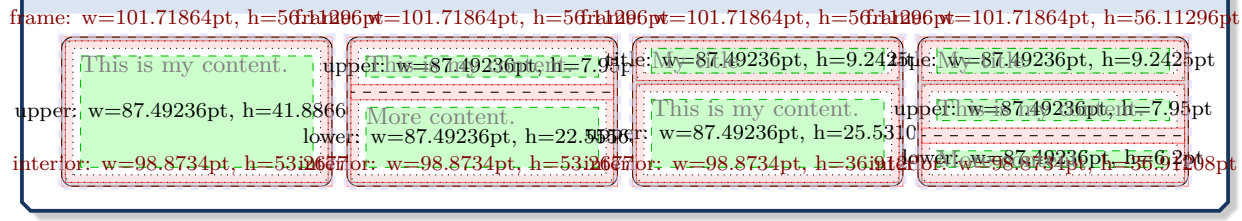
Environment and engines for the skin 'draft'

/tcb/graphical environment → P. 125:	`tikzpicture`
/tcb/frame engine → P. 126:	*special*
/tcb/interior titled engine → P. 126:	*special*
/tcb/interior engine → P. 127:	*special*
/tcb/segmentation engine → P. 127:	`path`
/tcb/title engine → P. 127:	`path`

/tcb/draft (style, no value)

This is an abbreviation for setting `skin=draft`.

```
\begin{tcbraster}[draft,raster equal height,raster columns=4,
    colback=LightGreen,colframe=DarkGreen,colbacktitle=LimeGreen!75!DarkGreen,
    left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm]
  \begin{tcolorbox}
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
  \end{tcolorbox}
  \begin{tcolorbox}[adjusted title=My title]
    This is my content.
    \tcblower
    More content.
  \end{tcolorbox}
\end{tcbraster}
```



```
\vspace*{3mm}
\begin{tcolorbox}[draft,title=A colored box with the 'draft' skin]
\lipsum[1-3]
\tcblower
\lipsum[4-6]
\end{tcolorbox}
```

frame: w=423.94617pt, h=506.20157pt

A colored box with the 'draft' skin title: w=392.64822pt, h=6.2pt

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

upper: w=392.64822pt, h=249.95pt

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

lower: w=392.64822pt, h=205.95pt

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

interior: w=421.10092pt, h=490.04318pt

10.21 Skin Family 'freelance'

! This skin family 'freelance' is deprecated with `tcolorbox` 3.00. It is not longer needed, because `/tcb/frame code` → P. 129, `/tcb/interior code` → P. 130, `/tcb/interior titled code` → P. 129, and `/tcb/title code` → P. 131 can be applied to every skin now. In this sense, everything has become *freelance* now.
For users of `/tcb/freelance`: Old code should continue to work. There may be exceptions for breakable freelance boxes under certain circumstances. For new code, use `/tcb/empty` → P. 226 or `/tcb/enhanced` → P. 195 where you would have used `/tcb/freelance` before.

`/tcb/skin=freelance` (skin)

This skin gives full freedom for the appearance of the `tcolorbox`. All drawing engines are set to type `freelance`; they use the `tikz` package and compute the `/tcb/geometry nodes` → P. 128.

Environment and engines for the skin 'freelance'

`/tcb/graphical environment` → P. 125:	`tikzpicture`
`/tcb/frame engine` → P. 126:	`freelance`
`/tcb/interior titled engine` → P. 126:	`freelance`
`/tcb/interior engine` → P. 127:	`freelance`
`/tcb/segmentation engine` → P. 127:	`freelance`
`/tcb/title engine` → P. 127:	`freelance`

`/tcb/freelance` (style, no value)

This is an abbreviation for setting `skin=freelance`.

`/tcb/skin=freelancefirst` (skin)

This skin equals `freelance` with exception of the break sequence, see Section 17.7 on page 362.

`/tcb/skin=freelancemiddle` (skin)

This skin equals `freelance` with exception of the break sequence, see Section 17.7 on page 362.

`/tcb/skin=freelancelast` (skin)

This skin equals `freelance` with exception of the break sequence, see Section 17.7 on page 362.

`/tcb/extend freelance=`⟨*options*⟩ (no default, initially empty)

The ⟨*options*⟩ are added to the skin definition of `freelance`.

`/tcb/extend freelancefirst=`⟨*options*⟩ (no default, initially empty)

The ⟨*options*⟩ are added to the skin definition of `freelancefirst` which is used as first part of the break sequence of `freelance`. See `/tcb/skin first is subskin of` → P. 132 for a substitute of this key.

`/tcb/extend freelancemiddle=`⟨*options*⟩ (no default, initially empty)

The ⟨*options*⟩ are added to the skin definition of `freelancemiddle` which is used as middle part of the break sequence of `freelance`. See `/tcb/skin middle is subskin of` → P. 132 for a substitute of this key.

`/tcb/extend freelancelast=`⟨*options*⟩ (no default, initially empty)

The ⟨*options*⟩ are added to the skin definition of `freelancelast` which is used as last part of the break sequence of `freelance`. See `/tcb/skin last is subskin of` → P. 132 for a substitute of this key.

11 Inclusion of Boxed Image Files

The LIB skins library adds some commands to conveniently include boxed image files. For the following macros and options, the LIB skins library has to be loaded by a package option or inside the preamble by:

```
\tcbuselibrary{skins}
```

See Section 10 on page 139 for the documentation of all other options of the LIB skins library.

11.1 Macros

N 2014-11-14
U 2016-02-19

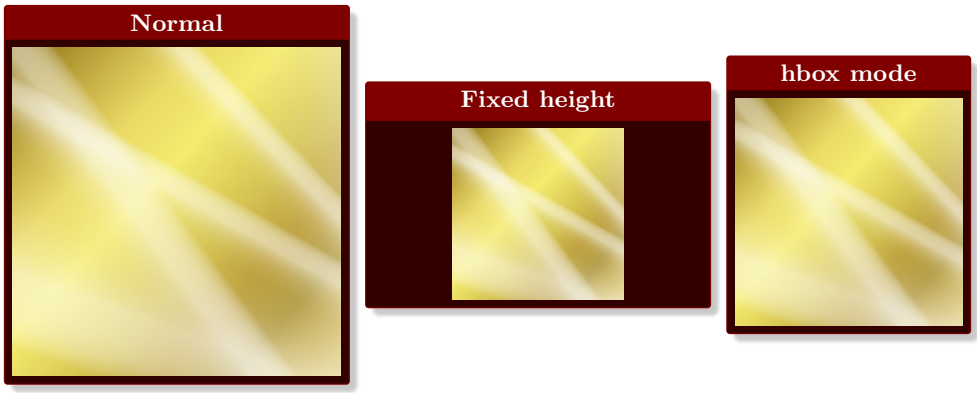
`\tcbincludegraphics[⟨options⟩]{⟨file name⟩}`

In principle, this macro includes an image file denoted by ⟨*file name*⟩ using the standard `\includegraphics` and puts it into a `tcolorbox`→P. 12. The ⟨*options*⟩ are `tcolorbox` keys to set up the colored box. Use `/tcb/graphics options`→P. 243 to specify options for the underlying `\includegraphics`. Some `tcolorbox` option keys are automatically set, namely `/tcb/enhanced`→P. 195 and options to center the image inside the box.

The sizing of the included image is done depending on the following:

- If a `/tcb/width`→P. 34 is specified, but no fixed `/tcb/height`→P. 52, the image is sized to fill the inner width of the box. The height of the box adapts to the image.
- If a fixed `/tcb/height`→P. 52 is specified, the image is sized to fill the fixed inner area of the box.
- If the `/tcb/capture`→P. 90 mode `/tcb/hbox`→P. 90 is specified, the image is sized according to given `\includegraphics` options only. The box adapts to the image.

```
% \tcbuselibrary{raster}
\begin{tcbraster}[raster columns=3,raster force size=false,size=fbox,
    colframe=red!50!black,colback=red!20!black,
    fonttitle=\bfseries,center title,drop fuzzy shadow]
  \tcbincludegraphics[title=Normal]{goldshade.png}
  \tcbincludegraphics[title=Fixed height,height=3cm]{goldshade.png}
  \tcbincludegraphics[title=hbox mode,hbox,graphics options={width=3cm}]
    {goldshade.png}
\end{tcbraster}
```



The auxiliary macro **\imagename** may be used inside \tcbincludegraphics → P. 240 to display the name of the file. **\imagename** is already partially detokenized and is allowed to contain special characters like the underscore. Note that an appropriate font is required to display such characters.

```
% \tcbuselibrary{raster}
\begin{tcbraster}[size=fbox,
    colframe=red!50!black,colback=red!20!black,
    fonttitle=\bfseries\ttfamily,center title,drop fuzzy shadow]
  \tcbincludegraphics[title=\imagename]{goldshade.png}
  \tcbincludegraphics[finish={
    \node[fill=white,fill opacity=0.5,text opacity=1]
    at (frame.center) {\bfseries\ttfamily\imagename};}]{blueshade.png}
\end{tcbraster}
```

N 2014-11-14 `\tcbincludepdf[⟨options⟩]{⟨file name⟩}`

This is a generalized version of `\tcbincludegraphics` → P. 240 which allows to include a complete PDF file denoted by ⟨*file name*⟩. Every page is boxed into an own `tcolorbox` → P. 12 customized by the given ⟨*options*⟩. It is reasonable to put such a series of boxes inside a `tcbraster` → P. 267 for alignment.

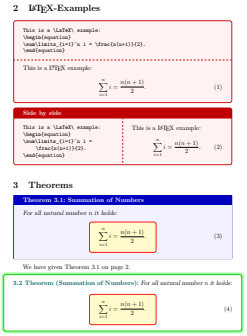
Use `/tcb/graphics pages` → P. 243 to use a selection of pages instead of using the whole file.

The auxiliary macro `\imagepage` may be used inside `\tcbincludepdf` to display the current page number.

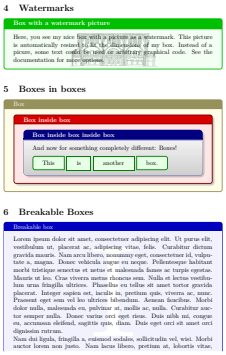
```
% \tcbuselibrary{raster}
\begin{tcbraster}[raster columns=3,colframe=blue,colback=white,
  colbacktitle=blue!50!white,fonttitle=\small\bfseries\ttfamily,
  left=0pt,right=0pt,top=0pt,bottom=0pt,boxsep=0pt,boxrule=0.6pt,
  toptitle=1mm,bottomtitle=1mm,drop lifted shadow,center title,
  graphics pages={1,...,6},title={\imagename\ [\imagepage]}]
\tcbincludepdf{tcolorbox-example.pdf}
\end{tcbraster}
```

tcolorbox-example.pdf [1]

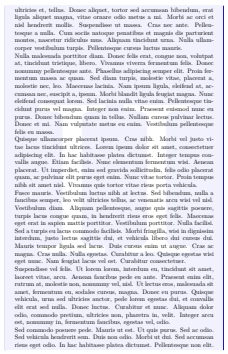
tcolorbox-example.pdf [2]



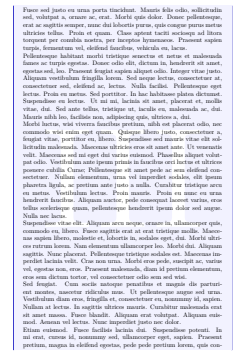
tcolorbox-example.pdf [3]



tcolorbox-example.pdf [4]



tcolorbox-example.pdf [5]



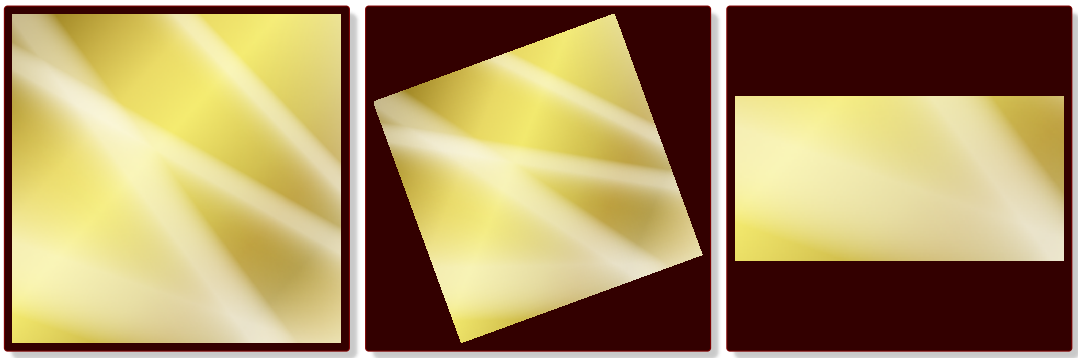
tcolorbox-example.pdf [6]

11.2 Option Keys

N 2014-11-14 **/tcb/graphics options=**⟨*options*⟩ (no default, initially empty)

Used for \tcbincludegraphics → P. 240 and \tcbincludepdf → P. 242 to specify \includegraphics ⟨*options*⟩.

```
% \tcbuselibrary{raster}
\begin{tcbraster}[raster columns=3,size=fbox,raster equal height,
    colframe=red!50!black,colback=red!20!black,drop fuzzy shadow]
  \tcbincludegraphics{goldshade.png}
  \tcbincludegraphics[graphics options={angle=20}]{goldshade.png}
  \tcbincludegraphics[graphics options={viewport=0cm 0cm 8cm 4cm,clip}]
    {goldshade.png}
\end{tcbraster}
```



N 2014-11-14 **/tcb/graphics directory=**⟨*directory*⟩ (no default, initially empty)

Used for \tcbincludegraphics → P. 240 and \tcbincludepdf → P. 242 to specify a file system ⟨*directory*⟩ where the image files are located.

```
\tcbset{
  graphics directory={.},
  graphics directory={examples},
  graphics directory={../../pictures},
}
```

! The \graphicspath macro from the graphics package is superior to this option. /tcb/graphics directory may be used especially for \tcbincludepdf → P. 242.

N 2014-11-14 **/tcb/graphics pages=**⟨*selection*⟩ (no default, initially 1,...,\pdfpages)

Used for \tcbincludepdf → P. 242 to specify a ⟨*selection*⟩ of pages to be included. The largest page number is accessible by **\pdfpages**. The ⟨*selection*⟩ has to be given using the \foreach syntax of Ti*k*Z.

```
\tcbset{
  graphics pages={1,3,7},
  graphics pages={1,...,10},
  graphics pages={1,3,...,18},
  graphics pages={100,...,\pdfpages},
}
```

12 TikZ Image and Picture Fill Extensions; Auxiliary Macros

The LIB skins library adds some image and picture fill options to the vast option set of TikZ [20]. These options can be used in any `tikzpicture`. For the following options, the LIB skins library has to be loaded by a package option or inside the preamble by:

```
\tcbuselibrary{skins}
```

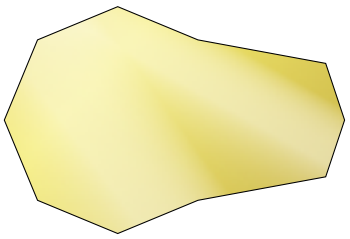
See Section 10 on page 139 for the documentation of all other options of the LIB skins library.

12.1 Fill Plain

`/tikz/fill plain image=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is put in the center of the path, but it is not resized to fit into the path area.

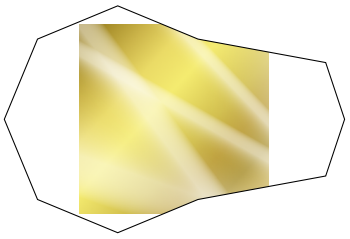
```
\begin{tikzpicture}
\path[draw,fill plain image=goldshade.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill plain image*=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is put in the center of the path, but it is not resized to fit into the path area. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command.

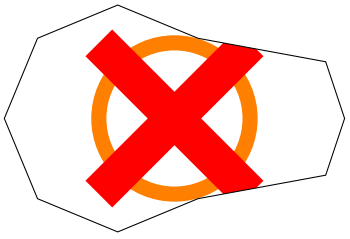
```
\begin{tikzpicture}
\path[draw,fill plain ↩
      ↪image*={width=2.5cm}{goldshade.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill plain picture=`⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with the given ⟨*graphical code*⟩. The result is put in the center of the path, but it is not resized to fit into the path area. Note that this is almost identical to the standard `path picture` option.

```
\begin{tikzpicture}
\path[draw,fill plain picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (1cm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

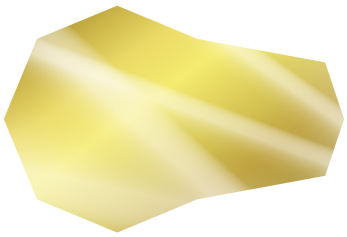


12.2 Fill Stretch

/tikz/fill stretch image=⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is stretched to fill the path area.

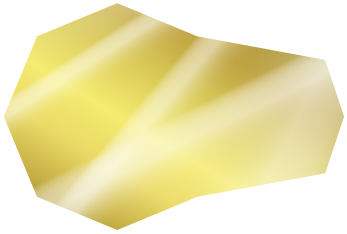
```
\begin{tikzpicture}
\path[fill stretch image=goldshade.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



/tikz/fill stretch image*={⟨*graphics options*⟩}{⟨*file name*⟩} (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. The image is stretched to fill the path area.

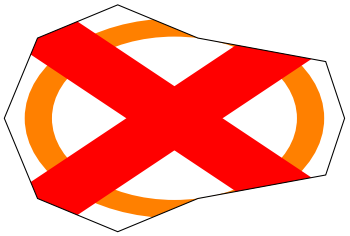
```
\begin{tikzpicture}
\path[fill stretch image*=
  {angle=90,origin=c}{goldshade.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



/tikz/fill stretch picture=⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with the given ⟨*graphical code*⟩. The result is stretched to fill the path area.

```
\begin{tikzpicture}
\path[draw,fill stretch picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (1cm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

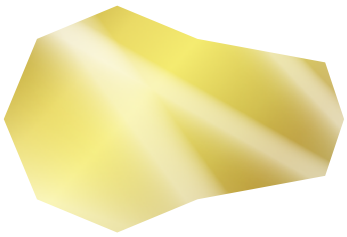


12.3 Fill Overzoom

`/tikz/fill overzoom image=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is zoomed such that the path area fills the image.

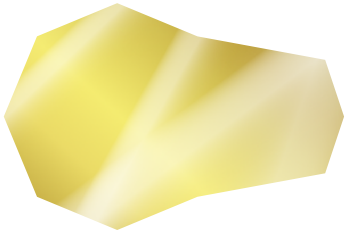
```
\begin{tikzpicture}
\path[fill overzoom image=goldshade.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill overzoom image*={`⟨*graphics options*⟩`}{`⟨*file name*⟩`}` (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. The image is zoomed such that the path area fills the image.

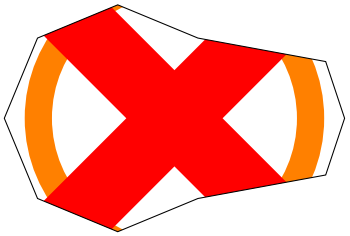
```
\begin{tikzpicture}
\path[fill overzoom image*=
  {angle=90,origin=c}{goldshade.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill overzoom picture=`⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with the given ⟨*graphical code*⟩. The result is zoomed such that the path area fills the image.

```
\begin{tikzpicture}
\path[draw,fill overzoom picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (1cm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

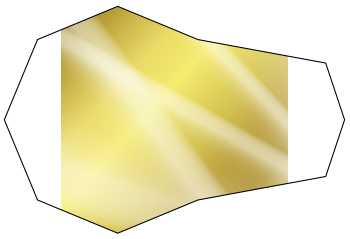


12.4 Fill Zoom

`/tikz/fill zoom image=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is zoomed such that it fits inside the path area. Typically, some parts of the path area will stay unfilled.

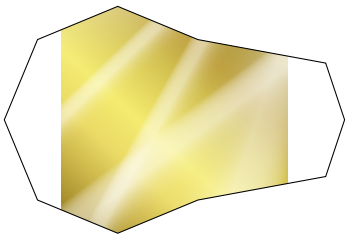
```
\begin{tikzpicture}
\path[draw,fill zoom image=goldshade.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill zoom image*=`{⟨*graphics options*⟩}{⟨*file name*⟩} (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. The image is zoomed such that it fits inside the path area. Typically, some parts of the path area will stay unfilled.

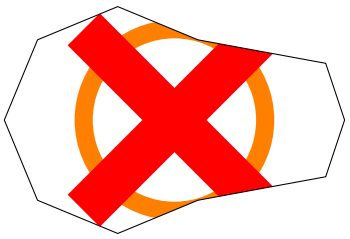
```
\begin{tikzpicture}
\path[draw,fill zoom image*=
  {angle=90,origin=c}{goldshade.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill zoom picture=`⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with the given ⟨*graphical code*⟩. The result is zoomed such that it fits inside the path area. Typically, some parts of the path area will stay unfilled.

```
\begin{tikzpicture}
\path[draw,fill zoom picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (1cm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

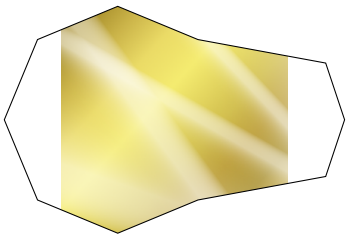


12.5 Fill Shrink

`/tikz/fill shrink image=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is zoomed such that it fits inside the path area, but it never gets enlarged. Typically, some parts of the path area will stay unfilled.

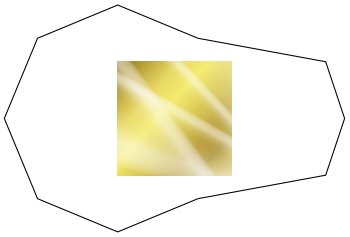
```
\begin{tikzpicture}
\path[draw,fill shrink image=goldshade.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill shrink image*=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. The image is zoomed such that it fits inside the path area, but it never gets enlarged. Typically, some parts of the path area will stay unfilled.

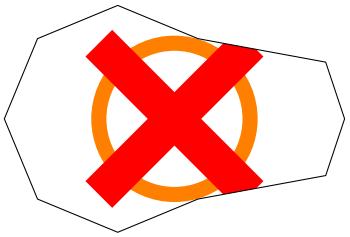
```
\begin{tikzpicture}
\path[draw,fill shrink ⤸
      ⤹image*={width=1.5cm}{goldshade.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill shrink picture=`⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with the given ⟨*graphical code*⟩. The result is zoomed such that it fits inside the path area, but it never gets enlarged. Typically, some parts of the path area will stay unfilled.

```
\begin{tikzpicture}
\path[draw,fill shrink picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (1cm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



12.6 Fill Tile

`/tikz/fill tile image=`⟨*file name*⟩ (no default, initially unset)

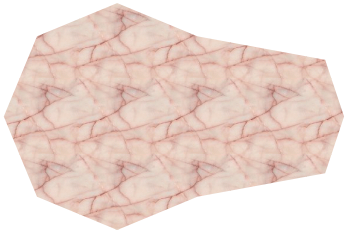
Fills the current path with a tile pattern using an external image referenced by ⟨*file name*⟩.

```
\begin{tikzpicture}
\path[fill tile image=pink_marble.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

`/tikz/fill tile image*={`⟨*graphics options*⟩`}{`⟨*file name*⟩`}` (no default, initially unset)

Fills the current path with a tile pattern using an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command.

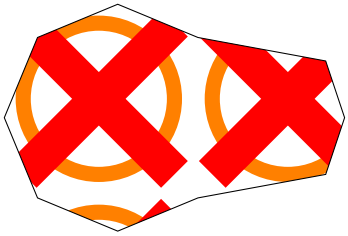
```
\begin{tikzpicture}
\path[fill tile image*={width=1cm}{pink_marble.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill tile picture=`⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with a tile pattern using the given ⟨*graphical code*⟩.

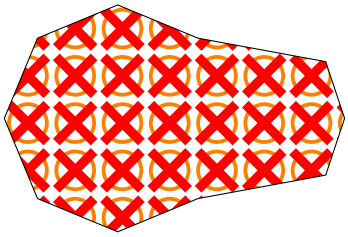
```
\begin{tikzpicture}
\path[draw,fill tile picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (1cm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill tile picture*={`⟨*fraction*⟩`}{`⟨*graphical code*⟩`}` (no default, initially unset)

Fills the current path with a tile pattern using the given ⟨*graphical code*⟩. The graphic is resized by ⟨*fraction*⟩.

```
\begin{tikzpicture}
\path[draw,fill tile picture*={0.25}{%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (1cm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

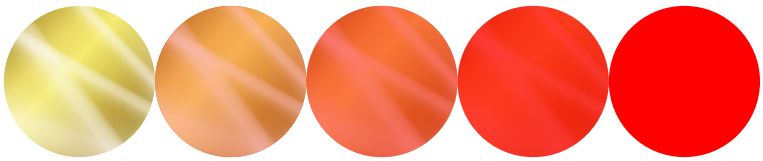


12.7 Filling Options

`/tikz/fill image opacity=`⟨*fraction*⟩ (no default, initially `1.0`)

Sets the fill opacity for the image or picture fill options to the given ⟨*fraction*⟩.

```
\begin{tikzpicture}
\path[fill stretch image=goldshade.png] (0,0) circle (1cm);
\path[fill=red,fill stretch image=goldshade.png,fill image opacity=0.75]
  (2,0) circle (1cm);
\path[fill=red,fill stretch image=goldshade.png,fill image opacity=0.5]
  (4,0) circle (1cm);
\path[fill=red,fill stretch image=goldshade.png,fill image opacity=0.25]
  (6,0) circle (1cm);
\path[fill=red] (8,0) circle (1cm);
\end{tikzpicture}
```



`/tikz/fill image scale=`⟨*fraction*⟩ (no default, initially `1.0`)

Stretches, zooms, overzooms or shrinks the image or picture to the given ⟨*fraction*⟩ of the width and height of the current path.

```
\begin{tikzpicture}
\path[draw,fill zoom image=goldshade.png]
  (0,0) rectangle +(2,2);

\path[draw,fill zoom image=goldshade.png,fill image scale=0.75]
  (3,0) rectangle +(2,2);

\path[draw,fill zoom image=goldshade.png,fill image scale=1.5]
  (6,0) rectangle +(2,2);
\end{tikzpicture}
```

`/tikz/fill image options=`⟨*graphics options*⟩ (no default, initially empty)

The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command for the image fill options. This can be just together with `/tikz/fill stretch image` → P. 245, `/tikz/fill overzoom image` → P. 246, `/tikz/fill zoom image` → P. 247, and `/tikz/fill tile image` → P. 249.

```
\begin{tikzpicture}
\path[fill image options={width=1cm},
  fill tile image=pink_marble.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

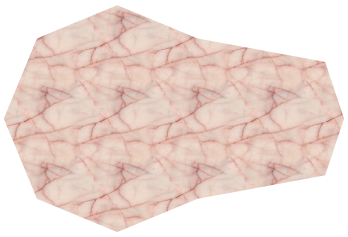
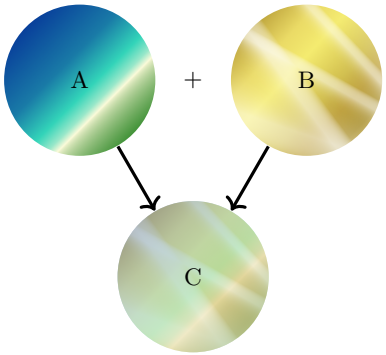


Image blending example

```
\begin{tikzpicture}[every node/.style=
  {circle,minimum width=2cm}]
\node[fill stretch image=blueshade.png]
  (A) at (120:3cm) {A};
\node[fill stretch image=goldshade.png]
  (B) at (60:3cm) {B};
\node[
  preaction={fill stretch image=blueshade.png},
  fill stretch image=goldshade.png,
  fill image opacity=0.5] (C) {C};
\path (A) -- node{$+$} (B);
\draw[->,very thick] (A)--(C);
\draw[->,very thick] (B)--(C);
\end{tikzpicture}
```



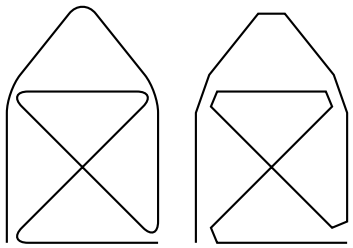
12.8 Straightening of the Arcs

! This patch is considered as an experimental feature. It changes some of the original Ti*k*Z code. This change may break with future updates of Ti*k*Z.

N 2014-05-05 **\tcbpatcharcangular**

The Ti*k*Z package provides a nice `rounded corners` option to replace all corners by little arcs. `\tcbpatcharcangular` is a patch which straightens the arcs. To say it more prosaic, the little arcs are replaced by little straight lines.

```
\begin{tikzpicture}
  \draw[thick,rounded corners=8pt]
    (0,0) -- (0,2) -- (1,3.25) -- (2,2) -- (2,0)
     -- (0,2) -- (2,2) -- (0,0) -- (2,0);
  \tcbpatcharcangular
  \draw[thick,rounded corners=8pt,xshift=2.5cm]
    (0,0) -- (0,2) -- (1,3.25) -- (2,2) -- (2,0)
     -- (0,2) -- (2,2) -- (0,0) -- (2,0);
\end{tikzpicture}
```



N 2014-05-05 **\tcbpatcharcround**

This macro reverts `\tcbpatcharcangular`, i.e., the patch from `\tcbpatcharcangular` is replaced by the original code.

12.9 Extracting Node Dimensions

The following auxiliary macros are defined by the LIB skins library. They allow to determine the width and height of an arbitrary TikZ node. To be more specific, they determine the east-to-west and the north-to-south dimensions which may be not the maximal dimensions for a non-rectangular node. Note that the following dimensions are measured exactly including the line width of the border line. If a new rectangle or node with the same dimensions and a border is to be drawn, this border width has to be substracted.

N 2014-09-19 **\tcbsettowidthofnode**{⟨*register*⟩}{⟨*node*⟩}

Sets the east-to-west dimension of the given ⟨*node*⟩ to the TEX ⟨*register*⟩.

N 2014-09-19 **\tcbsetmacrotowidthofnode**{⟨*macro*⟩}{⟨*node*⟩}

Defines ⟨*macro*⟩ as the east-to-west dimension of the given ⟨*node*⟩.

N 2014-09-19 **\tcbsettoheightofnode**{⟨*register*⟩}{⟨*node*⟩}

Sets the north-to-south dimension of the given ⟨*node*⟩ to the TEX ⟨*register*⟩.

N 2014-09-19 **\tcbsetmacrotoheightofnode**{⟨*macro*⟩}{⟨*node*⟩}

Defines ⟨*macro*⟩ as the north-to-south dimension of the given ⟨*node*⟩.

```
\begin{tikzpicture}
  \node[align=center,draw=red,fill=yellow] (A) {This is my\\example node};
  \tcbsetmacrotowidthofnode\mywidth{A}
  \tcbsetmacrotoheightofnode\myheight{A}
  \path[fill=blue!25!white]           %  rectangle widthout border
    ([xshift=2mm]A.south east)
    rectangle node{Copy} +(\mywidth,\myheight);
  \node[draw=blue,fill=blue!25!white, % standard border width 0.4pt
    minimum width=\mywidth-0.4pt,     % minus width of border
    minimum height=\myheight-0.4pt    % minus height of border
    ]
    at ([xshift=5cm]A) {Copy 2};
\end{tikzpicture}
```

13 Library LIB vignette

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{vignette}
```

This also loads the LIB skins library, see Section 10 on page 139, and the `fadings` library of `tikz` [20].

13.1 Vignette Drawing

N 2016-04-22 `\tcbvignette{⟨options⟩}`

In this context, a *vignette* is a four part rectangular frame. It is constructed as several Ti*k*Z paths and, therefore, can only be used inside a `tikzpicture` environment or inside `tcolorbox` → P. 12 options.

The ⟨*options*⟩ control position, size and style settings of the vignette. Theses options have the common key path `/tcb/vig/` and are described in the following.

The next examples show direct `\tcbvignette` usage without a `tcolorbox` → P. 12.

```
\begin{tikzpicture}
  \tcbvignette{}
\end{tikzpicture}
```

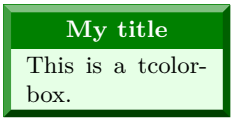
```
\begin{tikzpicture}
  \node[draw,fill=blue!15!white] (A) {Test};
  \tcbvignette{outside node=A,raised color=blue}
\end{tikzpicture}
```

```
\begin{tikzpicture}
  \node[draw,fill=blue!15!white] (A) {Another Test};
  \tcbvignette{size=3mm,outside node=A,
    north style=red,east style=yellow,
    south style=blue,west style=green}
\end{tikzpicture}
```

```
\begin{tikzpicture}
  \node[inner sep=3mm,fill=red!75] (A) {Test};
  \tcbvignette{over node=A,fade in}
\end{tikzpicture}
```

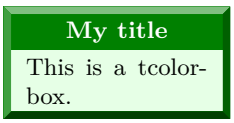
`\tcbvignette` can be used directly inside appropriate options keys for `tcolorbox` → P. 12. Note that options like `/tcb/underlay` → P. 186 need `/tcb/enhanced` → P. 195 or similar settings.

```
\begin{tcolorbox}[enhanced,size=small,sharp corners,
  colback=green!10,colframe=green!50!black,
  boxrule=1mm,titlerule=0mm,
  title=My title,center title,fonttitle=\bfseries,
  underlay={\tcbvignette{size=1mm,inside node=frame,
      raised color=green!50!black}}]
    This is a tcolorbox.
\end{tcolorbox}
```



Mostly, convenient short cuts like /tcb/underlay vignette → P. 260 can be used to add a *vignette* to a tcolorbox → P. 12. Here, \tcbvignette is used internally.

```
\begin{tcolorbox}[enhanced,size=small,sharp corners,
  colback=green!10,colframe=green!50!black,
  boxrule=1mm,titlerule=0mm,
  title=My title,center title,fonttitle=\bfseries,
  underlay vignette]
    This is a tcolorbox.
\end{tcolorbox}
```



13.2 Generic Geometry Settings

N 2016-04-22 **/tcb/vig/xmin=⟨*length*⟩** (no default, initially 0pt)

Sets the lower horizontal limit of a \tcbvignette → P. 253.

N 2016-04-22 **/tcb/vig/xmax=⟨*length*⟩** (no default, initially 1cm)

Sets the upper horizontal limit of a \tcbvignette → P. 253.

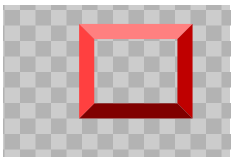
N 2016-04-22 **/tcb/vig/ymin=⟨*length*⟩** (no default, initially 0pt)

Sets the lower vertical limit of a \tcbvignette → P. 253.

N 2016-04-22 **/tcb/vig/ymax=⟨*length*⟩** (no default, initially 1cm)

Sets the upper vertical limit of a \tcbvignette → P. 253.

```
\begin{tikzpicture}
  \fill [black!20] (0,0) rectangle (3,2);
  \path [pattern=checkerboard,pattern color=black!30]
    (0,0) rectangle (3,2);
  \tcbvignette{xmin=1cm,xmax=2.5cm,ymin=0.5cm,ymax=1.75cm}
\end{tikzpicture}
```



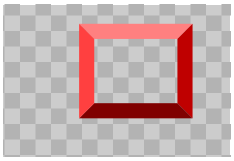
N 2016-04-22 **/tcb/vig/lower left corner=⟨*coordinates*⟩** (style, initially 0,0)

Sets the lower left corner of a \tcbvignette → P. 253. This style sets /tcb/vig/xmin and /tcb/vig/ymin.

N 2016-04-22 **/tcb/vig/upper right corner=⟨*coordinates*⟩** (style, initially 1,1)

Sets the upper right corner of a \tcbvignette → P. 253. This style sets /tcb/vig/xmax and /tcb/vig/ymax.

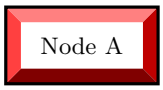
```
\begin{tikzpicture}
  \fill [black!20] (0,0) rectangle (3,2);
  \path [pattern=checkerboard,pattern color=black!30]
    (0,0) rectangle (3,2);
  \tcbvignette{lower left corner={1,0.5},
               upper right corner={2.5,1.75}}
\end{tikzpicture}
```



N 2016-04-22 **/tcb/vig/inside node=⟨*name*⟩** (style, initally unset)

Places the \tcbvignette → P. 253 inside the node with the given ⟨*name*⟩. The outer limits of the *vignette* are adapted to the node geometry.

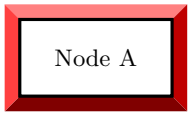
```
\begin{tikzpicture}
  \node[minimum width=2cm,minimum height=1cm] (A) {Node A};
  \tcbvignette{inside node=A}
  \draw[very thick] (A.south west) rectangle (A.north east);
\end{tikzpicture}
```



N 2016-04-22 **/tcb/vig/outside node=**⟨*name*⟩ (style, initally unset)

Places the \tcbvignette → P. 253 outside the node with the given ⟨*name*⟩. The inner limits of the *vignette* are adapted to the node geometry.

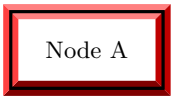
```
\begin{tikzpicture}
  \node[minimum width=2cm,minimum height=1cm] (A) {Node A};
  \tcbvignette{outside node=A}
  \draw[very thick] (A.south west) rectangle (A.north east);
\end{tikzpicture}
```



N 2016-04-22 **/tcb/vig/over node=**⟨*name*⟩ (style, initally unset)

Places the \tcbvignette → P. 253 over the node with the given ⟨*name*⟩. The outer limits of the *vignette* are adapted to the node geometry, but are shifted to the outside by /tcb/vig/over node offset.

```
\begin{tikzpicture}
  \node[minimum width=2cm,minimum height=1cm] (A) {Node A};
  \tcbvignette{over node offset=1mm,over node=A}
  \draw[very thick] (A.south west) rectangle (A.north east);
\end{tikzpicture}
```



N 2016-04-22 **/tcb/vig/over node offset=**⟨*length*⟩ (no default, initially 0.1mm)

Determines the shift value for /tcb/vig/over node. Note that /tcb/vig/over node offset has to be set *before* /tcb/vig/over node is used.

N 2016-04-22 **/tcb/vig/north size=**⟨*length*⟩ (no default, initially 2mm)

Sets the thickness of the north *vignette* part.

```
\begin{tikzpicture}
  \tcbvignette{north size=4mm}
\end{tikzpicture}
```

N 2016-04-22 **/tcb/vig/south size=**⟨*length*⟩ (no default, initially 2mm)

Sets the thickness of the south *vignette* part.

```
\begin{tikzpicture}
  \tcbvignette{south size=4mm}
\end{tikzpicture}
```

N 2016-04-22 **/tcb/vig/east size=**⟨*length*⟩ (no default, initially 2mm)

Sets the thickness of the east *vignette* part.

```
\begin{tikzpicture}
  \tcbvignette{east size=4mm}
\end{tikzpicture}
```

N 2016-04-22 **/tcb/vig/west size=**⟨*length*⟩ (no default, initially 2mm)

Sets the thickness of the west *vignette* part.

```
\begin{tikzpicture}
  \tcbvignette{west size=4mm}
\end{tikzpicture}
```

N 2016-04-22 **/tcb/vig/vertical size=⟨*length*⟩** (style, initially 2mm)

Sets /tcb/vig/north size → P. 255 and /tcb/vig/south size → P. 255, to the given ⟨*length*⟩.

```
\begin{tikzpicture}
  \tcbvignette{vertical size=4mm}
\end{tikzpicture}
```

N 2016-04-22 **/tcb/vig/horizontal size=⟨*length*⟩** (style, initially 2mm)

Sets /tcb/vig/east size → P. 255 and /tcb/vig/west size → P. 255, to the given ⟨*length*⟩.

```
\begin{tikzpicture}
  \tcbvignette{horizontal size=4mm}
\end{tikzpicture}
```

N 2016-04-22 **/tcb/vig/size=⟨*length*⟩** (style, initially 2mm)

Sets /tcb/vig/north size → P. 255, /tcb/vig/south size → P. 255, /tcb/vig/east size → P. 255, and /tcb/vig/west size → P. 255 to the given ⟨*length*⟩.

```
\begin{tikzpicture}
  \tcbvignette{size=4mm}
\end{tikzpicture}
```

! /tcb/vig/north size → P. 255, /tcb/vig/south size → P. 255, etc. have to be set *before* /tcb/vig/outside node → P. 255 is used.

13.3 Generic Color and Style Settings

N 2016-04-22 **/tcb/vig/north style={⟨*style*⟩}** (no default, initially red!50!white)

Sets TikZ ⟨*style*⟩ options for the north *vignette* part.

```
\begin{tikzpicture}
  \tcbvignette{north style=blue}
\end{tikzpicture}
```

N 2016-04-22 **/tcb/vig/south style={⟨*style*⟩}** (no default, initially red!50!black)

Sets TikZ ⟨*style*⟩ options for the south *vignette* part.

```
\begin{tikzpicture}
  \tcbvignette{south style={draw=blue,fill=yellow}}
\end{tikzpicture}
```

N 2016-04-22 **/tcb/vig/east style={⟨*style*⟩}** (no default, initially red!75!black)

Sets TikZ ⟨*style*⟩ options for the east *vignette* part.

```
\begin{tikzpicture}
  \tcbvignette{east style={left color=yellow!75!black,
    right color=blue!75!black}}
\end{tikzpicture}
```

N 2016-04-22 **/tcb/vig/west style={⟨style⟩}** (no default, initially `red!75!white`)

Sets TikZ ⟨*style*⟩ options for the west *vignette* part.

```
\begin{tikzpicture}
  \tcbvignette{west style={preaction={fill=black!20},
    pattern=checkerboard,
    pattern color=black!30}}
\end{tikzpicture}
```

N 2016-04-22 **/tcb/vig/raised color=⟨*color*⟩** (no default)

Creates a raised frame impression by setting the four style options /tcb/vig/north style →P. 256, /tcb/vig/south style →P. 256, /tcb/vig/east style →P. 256, and /tcb/vig/west style to darkened and lightened variations of the given ⟨*color*⟩.

```
\begin{tikzpicture}
  \tcbvignette{raised color=blue}
\end{tikzpicture}
```

N 2016-04-22 **/tcb/vig/lowered color=⟨*color*⟩** (no default)

Creates a lowered frame impression by setting the four style options /tcb/vig/north style →P. 256, /tcb/vig/south style →P. 256, /tcb/vig/east style →P. 256, and /tcb/vig/west style to darkened and lightened variations of the given ⟨*color*⟩.

```
\begin{tikzpicture}
  \tcbvignette{lowered color=green!75!black}
\end{tikzpicture}
```

N 2016-04-22 **/tcb/vig/color from=⟨*inner*⟩ to ⟨*outer*⟩** (no default)

Sets the four style options /tcb/vig/north style →P. 256, /tcb/vig/south style →P. 256, /tcb/vig/east style →P. 256, and /tcb/vig/west style such that the color shades from the ⟨*inner*⟩ color to the ⟨*outer*⟩ color.

```
\begin{tikzpicture}
  \tcbvignette{color from=red to blue!50}
\end{tikzpicture}
```

N 2016-04-22 **/tcb/vig/base color=⟨*color*⟩** (no default)

Sets the base color for /tcb/vig/raised color, /tcb/vig/lowered color, /tcb/finish fading vignette →P. 263. Typically, this value has not to be set directly.

N 2016-04-22 `/tcb/vig/draw method=direct|clipped` (no default, initially `direct`)

Especially, if shadings or fadings are used, the drawn *vignette* graphs are displayed sometimes not as perfect as expected. Glitches and imperfections are very dependent on the previewer software. The `/tcb/vig/draw method` intends to give a choice of alternative drawing methods.

- `direct`: The *vignette* parts are drawn/filled by using a single Ti*k*Z graph. This is the preferred (and default) method for solid color graphs.
- `clipped`: The *vignette* parts are drawn somewhat oversized and are clipped to the intended region. In combination with shadings and fadings this seems to give a better/different optical result (depends on the previewer).

```
\begin{tikzpicture}
  \tcbvignette{color from=red to yellow}
\end{tikzpicture}
```

```
\begin{tikzpicture}
  \tcbvignette{color from=red to yellow,draw method=clipped}
\end{tikzpicture}
```

This option is a stopgap and may be changed or preferably removed in future.

13.4 Generic Fading Settings

The `fadings` library of `tikz` [20] is loaded automatically by the LIB vignette library. Amongst others, the fadings `west`, `east`, `north`, and `south` are defined inside the `fadings` library.

The LIB vignette library adds some more fadings called `semi west`, `semi east`, `semi north`, and `semi south`. These fadings are much *weaker* than the normal fadings.

```
\begin{tikzpicture}
  \fill [black!20] (0,0) rectangle (1,1);
  \path [pattern=checkerboard,pattern color=black!30]
    (0,0) rectangle (1,1);
  \fill [path fading=semi west,blue] (0,0) rectangle (1,1);
\end{tikzpicture}
```

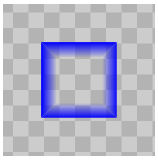
Comparison of the Fadings

west	east
north	south
semi west	semi east
semi north	semi south

N 2016-04-22 **/tcb/vig/fade in={⟨style⟩}** (style, default **white**)

Sets the four style options /tcb/vig/north style →P. 256, /tcb/vig/south style →P. 256, /tcb/vig/east style →P. 256, and /tcb/vig/west style →P. 257 such that the paths fade from outside to inside.

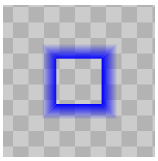
```
\begin{tikzpicture}
  \fill [black!20] (-0.5,-0.5) rectangle (1.5,1.5);
  \path [pattern=checkerboard,pattern color=black!30]
    (-0.5,-0.5) rectangle (1.5,1.5);
  \tcbvignette{fade in=blue}
\end{tikzpicture}
```



N 2016-04-22 **/tcb/vig/fade out={⟨style⟩}** (style, default **white**)

Sets the four style options /tcb/vig/north style →P. 256, /tcb/vig/south style →P. 256, /tcb/vig/east style →P. 256, and /tcb/vig/west style →P. 257 such that the paths fade from inside to outside.

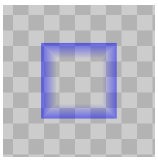
```
\begin{tikzpicture}
  \fill [black!20] (-0.5,-0.5) rectangle (1.5,1.5);
  \path [pattern=checkerboard,pattern color=black!30]
    (-0.5,-0.5) rectangle (1.5,1.5);
  \tcbvignette{fade out=blue}
\end{tikzpicture}
```



N 2016-04-22 **/tcb/vig/semi fade in={⟨style⟩}** (style, default **white**)

Sets the four style options /tcb/vig/north style →P. 256, /tcb/vig/south style →P. 256, /tcb/vig/east style →P. 256, and /tcb/vig/west style →P. 257 such that the paths fade weak from outside to inside.

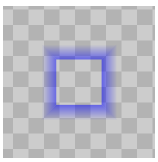
```
\begin{tikzpicture}
  \fill [black!20] (-0.5,-0.5) rectangle (1.5,1.5);
  \path [pattern=checkerboard,pattern color=black!30]
    (-0.5,-0.5) rectangle (1.5,1.5);
  \tcbvignette{semi fade in=blue}
\end{tikzpicture}
```



N 2016-04-22 **/tcb/vig/semi fade out={⟨style⟩}** (style, default **white**)

Sets the four style options /tcb/vig/north style →P. 256, /tcb/vig/south style →P. 256, /tcb/vig/east style →P. 256, and /tcb/vig/west style →P. 257 such that the paths fade weak from inside to outside.

```
\begin{tikzpicture}
  \fill [black!20] (-0.5,-0.5) rectangle (1.5,1.5);
  \path [pattern=checkerboard,pattern color=black!30]
    (-0.5,-0.5) rectangle (1.5,1.5);
  \tcbvignette{semi fade out=blue}
\end{tikzpicture}
```



13.5 Vignette as Underlay

N 2016-04-22 **/tcb/underlay vignette={⟨*options*⟩}** (style, no default)

This puts a \tcbvignette → P. 253 with the given ⟨*options*⟩ as /tcb/underlay → P. 186 to a tcolorbox → P. 12. The dimensions of the *vignette* are matched to the dimensions of the tcolorbox → P. 12. For example, /tcb/leftrule → P. 35 is used as /tcb/vig/west size → P. 255. Also, /tcb/colframe → P. 27 is used as /tcb/vig/raised color → P. 257.
For a /tcb/breakable → P. 351 tcolorbox, the *vignette* is also been broken. Alternatively, \tcbvignette → P. 253 could be used directly inside an /tcb/underlay → P. 186 with appropriate settings.

```
\begin{tcolorbox}[enhanced,size=small,sharp corners,
  colback=green!10,colframe=green!50!black,
  boxrule=2mm,titlerule=0mm,
  title=My title,center title,fonttitle=\bfseries,
  underlay vignette]
    This is a tcolorbox.
\end{tcolorbox}
```

My title
This is a tcolorbox.

```
\begin{tcolorbox}[enhanced,size=small,arc=0pt,
  colback=blue!10,colframe=blue,boxrule=2mm,
  underlay vignette={size=1.5mm}]
    This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

```
\begin{tcolorbox}[enhanced,size=small,sharp corners,
  colframe=red,interior hidden,boxrule=2mm,
  colupper=white,center upper,fontupper=\bfseries,
  underlay vignette]
    This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

```
\begin{tcolorbox}[enhanced,size=small,sharp corners,
  colback=red!50!yellow,frame hidden,boxrule=2mm,
  underlay vignette={color from=red!50!yellow to white,
     draw method=clipped,size=2.1mm}]
    This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

```
\tcbox[enhanced,sharp corners,colback=red!10,colframe=red]
  {Test}

\tcbox[enhanced,sharp corners,colback=red!10,colframe=red,
  underlay vignette]{Test}
```

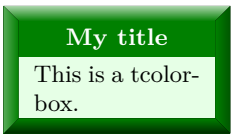
Test

Test

N 2016-04-22 **/tcb/underlay raised shading vignette={⟨*options*⟩}** (style, no default)

This is a special style derived from /tcb/underlay vignette → P. 260, where the frame color is shaded to create a soft raised frame impression.

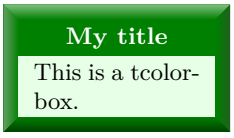
```
\begin{tcolorbox}[enhanced,sharp corners,
  colback=green!10,
  colframe=green!50!black,
  size=small,boxrule=2mm,titlerule=0mm,
  title=My title,center title,fonttitle=\bfseries,
  underlay raised shading vignette]
    This is a tcolorbox.
\end{tcolorbox}
```



N 2016-04-22 **/tcb/underlay raised fading vignette={⟨*options*⟩}** (style, no default)

This style gives a similar effect as /tcb/underlay raised shading vignette, but a path fading is used here. Different optical impression are very previewer-dependent.

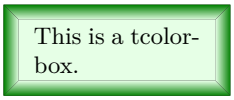
```
\begin{tcolorbox}[enhanced,sharp corners,
  colback=green!10,
  colframe=green!50!black,
  size=small,boxrule=2mm,titlerule=0mm,
  title=My title,center title,fonttitle=\bfseries,
  underlay raised fading vignette]
    This is a tcolorbox.
\end{tcolorbox}
```



N 2016-04-22 **/tcb/underlay shade in vignette={⟨*options*⟩}** (style, no default)

This is a special style derived from /tcb/underlay vignette → P. 260, where the frame color is shaded into the interior color.

```
\begin{tcolorbox}[enhanced,sharp corners,frame hidden,
  colback=green!10,
  colframe=green!50!black,
  size=small,boxrule=2mm,titlerule=0mm,
  underlay shade in vignette]
    This is a tcolorbox.
\end{tcolorbox}
```

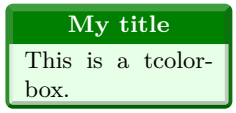


13.6 Vignette as Finish

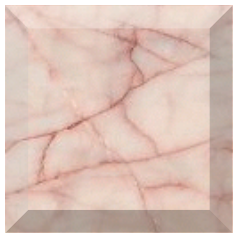
N 2016-04-22 **/tcb/finish vignette={⟨*options*⟩}** (style, no default)

This puts a \tcbvignette → P. 253 with the given ⟨*options*⟩ as /tcb/finish → P. 188 to a tcolorbox → P. 12. The default style settings create a raised frame impression by drawing black and white color parts with reduced opacity.

```
\begin{tcolorbox}[enhanced,size=small,
  colback=green!10,colframe=green!50!black,
  boxrule=0.5mm,titlerule=0mm,
  title=My title,center title,fonttitle=\bfseries,
  finish vignette={size=1mm}]
    This is a tcolorbox.
\end{tcolorbox}
```



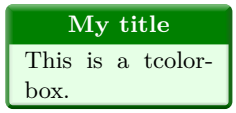

```
\tcbincludegraphics[blankest,width=3cm,
  finish vignette={size=3mm}]{pink_marble.png}
```



N 2016-04-22 **/tcb/finish raised fading vignette={⟨*options*⟩}** (style, no default)

This puts a \tcbvignette → P. 253 with the given ⟨*options*⟩ as /tcb/finish → P. 188 to a tcolorbox → P. 12. The default style settings create a soft raised frame impression by drawing fading black and white color parts.

```
\begin{tcolorbox}[enhanced,size=small,
  colback=green!10,colframe=green!50!black,
  boxrule=0.5mm,titlerule=0mm,
  title=My title,center title,fonttitle=\bfseries,
  finish raised fading vignette={size=1mm}]
    This is a tcolorbox.
\end{tcolorbox}
```




```
\tcbincludegraphics[blankest,width=3cm,
  finish raised fading vignette={size=3mm}]{pink_marble.png}
```

N 2016-04-22 **/tcb/finish fading vignette={⟨*options*⟩}** (style, no default)

This puts a \tcbvignette → P. 253 with the given ⟨*options*⟩ as /tcb/finish → P. 188 to a tcolorbox → P. 12. The default style settings fade the box into white from inside to outside. Note that /tcb/vig/over node → P. 255 is used here. /tcb/vig/over node offset → P. 255 can be adapted to overlap the box more or less. The fade color can be set using /tcb/vig/base color → P. 257.

```
\begin{tcolorbox}[enhanced,size=small,
  colback=green!10,colframe=green!50!black,
  boxrule=0.5mm,titlerule=0mm,
  title=My title,center title,fonttitle=\bfseries,
  finish fading vignette={size=2mm}]
    This is a tcolorbox.
\end{tcolorbox}
```

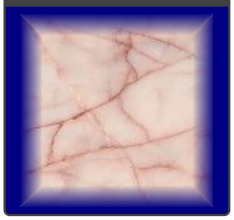
My title

This is a tcolorbox.

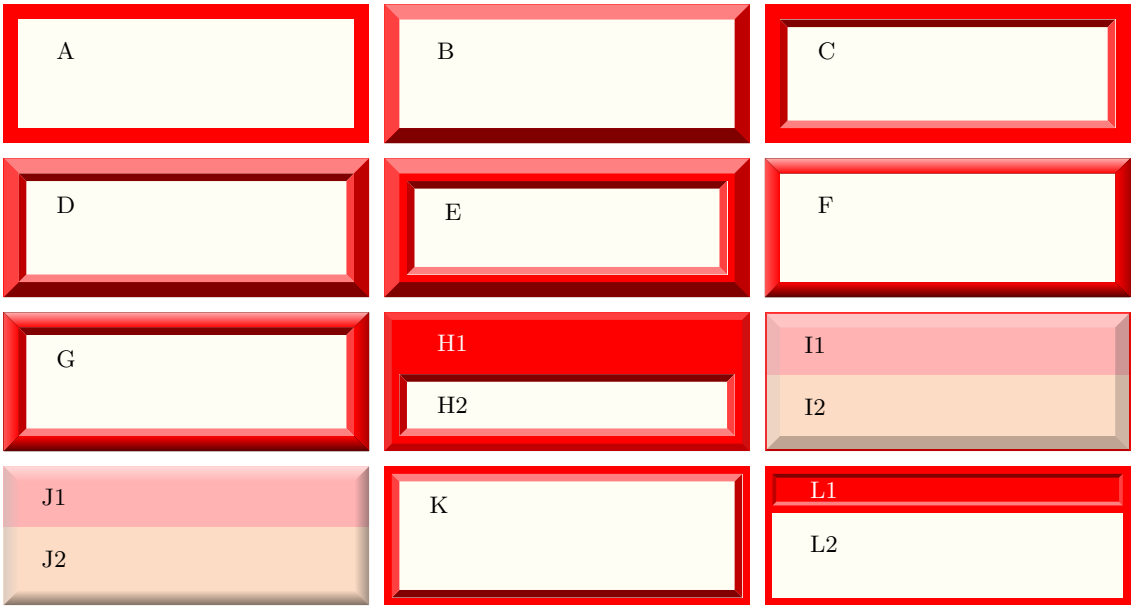
```
\tcbincludegraphics[blankest,width=3cm,
  finish fading vignette={size=3mm}]{pink_marble.png}
```

```
\begin{tcolorbox}[colback=blue!50!black,size=small,
  title=Example]
\tcbincludegraphics[blankest,
  finish fading vignette={base color=blue!50!black,size=3mm,
    over node offset=0.2mm}]{pink_marble.png}
\end{tcolorbox}
```

Example



```
\begin{tcbitemize}[raster columns=3,bicolor,
  raster equal height,sharp corners,boxrule=2mm,
  colframe=red,colback=yellow!5,colbacklower=yellow!25!red!20]
\tcbitem A
\tcbitem[underlay vignette] B
\tcbitem[underlay={\tcbvignette{inside node=interior,
  lowered color=red,size=1mm}}] C
\tcbitem[underlay vignette,
  underlay={\tcbvignette{inside node=interior,
  lowered color=red,size=1mm}}] D
\tcbitem[boxrule=3mm,underlay vignette={size=2mm},
  underlay={\tcbvignette{inside node=interior,
  lowered color=red,size=1mm}}] E
\tcbitem[underlay raised shading vignette] F
\tcbitem[underlay raised shading vignette,
  underlay={\tcbvignette{inside node=interior,
  lowered color=red,size=1mm}}] G
\tcbitem[title=H1,underlay={\tcbvignette{inside node=interior,
  lowered color=red,size=1mm}},finish vignette] H2
\tcbitem[boxrule=0.25mm,colback=red!30,finish vignette] I1 \tcblower I2
\tcbitem[tile,colback=red!30,finish raised fading vignette] J1 \tcblower J2
\tcbitem[boxrule=1mm,underlay={\tcbvignette{inside node=interior,
  raised color=red,size=1mm}}] K
\tcbitem[boxrule=1mm,title=L1,underlay={\tcbvignette{inside node=title,
  lowered color=red,size=0.5mm}}] L2
\end{tcbitemize}
```



14 Library raster

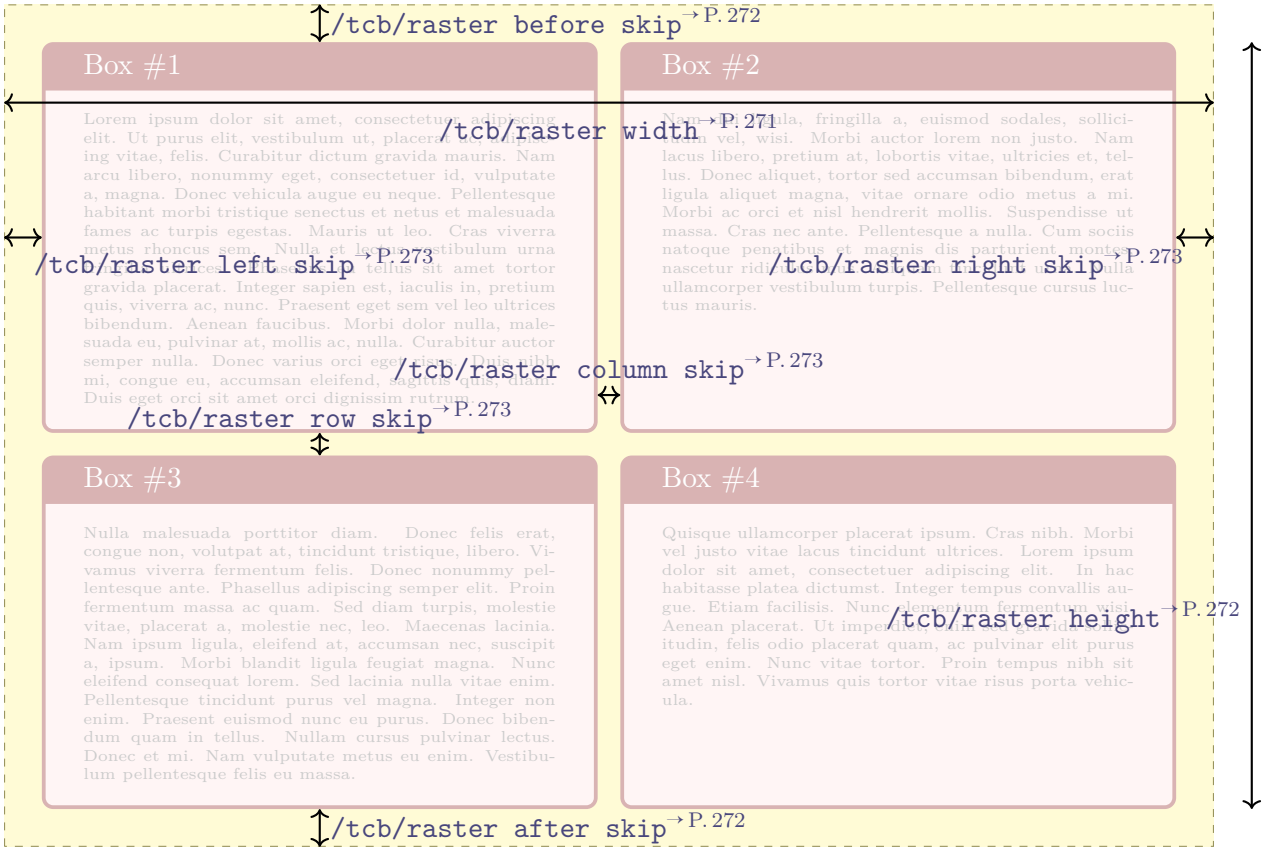
The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{raster}
```

14.1 Concept of Rasters

A *raster* is used to align several colored boxes in a regular way. It can be seen as a far related counterpart to the `matrix` construct of Ti*k*Z, but it differs in many aspects.

In principle, `tcolorbox`es are arranged in rows and columns when put inside a `tcbraster` → P. 267 environment. The boxes are fluently added to the raster like adding text to a paragraph. Especially, line/row breaks are done automatically and one cannot end a line/row ahead of schedule. Further, a *raster* is not restricted to a single page but may break into an arbitrary series of pages.



```
\begin{tcbraster}[raster columns=3,raster rows=3,raster height=\linewidth,
  enhanced,size=small,sharp corners,arc=8mm,colframe=red!50!black,
  colback=yellow!10!white,watermark overzoom=1.0,fit algorithm=hybrid* ]
  \begin{tcolorbox}[rounded corners=northwest,boxrule=0pt,
    watermark graphics=lichtspiel.jpg]\end{tcolorbox}
  \tcboxfit{\lipsum[1]}
  \begin{tcolorbox}[rounded corners=northeast,boxrule=0pt,
    watermark graphics=goldshade.png]\end{tcolorbox}
  \tcboxfit{\lipsum[2]}
  \begin{tcolorbox}[valign=center,halign=center]Nine Boxes.\end{tcolorbox}
  \tcboxfit{\lipsum[3]}
  \begin{tcolorbox}[rounded corners=southwest,boxrule=0pt,
    watermark graphics=goldshade.png]\end{tcolorbox}
  \tcboxfit{\lipsum[4]}
  \begin{tcolorbox}[rounded corners=southeast,boxrule=0pt,
    watermark graphics=lichtspiel.jpg]\end{tcolorbox}
\end{tcbraster}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nine Boxes.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

14.2 Macros of the Library

N 2014-11-10

```
\begin{tcbraster}[⟨options⟩]
  ⟨environment content⟩
\end{tcbraster}
```

A raster arranges enclosed boxes in a regular way, mainly into rows and columns. The ⟨*options*⟩ are used to control the raster parameters and to set the properties for the enclosed boxes.

- The *raster* is only allowed to contain a series of `tcolorbox` → P. 12 environments or derived constructs. With some small restrictions, boxes created with `\tcboxfit` → P. 380 can also be added. Boxes created with `\tcbox` → P. 14 are not reasonable here, but may be used to a certain degree.
- Do not add anything else between the boxes inside the raster with exception of white-space. Especially, do not use `\\` or `\par` to end a row; row breaks are done automatically.
- The boxes inside a raster are numbered automatically. `\thetcbrasternum` may be used inside a box to access this number.

```
\begin{tcbraster}[raster columns=3, raster equal height,
  size=small,colframe=red!50!black,colback=red!10!white,colbacktitle=red!50!white,
  title={Box \# \thetcbrasternum}]
  \begin{tcolorbox}First box\end{tcolorbox}
  \begin{tcolorbox}Second box\end{tcolorbox}
  \begin{tcolorbox}This is a box\\with a second line\end{tcolorbox}
  \begin{tcolorbox}Another box\end{tcolorbox}
  \begin{tcolorbox}A box again\end{tcolorbox}
\end{tcbraster}
```

Box # 1
First box

Box # 2
Second box

Box # 3
This is a box
with a second line

Box # 4
Another box

Box # 5
A box again

```
\begin{tcbraster}[raster columns=2, raster equal height=rows,
  size=small,colframe=red!50!black,colback=red!10!white,colbacktitle=red!50!white,
  title={Box \# \thetcbrasternum}]
  \begin{tcolorbox}First box\end{tcolorbox}
  \begin{tcolorbox}Second box\end{tcolorbox}
  \begin{tcolorbox}This is a box\\with a second line\end{tcolorbox}
  \begin{tcolorbox}Another box\end{tcolorbox}
  \begin{tcolorbox}A box again\end{tcolorbox}
\end{tcbraster}
```

Box # 1
First box

Box # 2
Second box

Box # 3
This is a box
with a second line

Box # 4
Another box

Box # 5
A box again

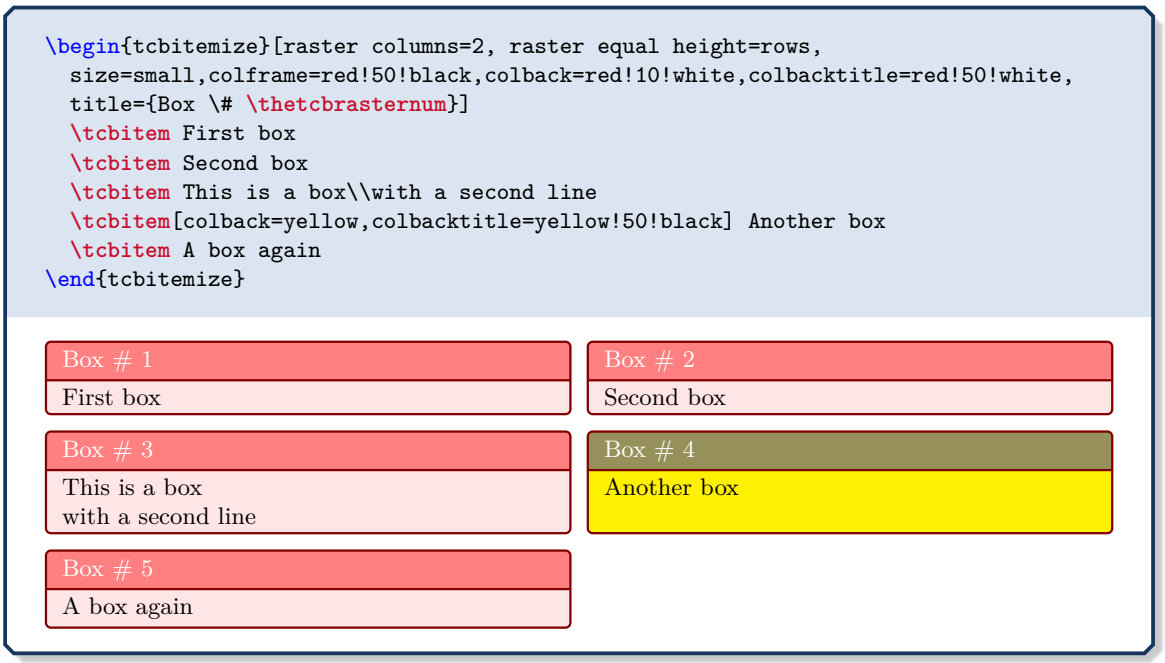
N 2014-11-10 `\begin{tcbitemize}[⟨options⟩]`
⟨*environment content*⟩
`\end{tcbitemize}`

This is a special case of a `tcbraster` → P. 267 with the given ⟨*options*⟩.

- Here, the enclosed boxes are created using `\tcbitem`.
- There has to be at least one `\tcbitem`.
- One cannot use anything else than `\tcbitem` to add something to the *raster*.

This leads to a very compact syntax.

```
\begin{tcbitemize}[raster columns=2, raster equal height=rows,
  size=small,colframe=red!50!black,colback=red!10!white,colbacktitle=red!50!white,
  title={Box \# \thetcbrasternum}]
  \tcbitem First box
  \tcbitem Second box
  \tcbitem This is a box\\with a second line
  \tcbitem[colback=yellow,colbacktitle=yellow!50!black] Another box
  \tcbitem A box again
\end{tcbitemize}
```



! `tcbitemize` has more restrictions than `tcbraster` → P. 267. Especially, the `/tcb/capture` → P. 90 mode has to be `minipage`. For example, `/tcb/fit` → P. 382 cannot be used safely. If `/tcb/fit` → P. 382 should be used, turn over to `tcbraster` → P. 267.

N 2014-11-10 `\tcbitem[⟨options⟩]`

Used inside `tcbitemize` to create a new `tcolorbox` → P. 12 with the given ⟨*options*⟩.

N 2016-02-19

```
\begin{tcboxedraster}[⟨raster options⟩]{⟨box options⟩}
  ⟨environment content⟩
\end{tcboxedraster}
```

This is a convenience environment which combines a `tcolorbox` → P. 12 with an embedded `tcbraster` → P. 267. The ⟨*box options*⟩ are given to the outer `tcolorbox` → P. 12, while the ⟨*raster options*⟩ are given to the embedded `tcbraster` → P. 267. This environment is especially useful for rasters inside rasters.

```
\begin{tcboxedraster}[raster columns=3, raster equal height,
  size=small,colframe=red!50!black,colback=red!10!white,colbacktitle=red!50!white,
  title={Box \# \thetcbrasternum}]
  {colback=yellow!10,fonttitle=\bfseries,title=Boxed Raster}
  \begin{tcolorbox}First box\end{tcolorbox}
  \begin{tcolorbox}Second box\end{tcolorbox}
  \begin{tcolorbox}This is a box\\with a second line\end{tcolorbox}
  \begin{tcolorbox}Another box\end{tcolorbox}
  \begin{tcolorbox}A box again\end{tcolorbox}
\end{tcboxedraster}
```

Boxed Raster

Box # 1	Box # 2	Box # 3
First box	Second box	This is a box with a second line
Box # 4	**Box # 5**	
Another box	A box again	

```
% \tcbuselibrary{skins}
\begin{tcbraster}[raster columns=2, raster equal height,
  raster every box/.style={size=small,colframe=red!50!black,colback=red!10!white,
    valign=center,halign=center}]
  \begin{tcolorbox}One\end{tcolorbox}
  \begin{tcolorbox}Two\end{tcolorbox}
    \begin{tcboxedraster}{blankest}
      \begin{tcolorbox}Three\end{tcolorbox}
      \begin{tcolorbox}Four\end{tcolorbox}
      \begin{tcolorbox}Five\end{tcolorbox}
      \begin{tcolorbox}Six\end{tcolorbox}
    \end{tcboxedraster}
    \begin{tcolorbox}Seven\end{tcolorbox}
\end{tcbraster}
```

One

Two

Three

Four

Five

Six

Seven

N 2016-04-27

```
\begin{tcboxeditemize}[⟨raster options⟩]{⟨box options⟩}
  ⟨environment content⟩
\end{tcboxeditemize}
```

This is a convenience environment which combines a `tcolorbox` → P. 12 with an embedded `tcbitemize` → P. 268. The ⟨*box options*⟩ are given to the outer `tcolorbox` → P. 12, while the ⟨*raster options*⟩ are given to the embedded `tcbitemize` → P. 268. This environment is especially useful for rasters inside rasters.

```
\begin{tcboxeditemize}[raster columns=3, raster equal height,
  size=small,colframe=red!50!black,colback=red!10!white,colbacktitle=red!50!white,
  title={Box \# \thetcbrasternum}]
  {colback=yellow!10,fonttitle=\bfseries,title=Boxed Itemize}
  \tcbitem First box
  \tcbitem Second box
  \tcbitem This is a box\\with a second line
  \tcbitem Another box
  \tcbitem A box again
\end{tcboxeditemize}
```

Boxed Itemize

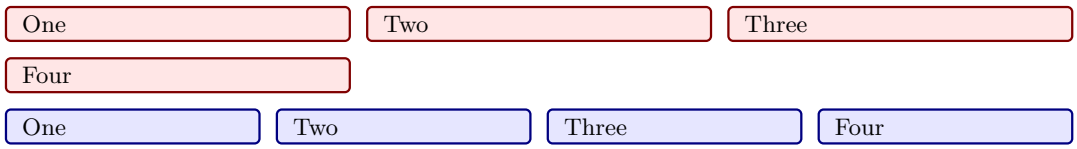
- Box # 1: First box
- Box # 2: Second box
- Box # 3: This is a box with a second line
- Box # 4: Another box
- Box # 5: A box again

14.3 Option Keys of the Library

N 2014-11-10 **/tcb/raster columns**=⟨*number*⟩ (no default, initially 2)

Sets the ⟨*number*⟩ of columns for a *raster*.

```
\begin{tcbitemize}[raster columns=3,
  size=small,colframe=red!50!black,colback=red!10!white]
  \tcbitem One
  \tcbitem Two
  \tcbitem Three
  \tcbitem Four
\end{tcbitemize}
\begin{tcbitemize}[raster columns=4,
  size=small,colframe=blue!50!black,colback=blue!10!white]
  \tcbitem One
  \tcbitem Two
  \tcbitem Three
  \tcbitem Four
\end{tcbitemize}
```



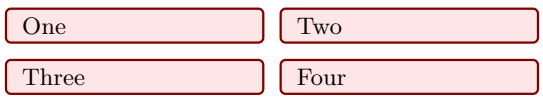
N 2014-11-10 **/tcb/raster rows**=⟨*number*⟩ (no default, initially 2)

Sets the ⟨*number*⟩ of rows for a *raster*. Note that this is only relevant in connection with setting /tcb/raster height → P. 272 to a value greater than 0pt. Then, it defines the number of rows *per* given height.

N 2014-11-10 **/tcb/raster width**=⟨*length*⟩ (no default, initially \linewidth)

Sets the total raster width to the given ⟨*length*⟩. /tcb/raster left skip → P. 273 and /tcb/raster right skip → P. 273 are part of the total width.

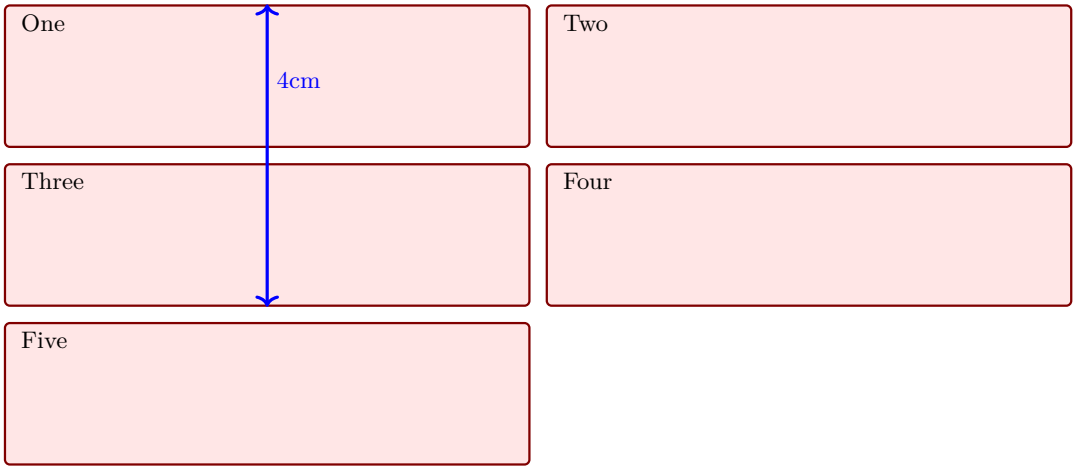
```
\begin{tcbitemize}[raster width=\linewidth/2,
  size=small,colframe=red!50!black,colback=red!10!white]
  \tcbitem One
  \tcbitem Two
  \tcbitem Three
  \tcbitem Four
\end{tcbitemize}
```



N 2014-11-10 **/tcb/raster height=⟨*length*⟩** (no default, initially 0pt)

Sets the raster height *per* /tcb/raster rows → P. 271 to the given ⟨*length*⟩. This forces an appropriate height for the enclosed boxes. /tcb/raster before skip and /tcb/raster after skip are not part of this calculation. If the ⟨*length*⟩ is set to 0pt, this feature is deactivated.

```
\begin{tcbitemize}[raster height=4cm, raster rows=2,
  size=small,colframe=red!50!black,colback=red!10!white]
  \tcbitem One
  \tcbitem Two
  \tcbitem[enhanced,
    finish={\draw[blue,very thick,<->] (frame.south)
      -- node[right,pos=.75]{4cm} +(0,4); }]
    Three
  \tcbitem Four
  \tcbitem Five
\end{tcbitemize}
```



N 2014-11-10
U 2014-12-16 **/tcb/raster before skip=⟨*glue*⟩** (no default, initially 2mm)

Space of the given ⟨*glue*⟩ is inserted vertically before the *raster*. This space is discardable.

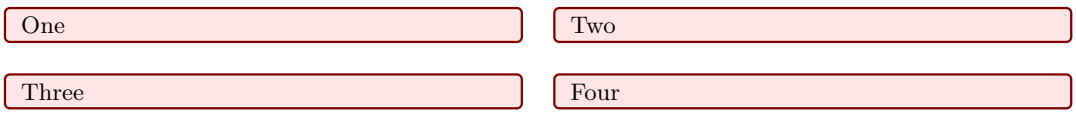
N 2014-11-10
U 2014-12-16 **/tcb/raster after skip=⟨*glue*⟩** (no default, initially 2mm)

Space of the given ⟨*glue*⟩ is inserted vertically after the *raster*. This space is discardable.

N 2015-01-08 **/tcb/raster equal skip=⟨*length*⟩** (style, no default)

Shortcut to set /tcb/raster before skip, /tcb/raster after skip, /tcb/raster column skip → P. 273, and /tcb/raster row skip → P. 273 to the same ⟨*length*⟩ value.

```
\begin{tcbitemize}[raster equal skip=4mm,
  size=small,colframe=red!50!black,colback=red!10!white]
  \tcbitem One
  \tcbitem Two
  \tcbitem Three
  \tcbitem Four
\end{tcbitemize}
```



N 2014-11-10 **/tcb/raster left skip=⟨*length*⟩** (no default, initially 0pt)

Space of the given ⟨*length*⟩ is inserted horizontally left of the *raster*.

```
\begin{tcbitemize}[raster left skip=2cm,
  size=small,colframe=red!50!black,colback=red!10!white]
  \tcbitem One
  \tcbitem Two
  \tcbitem Three
  \tcbitem Four
\end{tcbitemize}
```

One | Two

Three | Four

N 2014-11-10 **/tcb/raster right skip=⟨*length*⟩** (no default, initially 0pt)

Space of the given ⟨*length*⟩ is inserted horizontally right of the *raster*.

```
\begin{tcbitemize}[raster right skip=2cm,
  size=small,colframe=red!50!black,colback=red!10!white]
  \tcbitem One
  \tcbitem Two
  \tcbitem Three
  \tcbitem Four
\end{tcbitemize}
```

One | Two

Three | Four

N 2014-11-10 **/tcb/raster column skip=⟨*length*⟩** (no default, initially 2mm)

Space of the given ⟨*length*⟩ is inserted horizontally between the columns.

```
\begin{tcbitemize}[raster column skip=2cm,
  size=small,colframe=red!50!black,colback=red!10!white]
  \tcbitem One
  \tcbitem Two
  \tcbitem Three
  \tcbitem Four
\end{tcbitemize}
```

One | Two

Three | Four

N 2014-11-10 **/tcb/raster row skip=⟨*length*⟩** (no default, initially 2mm)

Space of the given ⟨*length*⟩ is inserted vertically between the rows.

```
\begin{tcbitemize}[raster row skip=0pt,
  size=small,colframe=red!50!black,colback=red!10!white]
  \tcbitem One
  \tcbitem Two
  \tcbitem Three
  \tcbitem Four
\end{tcbitemize}
```

One | Two

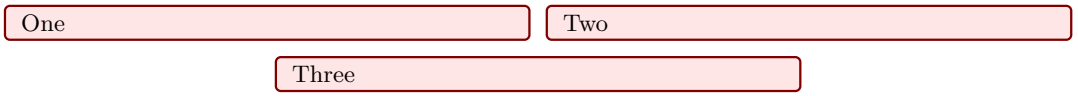
Three | Four

N 2014-11-10 **/tcb/raster halign=⟨*alignment*⟩** (no default, initially **left**)

Defines the horizontal alignment for the boxes of the rows of a *raster*, if these rows are not completely filled (mainly: the last one).
Feasible values for ⟨*alignment*⟩ are:

- **left**: align to the left side,
- **center**: align to the center,
- **right**: align to the right side.

```
\begin{tcbitemize}[raster halign=center,
  size=small,colframe=red!50!black,colback=red!10!white]
  \tcbitem One
  \tcbitem Two
  \tcbitem Three
\end{tcbitemize}
```



N 2014-11-10 **/tcb/raster valign=⟨*alignment*⟩** (no default, initially **center**)

Defines the vertical alignment for the boxes of a row, if the boxes do not have equal height. This sets the /tcb/box align → P. 77 option.
Feasible values for ⟨*alignment*⟩ are:

- **top**: align to the top side,
- **center**: align to the center,
- **bottom**: align to the bottom side.

```
\begin{tcbitemize}[raster valign=top, raster columns=3,
  size=small,colframe=red!50!black,colback=red!10!white]
  \tcbitem \Huge One
  \tcbitem \Large Two
  \tcbitem Three
\end{tcbitemize}
\begin{tcbitemize}[raster valign=center, raster columns=3,
  size=small,colframe=blue!50!black,colback=blue!10!white]
  \tcbitem \Huge One
  \tcbitem \Large Two
  \tcbitem Three
\end{tcbitemize}
\begin{tcbitemize}[raster valign=bottom, raster columns=3,
  size=small,colframe=green!50!black,colback=green!10!white]
  \tcbitem \Huge One
  \tcbitem \Large Two
  \tcbitem Three
\end{tcbitemize}
```

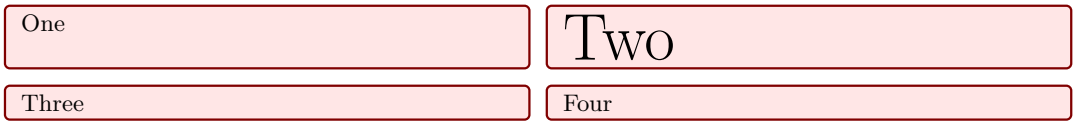
N 2014-11-10 **/tcb/raster equal height=⟨*type*⟩** (default **all**, initially **none**)

Puts the enclosed boxes into a common /tcb/equal height group → P. 59. The ⟨*id*⟩ of the equal height group is chosen automatically, but it may be set manually by /tcb/raster equal height group. Also see /tcb/minimum for current equal height group → P. 60.
Feasible values for ⟨*type*⟩ are:

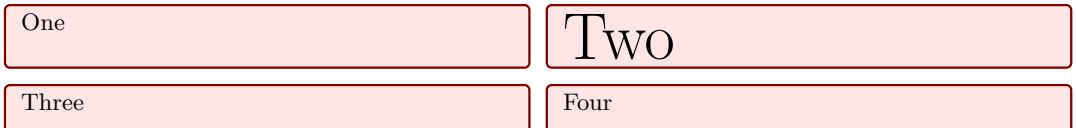
- **none**: no equal height setting,
- **rows**: all boxes in a row are set to equal height,
- **all**: all boxes in the raster are set to equal height.

Note that you have to compile twice to see changes.

```
\begin{tcbitemize}[raster equal height=rows,
  size=small,colframe=red!50!black,colback=red!10!white]
  \tcbitem One
  \tcbitem \Huge Two
  \tcbitem Three
  \tcbitem Four
\end{tcbitemize}
```



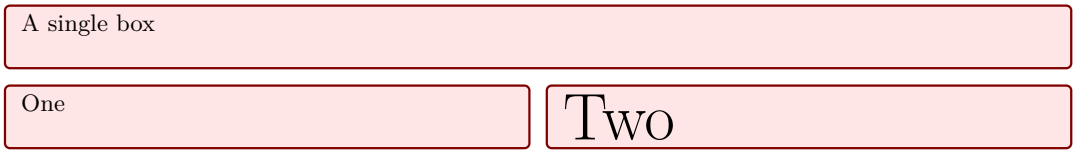

```
\begin{tcbitemize}[raster equal height,
  size=small,colframe=red!50!black,colback=red!10!white]
  \tcbitem One
  \tcbitem \Huge Two
  \tcbitem Three
  \tcbitem Four
\end{tcbitemize}
```



N 2014-11-10 **/tcb/raster equal height group=⟨*id*⟩** (no default)

Overwrites the automatically chosen id with the given ⟨*id*⟩. If this is used to share a common height between the *raster* and another raster or box, the /tcb/raster equal height option should be set to **all**.

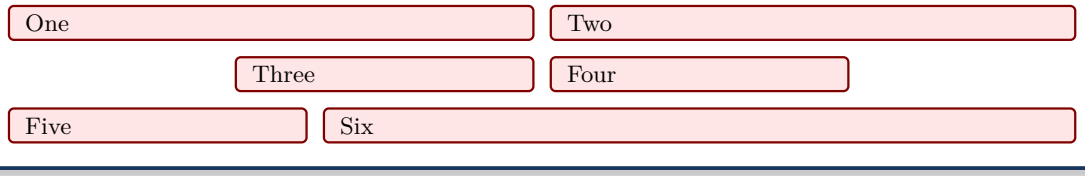
```
\tcbset{size=small,colframe=red!50!black,colback=red!10!white}
\begin{tcolorbox}[equal height group=raster-manual-id]
  A single box
\end{tcolorbox}
\begin{tcbitemize}[raster equal height,raster equal height group=raster-manual-id]
  \tcbitem One
  \tcbitem \Huge Two
\end{tcbitemize}
```



N 2014-11-10 **/tcb/raster force size=true|false** (default **true**, initially **true**)

Enforces the raster size computations onto the enclosed boxes. If set to **false**, individual settings can be used (for the better or worse).

```
\begin{tcbitemize}[raster force size=false, raster halign=center,
  size=small,colframe=red!50!black,colback=red!10!white]
  \tcbitem One
  \tcbitem Two
  \tcbitem[add to width=-3cm] Three
  \tcbitem[add to width=-3cm] Four
  \tcbitem[add to width=-3cm] Five
  \tcbitem[add to width=3cm] Six
\end{tcbitemize}
```



N 2014-11-10 **/tcb/raster reset** (no value)

Sets all raster settings back to their default values. Note that /tcb/reset → P. 98 does not execute this option. Style settings like /tcb/raster odd column etc. are not touched by /tcb/raster reset.

14.4 Adding Styles for Specific Boxes

The following styles can be defined to address certain boxes inside a *raster*. Note that such style definitions are not removed by /tcb/reset → P. 98 or /tcb/raster reset. The style definitions are used in the order given below.

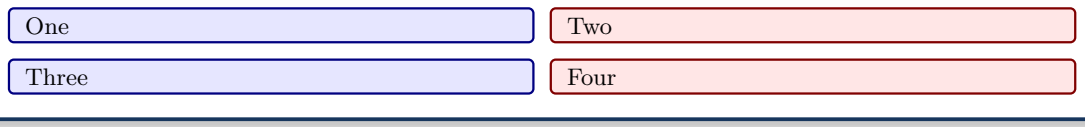
N 2014-11-24 **/tcb/raster every box** (style)

This style is used for every box.

N 2014-11-10 **/tcb/raster odd column** (style)

This style is used for every box in an odd column.

```
\begin{tcbitemize}[size=small,colframe=red!50!black,colback=red!10!white,
  raster odd column/.style={colframe=blue!50!black,colback=blue!10!white}]
  \tcbitem One
  \tcbitem Two
  \tcbitem Three
  \tcbitem Four
\end{tcbitemize}
```



N 2014-11-10 **/tcb/raster even column** (style)

This style is used for every box in an even column.

N 2014-11-10 **/tcb/raster column n** (style)

This style is used for every box in the **n**-th column. **n** has to be replaced by a number.

N 2014-11-10 **/tcb/raster odd row** (style)

This style is used for every box in an odd row.

N 2014-11-10 **/tcb/raster even row** (style)

This style is used for every box in an even row.

N 2014-11-10 **/tcb/raster row m** (style)

This style is used for every box in the m-th row. m has to be replaced by a number.

```
\begin{tcbitemize}[size=small,colframe=red!50!black,colback=red!10!white,
  raster row 2/.style={colframe=blue!50!black,colback=blue!10!white}]
  \tcbitem One
  \tcbitem Two
  \tcbitem Three
  \tcbitem Four
\end{tcbitemize}
```

One | Two

Three | Four

N 2014-11-10 **/tcb/raster odd number** (style)

This style is used for every box with an odd number.

N 2014-11-10 **/tcb/raster even number** (style)

This style is used for every box with an even number.

```
\begin{tcbitemize}[size=small,colframe=red!50!black,colback=red!10!white,
  raster columns=3,
  raster even number/.style={colframe=blue!50!black,colback=blue!10!white}]
  \tcbitem One
  \tcbitem Two
  \tcbitem Three
  \tcbitem Four
  \tcbitem Five
  \tcbitem Six
\end{tcbitemize}
```

One | Two | Three

Four | Five | Six

N 2014-11-10 **/tcb/raster row m column n** (style)

This style is used for the box in the m-th row and n-th column. m and n have to be replaced by numbers.

N 2014-11-10 **/tcb/raster number n** (style)

This style is used for the box with number n. n has to be replaced by a number.

```
\begin{tcbitemize}[size=small,colframe=red!50!black,colback=red!10!white,
  raster number 4/.style={colframe=blue!50!black,colback=blue!10!white}]
  \tcbitem One
  \tcbitem Two
  \tcbitem Three
  \tcbitem Four
\end{tcbitemize}
```

One | Two

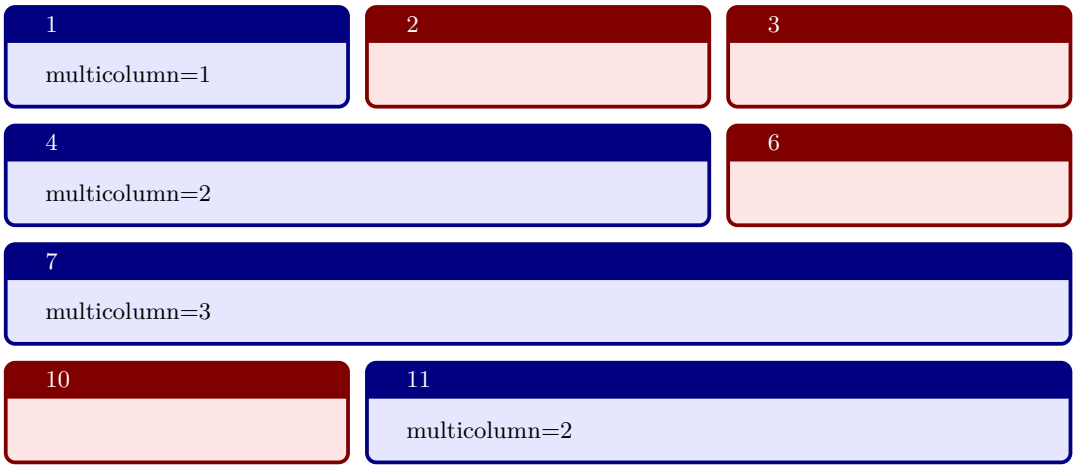
Three | Four

14.5 Combining Columns or Rows

N 2016-02-19 **/tcb/raster multicolumn=**⟨*number*⟩ (no default, initially unset)

This option has to be set inside the option list of a `tcolorbox` → P. 12 inside a `tcbraster` → P. 267 or inside `\tcbitem` → P. 268 inside `tcbitemize` → P. 268. It merges the given ⟨*number*⟩ of boxes into one single box on the same line. The resulting box gets the `\thetcbrasternum` of the first box. If there are not enough boxes available on the current line, this option is ignored and a warning is given.

```
\begin{tcbitemize}[raster equal height=rows,raster columns=3,
  title=\thetcbrasternum,colframe=red!50!black,colback=red!10!white]
\tcbitem[colframe=blue!50!black,colback=blue!10!white,raster multicolumn=1]
  multicolumn=1
\tcbitem
\tcbitem
\tcbitem[colframe=blue!50!black,colback=blue!10!white,raster multicolumn=2]
  multicolumn=2
\tcbitem
\tcbitem[colframe=blue!50!black,colback=blue!10!white,raster multicolumn=3]
  multicolumn=3
\tcbitem
\tcbitem[colframe=blue!50!black,colback=blue!10!white,raster multicolumn=2]
  multicolumn=2
\end{tcbitemize}
```

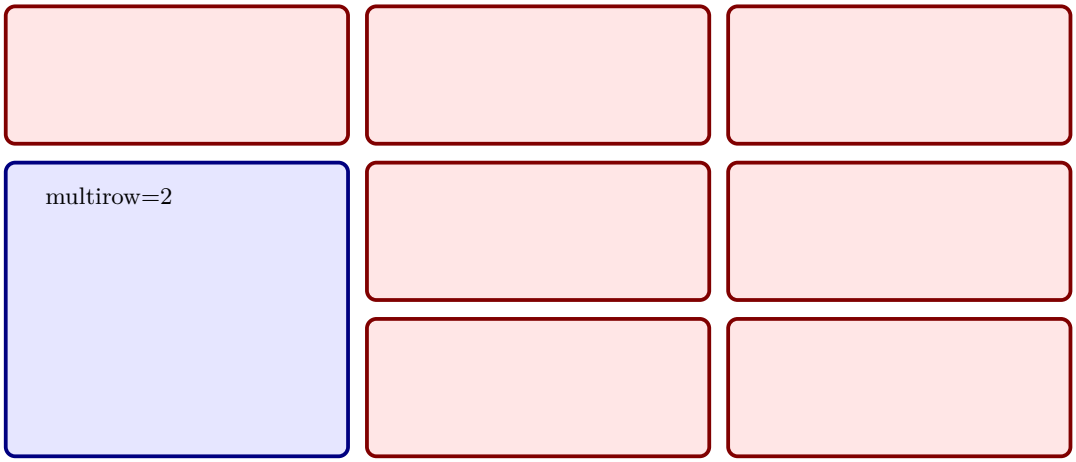


N 2016-02-19 **/tcb/raster multirow=**⟨*number*⟩ (no default, initially unset)

This option has to be set inside the option list of a tcolorbox → P. 12 inside a tcbraster → P. 267 or inside \tcbitem → P. 268 inside tcbitemize → P. 268. This option not really merges boxes, but simply sizes the current box to fit the space of ⟨*number*⟩ rows.

! /tcb/raster multirow needs /tcb/raster height → P. 272 to be set. How to achieve a similar result for boxes without fixed /tcb/raster height → P. 272 is shown afterwards.

```
\begin{tcbitemize}[raster rows=3,raster columns=3,raster height=6cm,
  raster every box/.style={colframe=red!50!black,colback=red!10!white}]
\tcbitem
\tcbitem
\tcbitem
\tcbitem[colframe=blue!50!black,colback=blue!10!white,raster multirow=2]
  multirow=2
\tcbitem[raster multicolumn=2,raster multirow=2,blankest]
  \begin{tcbitemize}[raster rows=2,raster columns=2,raster height=\tcbtextheight]
  \tcbitem
  \tcbitem
  \tcbitem
  \tcbitem
  \end{tcbitemize}
\end{tcbitemize}
```



For rasters without fixed /tcb/raster height →P. 272, /tcb/raster multirow →P. 279 cannot be used. Note that \tcbtextheight →P. 138 also cannot be used like in the previous example.
But, with combination of /tcb/raster equal height →P. 275 and /tcb/space to →P. 57, a similar effect can be created:

```
\begin{tcbitemize}[raster columns=3,raster equal height=rows,
  raster every box/.style={colframe=red!50!black,colback=red!10!white}]
\tcbitem
\tcbitem
\tcbitem
\tcbitem[colframe=blue!50!black,colback=blue!10!white]
 \lipsum[2]
\tcbitem[raster multicolumn=2,blankest,space to=\myspace]
  \begin{tcbitemize}[raster columns=2]
  \tcbitem This is a box of the inner raster.
  \tcbitem
  \tcbitem[height=\myspace]
  \tcbitem[height=\myspace]
  \end{tcbitemize}
\end{tcbitemize}
```

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

This is a box of the inner raster.

14.6 Rasters inside Rasters

A *raster* inside a *raster* cannot be used directly, because a *raster* can only contain a *tcolorbox* or something derived from a *tcolorbox*. So, a *raster* can be put inside a *tcolorbox* inside a *raster*.

Some examples for such constructions can be found at `tcboxedraster` → P. 269, `/tcb/raster multicolumn` → P. 278, `/tcb/raster multirow` → P. 279.

14.6.1 Raster Setup

The intermediating `tcolorbox` → P. 12 can be made invisible by using `/tcb/blankest` → P. 228.

```
\begin{tcbraster}[raster equal height=rows,
  raster every box/.style={colframe=red!50!black,colback=red!10!white}]
  \begin{tcolorbox}[blankest]
    \begin{tcbraster}[raster columns=1]
      \begin{tcolorbox}One\end{tcolorbox}
      \begin{tcolorbox}Two\end{tcolorbox}
    \end{tcbraster}
  \end{tcolorbox}
  \begin{tcolorbox}raster+tcolorbox+raster\end{tcolorbox}
\end{tcbraster}
```

One

Two

raster+tcolorbox+raster

```
\begin{tcbraster}[raster equal height=rows,
  raster every box/.style={colframe=red!50!black,colback=red!10!white}]
  \begin{tcboxedraster}[raster columns=1]{blankest}
    \begin{tcolorbox}One\end{tcolorbox}
    \begin{tcolorbox}Two\end{tcolorbox}
  \end{tcboxedraster}
  \begin{tcolorbox}raster+tcboxedraster\end{tcolorbox}
\end{tcbraster}
```

One

Two

raster+tcboxedraster

```
\begin{tcbitemize}[raster equal height=rows,
  raster every box/.style={colframe=red!50!black,colback=red!10!white}]
  \tcbitem[blankest]
    \begin{tcbitemize}[raster columns=1]
      \tcbitem One
      \tcbitem Two
    \end{tcbitemize}
  \tcbitem tcbitemize+tcbitem+tcbitemize
\end{tcbitemize}
```

One

Two

tcbitemize+tcbitem+tcbitemize

14.6.2 Placing Spaces

If the heights of boxes inside staggered rasters should be matched, the space has to be distributed accordingly.

- For fixed height boxes/rasters using `/tcb/raster height` → P. 272, the height of boxes is available by `\tcbtextheight` → P. 138. This can be used to size deeper layered boxes/rasters.
- For boxes/rasters layed out using `/tcb/raster equal height` → P. 275, space can be distributed by `/tcb/space to` → P. 57. It can take several compilations until all spaces are distributed correctly.

```
\begin{tcbitemize}[raster rows=2,raster height=6cm,
  raster every box/.style={colframe=red!50!black,colback=red!10!white}]
  \tcbitem[blankest]
    \begin{tcbitemize}[raster columns=1,raster rows=2,raster height=\tcbtextheight]
      \tcbitem One
      \tcbitem Two
    \end{tcbitemize}
  \tcbitem This is a fixed height box.
  \tcbitem Three
  \tcbitem Four
\end{tcbitemize}
```

One

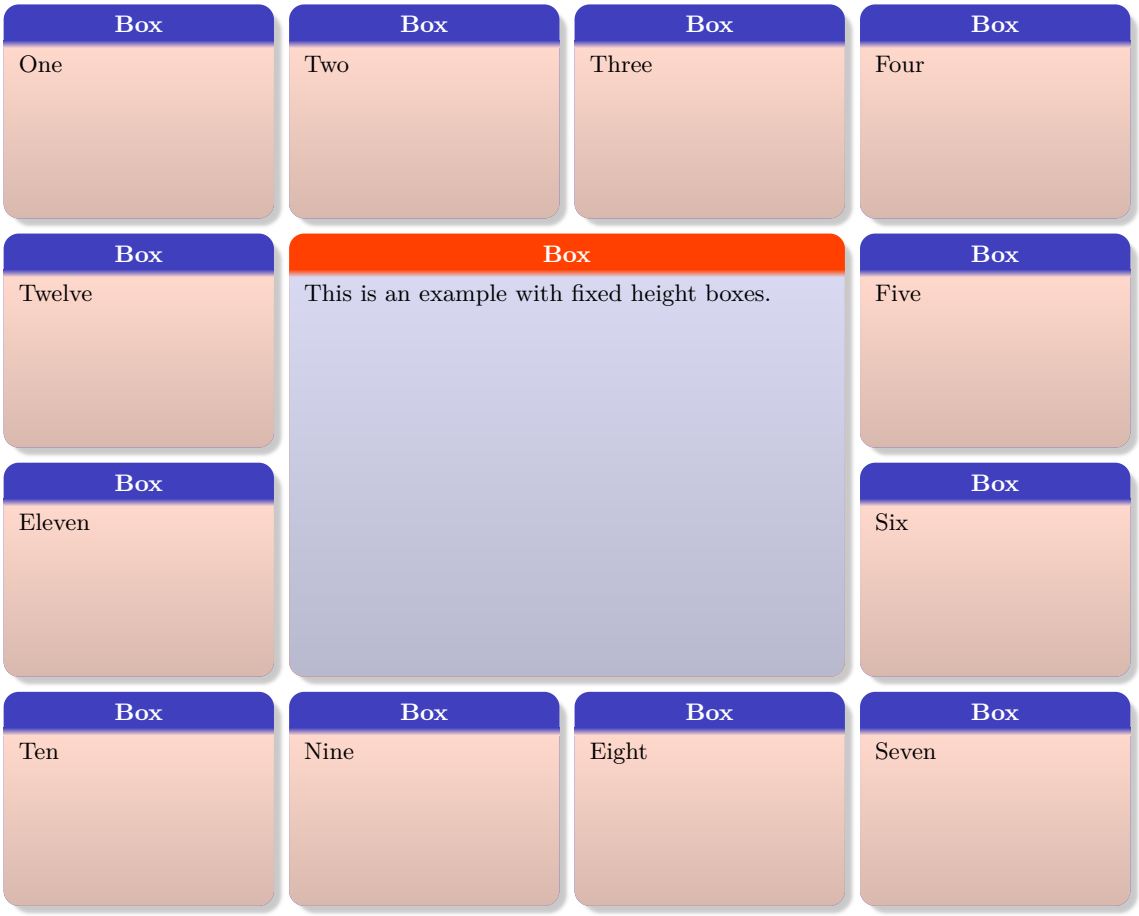
Two

This is a fixed height box.

Three

Four

```
\begin{tcbitemize}[raster columns=4,raster rows=4,raster height=0.8\linewidth,
  raster every box/.style={size=small,beamer,
    colframe=blue!75!yellow,colback=red!75!yellow!20,
    center title,title=Box}]
  \tcbitem One
  \tcbitem Two
  \tcbitem Three
  \tcbitem Four
  \tcbitem[raster multirow=2,blankest]
    \begin{tcbitemize}[raster columns=1,raster rows=2,raster height=\tcbtextheight]
      \tcbitem Twelve
      \tcbitem Eleven
    \end{tcbitemize}
  \tcbitem[raster multirow=2,raster multicolumn=2,
      colframe=red!75!yellow,colback=blue!75!yellow!20]
    This is an example with fixed height boxes.
  \tcbitem[raster multirow=2,blankest]
    \begin{tcbitemize}[raster columns=1,raster rows=2,raster height=\tcbtextheight]
      \tcbitem Five
      \tcbitem Six
    \end{tcbitemize}
  \tcbitem Ten
  \tcbitem Nine
  \tcbitem Eight
  \tcbitem Seven
\end{tcbitemize}
```



```
\begin{tcbitemize}[raster equal height=rows,
  raster every box/.style={colframe=red!50!black,colback=red!10!white}]
  \tcbitem[blankest,space to=\myspace]
    \begin{tcbitemize}[raster columns=1]
      \tcbitem One
      \tcbitem[add to natural height=\myspace]
        This box will adapt its height.
    \end{tcbitemize}
  \tcbitem This is a flexible height box.
  \tcbitem \lipsum[4]
  \tcbitem[blankest,space to=\myspace]
    \begin{tcbitemize}[raster columns=1]
      \tcbitem One
      \tcbitem[add to natural height=\myspace]
        This box will adapt its height.
    \end{tcbitemize}
\end{tcbitemize}
```

One

This box will adapt its height.

This is a flexible height box.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

One

This box will adapt its height.

```
\begin{tcbitemize}[raster equal height=rows,
  raster every box/.style={colframe=red!50!black,colback=red!10!white}]
  \tcbitem[blankest,space to=\myspace]
    \begin{tcbitemize}[raster columns=1]
      \tcbitem One
      \tcbitem[add to natural height=\myspace]
        This box will adapt its height.
      \tcbitem \lipsum[4]
    \end{tcbitemize}
  \tcbitem[blankest,space to=\myspace]
    \begin{tcbitemize}[raster columns=1]
      \tcbitem[blankest]\includegraphics[width=\linewidth]{goldshade.png}
      \tcbitem[add to natural height=\myspace]
        This box will adapt its height.
    \end{tcbitemize}
\end{tcbitemize}
```

One

This box will adapt its height.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

This box will adapt its height.

15 Libraries LIB listings, LIB listingsutf8, and LIB minted

15.1 Loading the Libraries

In contrast to other `tcolorbox` libraries, the libraries LIB listings, LIB listingsutf8, and LIB minted are concurrent in the sense that they all do the same thing, i. e. displaying listings with or without typesetting the listing in LaTeX parallel. The difference is the underlying LaTeX package which does the core job for displaying a listing. So, typically, you need just *one* of these libraries. If you do not have a clue, which one of them you should use, you should take LIB listingsutf8.

! The order in which the libraries are included influences the default settings and the `/tcb/reset` → P. 98 behavior. The settings of a later loaded library overwrite the settings of a previous loaded library. A library is never loaded twice.

15.1.1 Loading LIB listings

This library uses the package `listings` [6] to typeset listings. It is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{listings}
```

This also loads the package `listings` [6].

The `/tcb/listing engine` → P. 298 is set to `listings` by the library. To reactivate this setting, if overwritten by other libraries, use

```
\tcbset{listing engine=listings}
```

15.1.2 Loading LIB listingsutf8

To extend `listings` for UTF-8 encoded sources, you can use the support from the package `listingsutf8` [10] by loading the library variant LIB listingsutf8.

```
\tcbuselibrary{listingsutf8}
\tcbset{listing utf8=latin1}% optional; 'latin1' is the default.
```

This also loads the library LIB listings and the packages `listings` [6] and `listingsutf8` [10].

The `/tcb/listing engine` → P. 298 is set to `listings` by the library. To reactivate this setting, if overwritten by other libraries, use

```
\tcbset{listing engine=listings}
```

15.1.3 Loading LIB minted

This library uses the package `minted` [11] to typeset listings. It is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{minted}
```

This also loads the package `minted` [11].

> **!** The `minted` package uses the external tool `Pygments` [13] to apply syntax highlighting. It has to be installed and set up, before the library can be used, see [11] and [13]. The `tcolorbox` library LIB minted does not work, if the package `minted` [11] does not work.

The `/tcb/listing engine` → P. 298 is set to `minted` by the library. To reactivate this setting, if overwritten by other libraries, use

```
\tcbset{listing engine=minted}
```

15.2 Common Macros of the Libraries

`\begin{tcblisting}{`⟨*options*⟩`}`
⟨*environment content*⟩
`\end{tcblisting}`

Creates a colored box based on a `tcolorbox` → P. 12. Controlled by the given ⟨*options*⟩, the environment content is typeset normally and/or as a listing. Furthermore, the ⟨*options*⟩ control appearance and functions of the `tcolorbox`. By default, the listing is interpreted as a LaTeX listing.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black}
This is a \LaTeX\ example which displays the text as source code
and in compiled form.
\end{tcblisting}
```

```
This is a \LaTeX\ example which displays the text as source code
and in compiled form.
```

This is a LaTeX example which displays the text as source code and in compiled form.

```
% \tcbuselibrary{listings} /or/ \tcbuselibrary{listingsutf8}
\begin{tcblisting}{colback=yellow!5,colframe=yellow!50!black,listing only,
    title=This is source code in another language (XML), fonttitle=\bfseries,
  listing options={language=XML,columns=fullflexible,keywordstyle=\color{red}}}
<?xml version="1.0"?>
<project name="Package tcolorbox" default="documentation" basedir=".">
  <description>
    Apache Ant build file (http://ant.apache.org/)
  </description>
</project>
\end{tcblisting}
```

This is source code in another language (XML)

```
<?xml version="1.0"?>
<project name="Package␣tcolorbox" default="documentation" basedir=".">
  <description>
    Apache Ant build file (http://ant.apache.org/)
  </description>
</project>
```

```
% \tcbuselibrary{minted}
\begin{tcblisting}{colback=yellow!5,colframe=yellow!50!black,listing only,
    title=This is source code in another language (XML), fonttitle=\bfseries,
  listing engine=minted,minted language=xml}
<?xml version="1.0"?>
<project name="Package tcolorbox" default="documentation" basedir=".">
  <description>
    Apache Ant build file (http://ant.apache.org/)
  </description>
</project>
\end{tcblisting}
```

This is source code in another language (XML)

```
<?xml version="1.0"?>
<project name="Package tcolorbox" default="documentation" basedir=".">
  <description>
    Apache Ant build file (http://ant.apache.org/)
  </description>
</project>
```

```
% This box is as wide as needed (listing only !!)
% \tcbuselibrary{skins}
\begin{tcblisting}{colback=green!5!white,colframe=green!50!black,listing only,
  hbox,enhanced,drop fuzzy shadow,before=\begin{center},after=\end{center}}
\begin{tikzpicture}
\fill[red] (0,0) rectangle (1,1);
\end{tikzpicture}
\end{tcblisting}
```

```
\begin{tikzpicture}
\fill[red] (0,0) rectangle (1,1);
\end{tikzpicture}
```

\begin{tcboutputlisting}
⟨*environment content*⟩
\end{tcboutputlisting}

Saves the environment content to a file which is named by the key value of `listing file`. Later, this file can be loaded by `\tcbinputlisting` or `\tcbuselistingtext` or `\tcbuselistinglisting`.

```
\begin{tcboutputlisting}
This \textbf{text} is written to a standardized file for later usage.
\end{tcboutputlisting}
```

\tcbinputlisting{⟨*options*⟩}

Creates a colored boxed based on a `tcolorbox`. The text content is read from a file named by the key value of `listing file`. Apart from that, the function is equal to that of `tcblisting` → P. 287.

```
\tcbinputlisting{colback=red!5!white,colframe=red!75!black,text only}
\tcbinputlisting{colback=green!5,colframe=green!75!black,listing only}
```

```
\begin{tikzpicture}
\fill[red] (0,0) rectangle (1,1);
\end{tikzpicture}
```

\tcbuselistingtext

Loads text from a file named by the key value of `listing file`.

```
\tcbuselistingtext
```

\tcbuselistinglisting

Typesets text as listing from a file named by the key value of `listing file`.

```
\tcbuselistinglisting
```

```
\begin{tikzpicture}
\fill[red] (0,0) rectangle (1,1);
\end{tikzpicture}
```

\tcbusetemplisting

Typesets text as listing from a temporary file which was written by `tcbwritetemp` → P. 118.

See Section 21.4 on page 410 and Section 21.5 on page 412 for more elaborate methods to create new environments and commands.

If a new sort of `tcblisting` environments should be created with one optional argument only, one is highly recommended to use `\DeclareTCBListing` → P. 410 or `\NewTCBListing` → P. 410 instead of `\newtcblisting` to avoid content scanning problems.

`\newtcblisting[⟨init options⟩]{⟨name⟩}[⟨number⟩][⟨default⟩]{⟨options⟩}`

Creates a new environment ⟨*name*⟩ based on `tcblisting` → P. 287. Basically, `\newtcblisting` operates like `\newenvironment`. This means, the new environment ⟨*name*⟩ optionally takes ⟨*number*⟩ arguments, where ⟨*default*⟩ is the default value for the optional first argument. The ⟨*options*⟩ are given to the underlying `tcblisting`. Note that `/tcb/savedelimiter` → P. 26 is set to the given ⟨*name*⟩ automatically. The ⟨*init options*⟩ allow setting up automatic numbering, see Section 5 from page 100.

```
\newtcblisting{mybox}{%
  colback=red!5!white,
  colframe=red!75!black}

\begin{mybox}
This is my \LaTeX\ box.
\end{mybox}
```

This is my \LaTeX\ box.

This is my LaTeX box.

```
\newtcblisting{mybox}[1]{%
  colback=red!5!white,
  colframe=red!75!black,
  fonttitle=\bfseries,
  title=#1}

\begin{mybox}{Listing Box}
This is my \LaTeX\ box.
\end{mybox}
```

Listing Box

This is my \LaTeX\ box.

This is my LaTeX box.

```
\newtcblisting{mybox}[2][]{%
  colback=red!5!white,
  colframe=red!75!black,
  fonttitle=\bfseries,
  title=#2,#1}

\begin{mybox}[listing only]
  {Listing Box}
This is my \LaTeX\ box.
\end{mybox}
\bigskip

\begin{mybox}[listing side text]
  {Listing Box}
This is my
\LaTeX\ box.
\end{mybox}
```

Listing Box

This is my \LaTeX\ box.

Listing Box

This is my \LaTeX\ box.

This is my LaTeX box.

Definition in the preamble:

```
\newtcblisting[auto counter]{mycbox}[1]{%
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  title=Listing \thetcbcounter: #1}
```

```
\begin{mycbox}{Listing Box}
This is my \LaTeX\ box.
\end{mycbox}
```

Listing 1: Listing Box

This is my \LaTeX\ box.

This is my LaTeX box.

`\renewtcblisting`[⟨*init options*⟩]{⟨*name*⟩}[⟨*number*⟩][⟨*default*⟩]{⟨*options*⟩}

Operates like `\newtcblisting` → P. 290, but based on `\renewenvironment` instead of `\newenvironment`. An existing environment is redefined.

`\newtcbinputlisting`[⟨*init options*⟩]{\⟨*name*⟩}[⟨*number*⟩][⟨*default*⟩]{⟨*options*⟩}

Creates a new macro \⟨*name*⟩ based on `\tcbinputlisting` → P. 289. Basically, `\newtcbinputlisting` operates like `\newcommand`. The new macro \⟨*name*⟩ optionally takes ⟨*number*⟩ arguments, where ⟨*default*⟩ is the default value for the optional first argument. The ⟨*options*⟩ are given to the underlying `tcbinputlisting`. The ⟨*init options*⟩ allow setting up automatic numbering, see Section 5 from page 100.

```
\newtcbinputlisting[use counter from=mycbox]{\mylisting}[2][]{%
  listing file={#2},
  title=Listing (\thetcbcounter) of \texttt{#2},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  listing only,breakable,#1}

\mylisting[before upper=\textit{This is the included file content:}]
          {\jobname.tcbtemp}
```

Listing (2) of `tcolorbox.tcbtemp`

This is the included file content:

```
\newtcbinputlisting[use counter from=mycbox]{\mylisting}[2][]{%
  listing file={#2},
  title=Listing (\thetcbcounter) of \texttt{#2},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  listing only,breakable,#1}

\mylisting[before upper=\textit{This is the included file content:}]
          {\jobname.tcbtemp}
```

```
\newtcbinputlisting[use counter from=mycbox]{\mylisting}[2][]{%
  listing engine=minted,minted language=latex,minted style=colorful,
  listing file={#2},
  title=Listing (\thetcbcounter) of \texttt{#2},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  listing only,breakable,#1}

\mylisting[before upper=\textit{This is the included file content:}]
          {\jobname.tcbtemp}
```

Listing (3) of `tcolorbox.tcbtemp`

This is the included file content:

```
\newtcbinputlisting[use counter from=mycbox]{\mylisting}[2][]{%
  listing engine=minted,minted language=latex,minted style=colorful,
  listing file={#2},
  title=Listing (\thetcbcounter) of \texttt{#2},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  listing only,breakable,#1}

\mylisting[before upper=\textit{This is the included file content:}]
          {\jobname.tcbtemp}
```

`\renewtcbinputlisting`[⟨*init options*⟩]{\⟨*name*⟩}[⟨*number*⟩][⟨*default*⟩]{⟨*options*⟩}

Operates like `\newtcbinputlisting`, but based on `\renewcommand` instead of `\newcommand`. An existing macro is redefined.

15.3 Option Keys of the LIB listings Library

/tcb/listing options=⟨*key list*⟩ (no default, initially `style=tcblatex`)

Sets the options from the package `listings` [6] which are used during typesetting of the listing. For LaTeX listings, there is a predefined `listings` style named `tcblatex` which can be used.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!25,left=6mm,
listing options={style=tcblatex,numbers=left,numberstyle=\tiny\color{red!75!black}}}
This is a \LaTeX\ example which displays the text as source code
and in compiled form. Additionally, we use line numbers here.
\end{tcblisting}
```

```
1 This is a \LaTeX\ example which displays the text as source code
2 and in compiled form. Additionally, we use line numbers here.
```

This is a LaTeX example which displays the text as source code and in compiled form. Additionally, we use line numbers here.

/tcb/no listing options (no value, initially unset)

Abbreviation for `listing options={}`. This removes all options for the `listings` package. This includes the `tcblisting` standard style `tcblatex` and the encoding presets. Use this option, if you want to set the `listings` options outside of `tcblisting`, e. g. globally in the preamble.

```
\begin{tcblisting}{no listing options}
All \textit{listings} options removed.
\end{tcblisting}
```

```
All \textit{listings} options removed.
```

All *listings* options removed.

/tcb/listing style=⟨*style*⟩ (no default, initially `tcblatex`)

Abbreviation for `listing options={style=...}`. This key sets a ⟨*style*⟩ for the `listings` package, see [6]. For LaTeX, there is a predefined style named `tcblatex`.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,
listing style=tcblatex}
Here, we use the predefined style.
\end{tcblisting}
```

```
Here, we use the predefined style.
```

Here, we use the predefined style.

`/tcb/listing inputencoding=`⟨*encoding*⟩ (no default, initially `\inputencodingname`)

Sets the input encoding value for the predefined listing style `tcblatex` and `tcbdocumentation` from the library LIB documentation. The initial value is derived from the package `inputenc` if used.

`/tcb/listing remove caption=true|false` (default `true`, initially `true`)

If set to `true`, some part of the caption building code of the `listings` package is silenced to prevent some unwanted interaction with the `hyperref` package resulting in additional vertical space. If set to `false`, the `listings` package code is kept unchanged. Note that listings outside `tcblisting` → P. 287 and `\tcbinputlisting` → P. 289 are always processed normally. Typically, a user is not expected to use this key at all.

`/tcb/every listing line=`⟨*text*⟩ (no default, initially unset/empty)

Inserts some ⟨*text*⟩ to the begin of every line of a listing. Note that this a hack of the `listings` package code. This may become unusable or superfluous in the future.

```
\newtcblisting{commandshell}{colback=black,colupper=white,colframe=yellow!75!black,
  listing only,listing options={style=tcblatex,language=sh},
  every listing line={\textcolor{red}{\small\ttfamily\bfseries root \$> }}}

\begin{commandshell}
ls -al
cd /usr/lib
\end{commandshell}
```

```
root $> ls -al
root $> cd /usr/lib
```

`/tcb/every listing line*=`⟨*text*⟩ (no default, initially unset/empty)

Identical to `/tcb/every listing line` plus additional enlargement of `/tcb/rightupper` → P. 40 by the width of ⟨*text*⟩. Therefore, this option has to be used after the geometry settings are done. This option is intended to be used in conjunction with `/tcb/hbox` → P. 90.

```
\newtcblisting{commandshell}{colback=black,colupper=white,colframe=yellow!75!black,
  listing only,listing options={style=tcblatex,language=sh},hbox,
  every listing line*={\textcolor{red}{\small\ttfamily\bfseries root \$> }}}

\begin{commandshell}
ls -al
cd /usr/lib
\end{commandshell}
```

```
root $> ls -al
root $> cd /usr/lib
```

See further options in Section 15.6 on page 298.

! For an combined example of using `\lstinline` inside a `tcolorbox`, see `\DeclareTotalTCBox` → P. 408.

15.4 Option Keys of the LIB listingsutf8 Library

The LIB listingsutf8 library is an extension of the LIB listings library, so all options from Section 15.3 on page 293 are applicable.

`/tcb/listing utf8`=⟨*one-byte-encoding*⟩ (style, no default, initially `latin1`)
Abbreviation for using `/tcb/listing inputencoding` → P. 294 together with UTF-8 support from the package `listingsutf8` [10]. This option is available only for the library variant LIB listingsutf8. The ⟨*one-byte-encoding*⟩ is one of the applicable encodings from [10], e. g. `latin1`.

See further options in Section 15.6 on page 298.

15.5 Option Keys of the LIB minted Library

/tcb/minted language=⟨*programming language*⟩ (no default, initially `latex`)

Sets a ⟨*programming language*⟩ known to `Pygments` [13].

```
\begin{tcblisting}{listing engine=minted,minted style=trac,
  minted language=java,
  colback=red!5!white,colframe=red!75!black,listing only}
public class HelloWorld {
  // A 'Hello World' in Java
  public static void main(String[] args) {
    System.out.println("Hello World!");
  }
}
\end{tcblisting}
```

```
public class HelloWorld {
  // A 'Hello World' in Java
  public static void main(String[] args) {
    System.out.println("Hello World!");
  }
}
```

/tcb/minted options=⟨*key list*⟩ (no default, initially `tabsize=2,fontsize=\small`)

Sets the options from the package `minted` [11] which are used during typesetting of the listing.

```
% \tcbuselibrary{skins}
\newtcblisting{myjava}{listing engine=minted,minted style=colorful,
  minted language=java,minted options={fontsize=\small,linenos,numbersep=3mm},
  colback=blue!5!white,colframe=blue!75!black,listing only,
  left=5mm,enhanced,
  overlay={\begin{tcbclipinterior}\fill[red!20!blue!20!white] (frame.south west)
    rectangle ([xshift=5mm]frame.north west);\end{tcbclipinterior}}}

\begin{myjava}
public class HelloWorld {
  // A 'Hello World' in Java
  public static void main(String[] args) {
    System.out.println("Hello World!");
  }
}
\end{myjava}
```

```
1 public class HelloWorld {
2   // A 'Hello World' in Java
3   public static void main(String[] args) {
4     System.out.println("Hello World!");
5   }
6 }
```

`/tcb/minted style=`⟨*style*⟩ (no default, initially unset)

Sets a ⟨*style*⟩ known to `Pygments` [13]. This is independent from `/tcb/minted options` → P. 296. Note that styles are always applied globally; all following examples will be set in the given ⟨*style*⟩ until a new style is set. Also note that setting `\usemintedstyle{`⟨*style*⟩`}` only once per document is more economic, if all styles in a document are the same. For examples of different styles, see `/tcb/minted language` → P. 296 and `/tcb/minted options` → P. 296.

See further options in Section 15.6 on the following page.

15.6 Common Option Keys of all Libraries

For the ⟨*options*⟩ in `tcblisting` → P. 287 respectively `\tcbinputlisting` → P. 289 the following `pgf` keys can be applied. The key tree path `/tcb/` is not to be used inside these macros.

`/tcb/listing engine=`⟨*engine*⟩ (no default)

Sets the ⟨*engine*⟩ which typesets the listings. Feasible values are

- `listings`, if library LIB listings or LIB listingsutf8 is loaded.
- `minted`, if library LIB minted is loaded.

`/tcb/listing file=`⟨*file name*⟩ (no default, initially `\jobname.listing`)

Sets the ⟨*file name*⟩ of the file which is used to save listings.

`/tcb/listing and text` (no value, initially set)

Typesets the environment content as listing in the upper part and as compiled text in the lower part.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing and text}
This is a \LaTeX\ example.
\end{tcblisting}
```

```
This is a \LaTeX\ example.
```

This is a LaTeX example.

`/tcb/text and listing` (no value)

Typesets the environment content as compiled text in the upper part and as listing in the lower part.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,text and listing}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a LaTeX example.

```
This is a \LaTeX\ example.
```

`/tcb/listing only` (no value)

Typesets the environment content as listing.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing only}
This is a \LaTeX\ example.
\end{tcblisting}
```

```
This is a \LaTeX\ example.
```

/tcb/text only (no value)

Typesets the environment content as compiled text.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,text only}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a LaTeX example.

/tcb/comment=⟨*text*⟩ (no default, initially empty)

Records a comment with ⟨*text*⟩ as content. The comment is displayed e. g. in conjunction with /tcb/listing and comment → P. 302 and /tcb/comment and listing → P. 302.

```
\begin{tcblisting}{comment={This comment is really only a comment},
  colback=red!5!white,colframe=red!75!black}
This is a \textbf{tcolorbox}.
\end{tcblisting}
```

This is a \textbf{tcolorbox}.

This is a **tcolorbox**.

N 2014-11-17 **/tcb/comment only** (no value)

Typesets the environment content with the comment text.

```
\begin{tcblisting}{comment only,
  comment={This is a comment.},
  colback=red!5!white,colframe=red!75!black}
This is a \textbf{tcolorbox}.
\end{tcblisting}
```

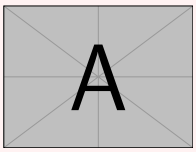
This is a comment.

/tcb/image comment={⟨*options*⟩**}{**⟨*filename*⟩**}** (style, no default, initially unset)

Uses an image denoted by ⟨*filename*⟩ as *comment* for the listing. The image is included by the standard **\includegraphics** macro with given ⟨*options*⟩.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing side comment,
  image comment={width=2.5cm}{example-image-a.pdf},center lower}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a \LaTeX\ example.



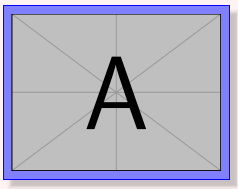
N 2014-11-14 **/tcb/tcbimage comment=⟨*filename*⟩** (style, no default, initially unset)

Uses an image denoted by ⟨*filename*⟩ as *comment* for the listing. The image is included by the \tcbincludegraphics → P. 240 macro. The inclusion can be customized by /tcb/comment style → P. 302.

! The library LIB skins is needed to apply this option.

```
% \tcbuselibrary{skins}
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing side comment,
  righthand width=3cm,lower separated=false,
  tcbimage comment={example-image-a.pdf},
  comment style={size=fbox,colframe=blue,colback=blue!50,sharp corners,
    drop fuzzy shadow}}
This is a \LaTeX\ example.
\end{tcblisting}
```

```
This is a \LaTeX\ example.
```



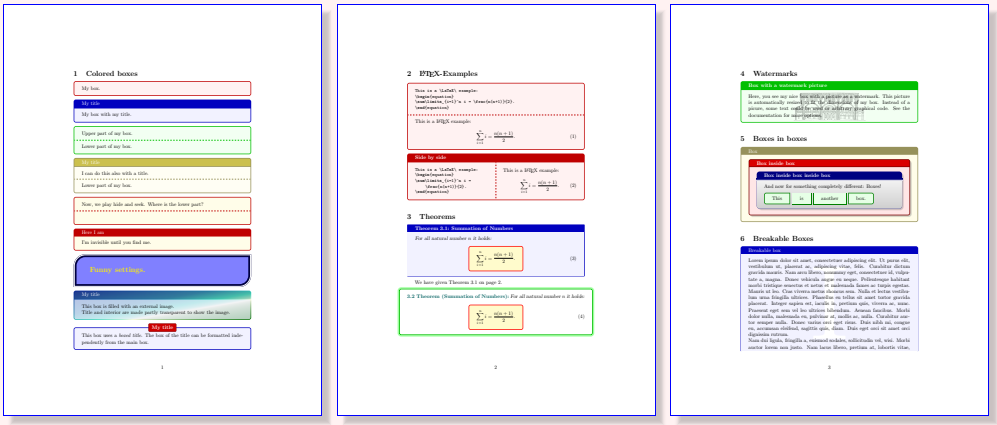
N 2014-11-14 **/tcb/pdf comment=⟨*filename*⟩** (style, default listing file, initially unset)

Uses a PDF file denoted by ⟨*filename*⟩ as *comment* for the listing. The image is included by \tcbincludepdf → P. 242 inside a tcbraster → P. 267. The inclusion can be customized by /tcb/comment style → P. 302.

! The libraries LIB skins and LIB raster are needed to apply this option.

```
% \tcbuselibrary{skins,raster}
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing and comment,
  righthand width=3cm,lower separated=false,middle=1mm,
  pdf comment={tcolorbox-example.pdf},
  comment style={raster columns=3,graphics pages={1,2,3},
    colframe=blue,drop fuzzy shadow}}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a **\LaTeX** example.



N 2014-11-14 **/tcb/pdf extension=⟨*extension*⟩** (no default, initially pdf)

Sets the PDF file name extension for /tcb/pdf comment → P. 300 to ⟨*extension*⟩. Note that ⟨*extension*⟩ always overwrites any actual extension given inside /tcb/pdf comment → P. 300.

N 2014-11-14 **/tcb/comment style=⟨*options*⟩** (no default, initially empty)

Sets the ⟨*options*⟩ for /tcb/tcbimage comment → P. 300 and /tcb/pdf comment → P. 300. These are tcolorbox options to customize the colored box drawn around the image(s), also image options encapsulated by /tcb/graphics options → P. 243, and tcbraster → P. 267 options for /tcb/pdf comment → P. 300.

/tcb/listing and comment (no value)

Typesets the environment content as listing in the upper part and a given comment in the lower part.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing and comment,
  comment={This is my comment. It may contain line breaks.\par
    It can even use the environment content
    \flqq\ignorespaces\tcbuselistingtext\unskip\frqq}}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a \LaTeX\ example.

This is my comment. It may contain line breaks.
It can even use the environment content «This is a LaTeX example.»

/tcb/comment and listing (no value)

Typesets a given comment in the upper part and the environment content as listing in the lower part.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,comment and listing,
comment={This is my comment.}}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is my comment.

This is a \LaTeX\ example.

/tcb/listing side text (no value)

Typesets the environment content side by side as listing in the left (upper) part and as compiled text in the right (lower) part.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing side text}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a \LaTeX\ example. | This is a LaTeX example.

/tcb/text side listing (no value)

Typesets the environment content side by side as compiled text in the left (upper) part and as listing in the right (lower) part.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,text side listing}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a LaTeX example. | This is a \LaTeX\ example.

/tcb/listing outside text (no value)

Typesets the environment content side by side as listing in a `tcolorbox` and as compiled text outside the box in the right part of the page. Nevertheless, the outside text is treated as *lower* part of the `tcolorbox` and can be formatted with all lower part options. The space partitioning is done with the side by side options from Section 6 on page 108.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing outside text}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a \LaTeX\ example.

This is a LaTeX example.

/tcb/text outside listing (no value)

Typesets the environment content side by side as listing in a `tcolorbox` and as compiled text outside the box in the left part of the page. Nevertheless, the outside text is treated as *lower* part of the `tcolorbox` and can be formatted with all lower part options. The space partitioning is done with the side by side options from Section 6 on page 108.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,text outside listing}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a LaTeX example.

This is a \LaTeX\ example.

/tcb/listing side comment (no value)

Typesets the environment content side by side as listing in the left (upper) part and a given comment in the right (lower) part.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing side comment,
  righthand width=1.5cm,image comment={width=1.5cm}{example-image-a.pdf}}
This is a \LaTeX\ example.
\end{tcblisting}
```

```
This is a \LaTeX\ example.
```

/tcb/comment side listing (no value)

Typesets the environment content side by side with a given comment in the left (upper) part and as listing in the right (lower) part.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,comment side listing,
  lefthand width=1.5cm,image comment={width=1.5cm}{example-image-a.pdf}}
This is a \LaTeX\ example.
\end{tcblisting}
```

```
This is a \LaTeX\ example.
```

/tcb/listing outside comment (no value)

Typesets the environment content side by side as listing in a `tcolorbox` and a given comment outside the box in the right part of the page. Nevertheless, the outside text is treated as *lower* part of the `tcolorbox` and can be formatted with all lower part options. The space partitioning is done with the side by side options from Section 6 on page 108.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing outside comment,
  righthand width=1.5cm,image comment={width=1.5cm}{example-image-a.pdf}}
This is a \LaTeX\ example.
\end{tcblisting}
```

```
This is a \LaTeX\ example.
```

/tcb/comment outside listing (no value)

Typesets the environment content side by side as listing in a `tcolorbox` and a given comment outside the box in the left part of the page. Nevertheless, the outside text is treated as *lower* part of the `tcolorbox` and can be formatted with all lower part options. The space partitioning is done with the side by side options from Section 6 on page 108.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,comment outside listing,
  lefthand width=1.5cm,image comment={width=1.5cm}{example-image-a.pdf}}
This is a \LaTeX\ example.
\end{tcblisting}
```

```
This is a \LaTeX\ example.
```

/tcb/listing above text (no value)

Typesets the environment content as listing in a tcolorbox and as compiled text outside and below the box. The outside text is treated as *lower* part of the tcolorbox and can be formatted with all lower part options. The distance between box and text is controlled by /tcb/middle → P. 42.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing above text}
This is a \LaTeX\ example.
\end{tcblisting}
```

```
This is a \LaTeX\ example.
```

This is a LaTeX example.

N 2014-11-07 /tcb/listing above* text (no value)

Widely equal to /tcb/listing above text, but the outside text is not formatted with the lower part options. Also, it is not put into a minipage and it may span several pages. The distance between box and text is controlled by /tcb/after → P. 76.

/tcb/text above listing (no value)

Typesets the environment content as listing in a tcolorbox and as compiled text outside and above the box. The outside text is treated as *lower* part of the tcolorbox and can be formatted with all lower part options. The distance between box and text is controlled by /tcb/middle → P. 42.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,text above listing}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a LaTeX example.

```
This is a \LaTeX\ example.
```

N 2014-11-07 /tcb/text above* listing (no value)

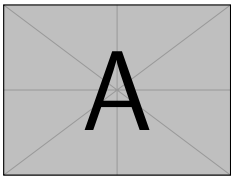
Widely equal to /tcb/text above listing, but the outside text is not formatted with the lower part options. Also, it is not put into a minipage and it may span several pages. The distance between box and text is controlled by /tcb/before → P. 76.

/tcb/listing above comment (no value)

Typesets the environment content as listing in a tcolorbox and a given comment outside and below the box. The outside text is treated as *lower* part of the tcolorbox and can be formatted with all lower part options. The distance between box and comment is controlled by /tcb/middle → P. 42.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing above comment,
  center lower,image comment={width=3cm}{example-image-a.pdf}}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a \LaTeX\ example.



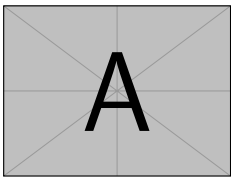
N 2014-11-07 /tcb/listing above* comment (no value)

Widely equal to /tcb/listing above comment, but the outside comment is not formatted with the lower part options. Also, it is not put into a minipage and it may span several pages. The distance between box and comment is controlled by /tcb/after → P. 76.

/tcb/comment above listing (no value)

Typesets the environment content as listing in a tcolorbox and a given comment outside and above the box. The outside text is treated as *lower* part of the tcolorbox and can be formatted with all lower part options. The distance between box and comment is controlled by /tcb/middle → P. 42.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,comment above listing,
  center lower,image comment={width=3cm}{example-image-a.pdf}}
This is a \LaTeX\ example.
\end{tcblisting}
```



This is a \LaTeX\ example.

N 2014-11-07 /tcb/comment above* listing (no value)

Widely equal to /tcb/comment above listing, but the outside comment is not formatted with the lower part options. Also, it is not put into a minipage and it may span several pages. The distance between box and comment is controlled by /tcb/before → P. 76.

15.7 Option Keys for Processing and Full Document Examples

A complete LaTeX document including `\documentclass`, `\begin{document}` and `\end{document}` cannot be processed directly by `tcolorbox`. It always has to be compiled separately. There are two methods supported by the package to process and display such a full document example:

- Prepare and compile the example document independent from your main document. The source file and the resulting PDF file can be included into the main document afterwards. This is the most economic way since the example document can be left untouched after the example is complete.
- The other possibility is to compile the example on the fly while the main document is compiled. This way has some charm, because the example can be edited inside the main document. But be aware that the compilation of the example is issued on every run of the main document. Also, there are fewer degrees of freedom how the example is compiled.

For both methods, the resulting example PDF file can be included as a `/tcb/pdf comment` → P. 300.

The following example shows how to apply the first method. There already is a file `tcolorbox-example.tex` and a PDF file `tcolorbox-example.pdf`. Both of them are input partly by the following:

```
% \tcbuselibrary{breakable,skins,raster}
\tcbinputlisting{
  enhanced jigsaw,breakable,pad at break*=2mm,height fixed for=first and middle,
  lower separated=false,
  leftlower=0pt,rightlower=0pt,middle=0pt,
  colframe=red!50!black,colback=yellow!10!white,
  listing and comment,
  listing file={tcolorbox-example},
  listing options=
    {style=tcblatex,texcsstyle=*\color{red!70!black},firstline=20,lastline=85},
  after upper={\par\bigskip\texttt{\ldots}\par},
  pdf comment,
  comment style={drop lifted shadow,graphics pages={1,...,4}},
}
```

```
\documentclass{article}
\usepackage{tikz,lipsum,lmodern}
\usepackage[most]{tcolorbox}

\begin{document}

%-------------------------------------------------------------
\section{Colored boxes}

\begin{tcolorbox}[colback=red!5!white,colframe=red!75!black]
  My box.
\end{tcolorbox}

\begin{tcolorbox}[colback=blue!5!white,colframe=blue!75!black,title=My title]
  My box with my title.
\end{tcolorbox}

\begin{tcolorbox}[colback=green!5!white,colframe=green!75!black]
  Upper part of my box.
  \tcblower
```

```
  Lower part of my box.
\end{tcolorbox}

\begin{tcolorbox}[colback=yellow!5!white,colframe=yellow!50!black,
  colbacktitle=yellow!75!black,title=My title]
  I can do this also with a title.
  \tcblower
  Lower part of my box.
\end{tcolorbox}

\begin{tcolorbox}[colback=yellow!10!white,colframe=red!75!black,lowerbox=invisible,
  savelowerto=\jobname_ex.tex]
  Now, we play hide and seek. Where is the lower part?
  \tcblower
  I'm invisible until you find me.
\end{tcolorbox}

\begin{tcolorbox}[colback=yellow!10!white,colframe=red!75!black,title=Here I am]
  \input{\jobname_ex.tex}
\end{tcolorbox}


\begin{tcolorbox}[enhanced,sharp corners=uphill,
    colback=blue!50!white,colframe=blue!25!black,coltext=yellow,
    fontupper=\Large\bfseries,arc=6mm,boxrule=2mm,boxsep=5mm,
    borderline={0.3mm}{0.3mm}{white}]
  Funny settings.
\end{tcolorbox}


\begin{tcolorbox}[enhanced,frame style image=blueshade.png,
  opacityback=0.75,opacitybacktitle=0.25,
  colback=blue!5!white,colframe=blue!75!black,
  title=My title]
  This box is filled with an external image.\par
  Title and interior are made partly transparent to show the image.
\end{tcolorbox}


\begin{tcolorbox}[enhanced,attach boxed title to top
    center={yshift=-3mm,yshifttext=-1mm},
  colback=blue!5!white,colframe=blue!75!black,colbacktitle=red!80!black,
  title=My title,fonttitle=\bfseries,
  boxed title style={size=small,colframe=red!50!black} ]
  This box uses a \textit{boxed title}. The box of the title can
  be formatted independently from the main box.
\end{tcolorbox}

...
```

1 Colored boxes

My box.

My title

My box with my title.

Upper part of my box.

Lower part of my box.

My title

I can do this also with a title.

Lower part of my box.

Now, we play hide and seek. Where is the lower part?

Here I am

I'm invisible until you find me.

Funny settings.

My title

This box is filled with an external image.
Title and interior are made partly transparent to show the image.

My title

This box uses a *boxed title*. The box of the title can be formatted independently from the main box.

2 LaTeX-Examples

```
This is a \LaTeX\ example:
\begin{equation}
\sum\limits_{i=1}^n i = \frac{n(n+1)}{2}.
\end{equation}
```

This is a LaTeX example:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}. \tag{1}$$

Side by side

```
This is a \LaTeX\ example:
\begin{equation}
\sum\limits_{i=1}^n i =
    \frac{n(n+1)}{2}.
\end{equation}
```

This is a LaTeX example:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}. \tag{2}$$

3 Theorems

Theorem 3.1: Summation of Numbers

For all natural number n it holds:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}. \tag{3}$$

We have given Theorem 3.1 on page 2.

3.2 Theorem (Summation of Numbers): *For all natural number n it holds:*

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}. \tag{4}$$

4 Watermarks

Box with a watermark picture

Here, you see my nice box, with a picture as a watermark. This picture is automatically resized to fit the dimensions of my box. Instead of a picure, some text could be used or arbitrary graphical code. See the documentation for more options.

5 Boxes in boxes

Box

Box inside box

Box inside box inside box

And now for something completely different: Boxes!

This | is | another | box.

6 Breakable Boxes

Breakable box

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.
Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.
Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.
Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.
Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit.

N 2014-11-14 **/tcb/no process** (no default)

Removes all processing commands if set before.

N 2014-11-14 **/tcb/process code=**⟨*code*⟩ (no default, initially empty)

Adds ⟨*code*⟩ which is executed during \tcbinputlisting → P. 289 and tcblisting → P. 287. At the time of executing the given ⟨*code*⟩, the listing is already written to /tcb/listing file → P. 298, but the colored box is not constructed yet. Its intended use is to process the listing somehow before displaying. The processing result can be used inside a /tcb/comment → P. 299. Several /tcb/process code options can be given which are processed in the given order.

! To use the further options, the compiler has to be called with the -shell-escape permission to authorize potentially dangerous system calls. Be warned that this is a security risk. Anyway, it's more economic to compile examples independent from the main document and to include them as shown in the previous pages.

N 2014-11-14 **/tcb/run system command=**⟨*system command*⟩ (style, no default, initially unset)

Runs a ⟨*system command*⟩, if the document is compiled with the -shell-escape permission. The current listing file can be accessed as \filename@area\filename@base\filename@ext.

N 2014-11-14 **/tcb/compilable listing** (style, no default)

Sets /tcb/listing file → P. 298 to \jobname-listing-⟨*counter*⟩.

! The default /tcb/listing file → P. 298 setting cannot be used to compile a listing, since the base name equals the \jobname and the included PDF files should be unique. Therefore, to use /tcb/run pdflatex etc., the /tcb/listing file → P. 298 has to be set to a unique value. One may use /tcb/compilable listing for this purpose.

N 2014-11-14 **/tcb/run pdflatex=**⟨*arguments*⟩ (style, no default, initially unset)

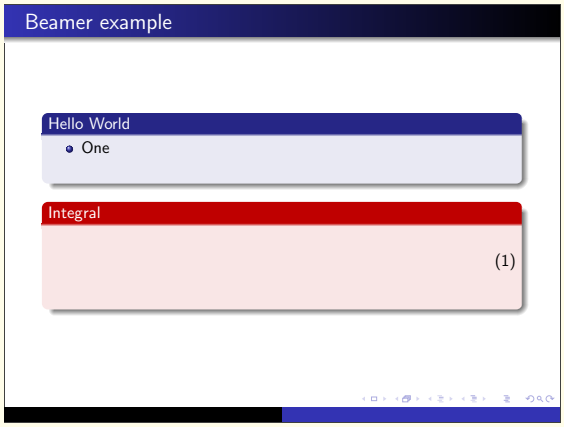
Issues a pdflatex compilation of the listing with the given ⟨*arguments*⟩.

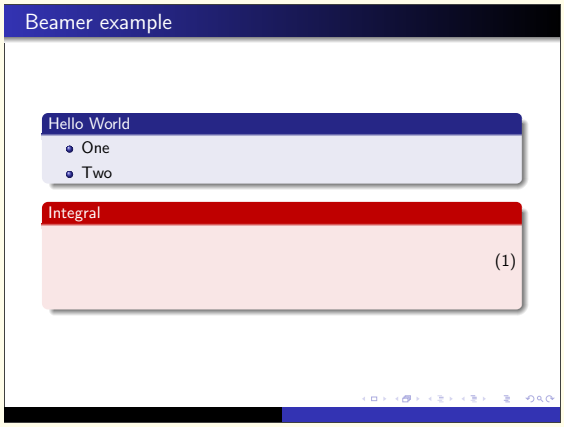
- The main document has to be compiled with the -shell-escape permission.
- The /tcb/listing file → P. 298 has to be unique for the listing.
- If the listing has to be compiled twice, add run pdflatex two times to the option list.

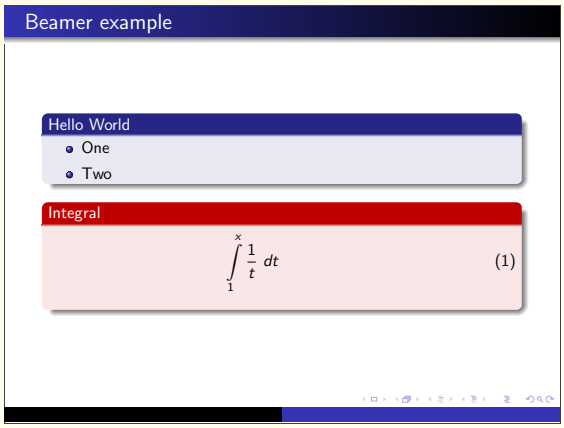
```
\begin{tcblisting}{enhanced jigsaw,lower separated=false,
  leftlower=0pt,rightlower=0pt,
  colframe=red!50!black,colback=yellow!10!white,
  listing options={style=tcblatex,texcsstyle=*\color{red!70!black}},
  listing and comment,
  pdf comment,
  compilable listing,
  run pdflatex
}
\documentclass{beamer}
\usetheme{Warsaw}
\begin{document}
\begin{frame}{Beamer example}
  \begin{block}{Hello World}
    \begin{itemize}[<+->]
      \item One
      \item Two
    \end{itemize}
  \end{block}

  \begin{alertblock}{Integral}
    \begin{equation}
      \visible<3->{\int\limits_1^x \frac{1}{t}~dt}
      \visible<4->{ = \ln(x).}
```

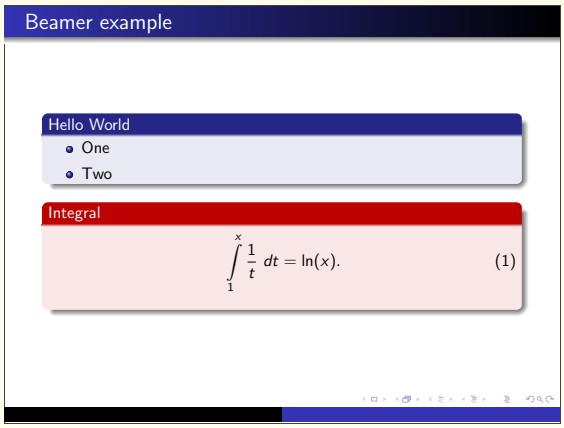
```
    \end{equation}
  \end{alertblock}
\end{frame}
\end{document}
\end{tcblisting}
```

```
\documentclass{beamer}
\usetheme{Warsaw}
\begin{document}
\begin{frame}{Beamer example}
  \begin{block}{Hello World}
    \begin{itemize}[<+->]
      \item One
      \item Two
    \end{itemize}
  \end{block}

  \begin{alertblock}{Integral}
    \begin{equation}
      \visible<3->{\int\limits_1^x \frac{1}{t}~dt}
      \visible<4->{ = \ln(x).}
    \end{equation}
  \end{alertblock}
\end{frame}
\end{document}
```









N 2014-11-14 **/tcb/run xelatex=⟨*arguments*⟩** (style, no default, initially unset)

Issues a `xelatex` compilation of the listing with the given ⟨*arguments*⟩.

N 2014-11-14 **/tcb/run lualatex=⟨*arguments*⟩** (style, no default, initially unset)

Issues a `lualatex` compilation of the listing with the given ⟨*arguments*⟩.

N 2014-11-14 **/tcb/run makeindex=⟨*arguments*⟩** (style, no default, initially unset)

Issues a `makeindex` compilation of the listing with the given ⟨*arguments*⟩.

N 2014-11-14 **/tcb/run bibtex=⟨*arguments*⟩** (style, no default, initially unset)

Issues a `bibtex` compilation of the listing with the given ⟨*arguments*⟩.

N 2014-11-14 **/tcb/run biber=⟨*arguments*⟩** (style, no default, initially unset)

Issues a `biber` compilation of the listing with the given ⟨*arguments*⟩.

N 2014-11-14 **/tcb/run arara=⟨*arguments*⟩** (style, no default, initially unset)

Issues an `arara` compilation of the listing with the given ⟨*arguments*⟩.

N 2014-11-14 **/tcb/run latex=⟨*arguments*⟩** (style, no default, initially unset)

Issues a `latex` compilation of the listing with the given ⟨*arguments*⟩.

N 2014-11-14 **/tcb/run dvips=⟨*arguments*⟩** (style, no default, initially unset)

Issues a `dvips` compilation of the listing with the given ⟨*arguments*⟩.

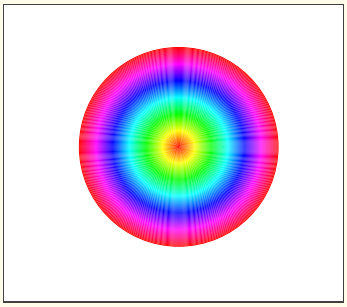
N 2014-11-14 **/tcb/run ps2pdf=⟨*arguments*⟩** (style, no default, initially unset)

Issues a `ps2pdf` compilation of the listing with the given ⟨*arguments*⟩.

```
\begin{tcblisting}{enhanced jigsaw,
  title={PSTricks with pdflatex},fonttitle=\bfseries,
  colframe=red!50!black,colback=yellow!10!white,
  listing options={style=tcblatex,texcsstyle=*\color{red!70!black}},
  lower separated=false,middle=0pt,
  listing side comment,righthand width=4.5cm,
  compilable listing,
  run latex,run dvips,run ps2pdf,
  pdf comment,
  comment style={raster columns=1,graphics options={viewport=0in 8in 4in 11.5in,clip}},
}
\documentclass{article}
\usepackage{pstricks,multido}
\begin{document}
\psset{unit=3}%
\multido{\nHue=0.01+0.01}{100}{%
\definecolor{MyColor}{hsb}{\nHue,1,1}%
\pscircle[linewidth=0.01,linecolor=MyColor]{\nHue}}
\end{document}
\end{tcblisting}
```

PSTricks with pdflatex

```
\documentclass{article}
\usepackage{pstricks,multido}
\begin{document}
\psset{unit=3}%
\multido{\nHue=0.01+0.01}{100}{%
\definecolor{MyColor}{hsb}{\nHue,1,1}%
\pscircle[linewidth=0.01,linecolor=MyColor]{\nHue}}
\end{document}
```



15.8 Creation of LaTeX Tutorials

The following source code gives a guideline for the creation of LaTeX tutorials. In the next section, a framework for LaTeX exercises is described. All examples shall be numbered optionally.

Firstly, some additional `tcb` keys are defined for the appearance. For the examples, three environments `texexp`, `texexptitled`, and `texexptitledspec` are defined with automatic numbering.

- `texexp` is used for untitled examles,
- `texexptitled` is used for titled examles,
- `texexptitledspec` is used for titled examles with special treatment.

Definition in the preamble:

```
\tcbset{
  texexp/.style={colframe=red!50!yellow!50!black, colback=red!50!yellow!5!white,
      coltitle=red!50!yellow!3!white,
      fonttitle=\small\sffamily\bfseries, fontupper=\small, fontlower=\small},
  example/.style 2 args={texexp,
      title={Example \thetcbcounter: #1},label={#2}},
}


\newtcblisting{texexp}[1]{texexp,#1}
\newtcblisting[auto counter,number within=section]{texexptitled}[3][]{%
  example={#2}{#3},#1}
\newtcolorbox[use counter from=texexptitled]{texexptitledspec}[3][]{%
  example={#2}{#3},#1}
```

```
\begin{tcblisting}{texexp}
This is a \LaTeX\ example which displays the text as source code
and in compiled form.
\end{tcblisting}
```

```
This is a \LaTeX\ example which displays the text as source code
and in compiled form.
```

This is a LaTeX example which displays the text as source code and in compiled form.

```
\begin{texexptitled}{First example with a title line}{firstExample}
Here, we use Example \ref{firstExample} with a title line.
\end{texexptitled}
```

Example 15.1: First example with a title line

```
Here, we use Example \ref{firstExample} with a title line.
```

Here, we use Example 15.1 with a title line.

```
\begin{texexp}{}
This is a \LaTeX\ example which displays the text as source code
and in compiled form.
\end{texexp}
```

```
This is a \LaTeX\ example which displays the text as source code
and in compiled form.
```

This is a LaTeX example which displays the text as source code and in compiled form.

```
\begin{texexp}{text and listing}
This is a \LaTeX\ example which displays the text as source code
and in compiled form.
\end{texexp}
```

This is a LaTeX example which displays the text as source code and in compiled form.

```
This is a \LaTeX\ example which displays the text as source code
and in compiled form.
```

```
\begin{texexp}{listing only}
This is a \LaTeX\ example which displays the text as source code only.
\end{texexp}
```

```
This is a \LaTeX\ example which displays the text as source code only.
```

```
\begin{texexp}{text only}
This is a \LaTeX\ example which displays the text in compiled form only.
\end{texexp}
```

This is a LaTeX example which displays the text in compiled form only.

```
\begin{texexptitled}{An Example with a Heading}{heading1}
This is a \LaTeX\ example with a numbered heading line
which can be referred to.
\end{texexptitled}
Here, we see Example \ref{heading1}.
```

Example 15.2: An Example with a Heading

```
This is a \LaTeX\ example with a numbered heading line
which can be referred to.
```

This is a LaTeX example with a numbered heading line which can be referred to.

Here, we see Example 15.2.

```
\begin{texexptitled}[listing only]{Another Example with a Heading}{heading2}
The keys can be used in combination. Here, an example with a heading line
and source code only is given.
\end{texexptitled}
Here, we see Example \ref{heading2}.
```

Example 15.3: Another Example with a Heading

The keys can be used in combination. Here, an example with a heading line and source code only is given.

Here, we see Example 15.3.

```
\begin{texexptitled}[float]{A floating Example with a Heading}{heading3}
This is another \LaTeX\ example with numbered heading line.
But now, the box is a floating object.
\end{texexptitled}
```

Example 15.4: A floating Example with a Heading

```
This is another \LaTeX\ example with numbered heading line.
But now, the box is a floating object.
```

This is another LaTeX example with numbered heading line. But now, the box is a floating object.

```
The floating box of the last example is seen as Example \ref{heading3}
on page \pageref{heading3}.
```

The floating box of the last example is seen as Example 15.4 on page 315.

```
\begin{texexptitledspec}{Special application}{texexpbox1}
\begin{lstlisting}[style=tcblatex]
Some \LaTeX\ source code.
\end{lstlisting}
\tcblower
For special cases, the environment |texexptitledspec| with style
|example| can be used directly. As one can see, the upper and the lower
part of the box can be used uncoupled also.
\end{texexptitledspec}
```

Example 15.5: Special application

Some **\LaTeX** source code.

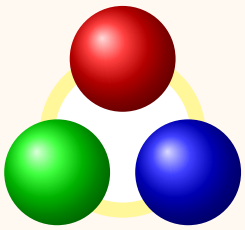
For special cases, the environment **texexptitledspec** with style **example** can be used directly. As one can see, the upper and the lower part of the box can be used uncoupled also.

The following series of examples demonstrate the application of tcolorbox → P. 12 options for diversification.

```
\begin{texexptitled}{How to use options (1):\par The basic example}{options1}
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c] (\w:1cm) circle (7mm);}
\end{tikzpicture}
\end{texexptitled}
```

Example 15.6: How to use options (1): The basic example

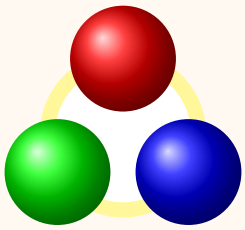
```
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c] (\w:1cm) circle (7mm);}
\end{tikzpicture}
```



```
\begin{texexptitled}[center lower,enhanced,segmentation hidden,middle=0mm]
  {How to use options (2):\par The text output is centered and the
    segmentation line has vanished.}{options2}
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c] (\w:1cm) circle (7mm);}
\end{tikzpicture}
\end{texexptitled}
```

Example 15.7: How to use options (2): The text output is centered and the segmentation line has vanished.

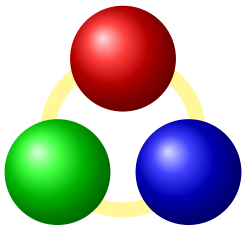
```
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c] (\w:1cm) circle (7mm);}
\end{tikzpicture}
```



```
\begin{texexptitled}[tikz lower,bicolor,colbacklower=white]
  {How to use options (3):\par Here, the |tikzpicture| is totally hidden.
    The |bicolor| skin highlights the output.}{options3}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c] (\w:1cm) circle (7mm);}
\end{texexptitled}
```

Example 15.8: How to use options (3):
Here, the `tikzpicture` is totally hidden. The `bicolor` skin highlights the output.

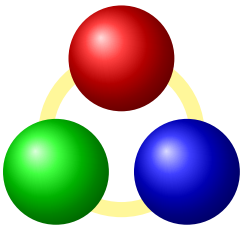
```
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c] (\w:1cm) circle (7mm);}
```



```
\begin{texexptitled}[center lower,listing side text,righthand width=3.5cm,
  bicolor,colbacklower=white]
  {How to use options (4):\par The |bicolor| skin also works with side
   by side mode}{options4}
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
\end{texexptitled}
```

Example 15.9: How to use options (4):
The `bicolor` skin also works with side by side mode

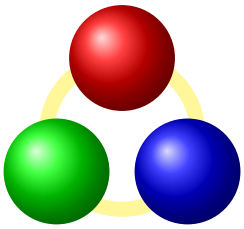
```
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
```



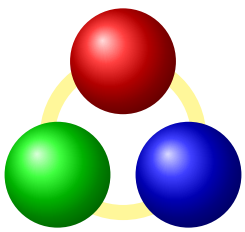
```
\begin{texexptitled}[center lower,listing outside text,righthand width=3.5cm]
  {How to use options (5):\par Putting our picture outside is just
    a matter of one word.}{options5}
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
\end{texexptitled}
```

Example 15.10: How to use options (5): Putting our picture outside is just a matter of one word.

```
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
```



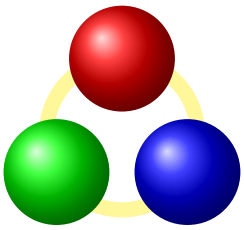
```
\begin{texexptitled}[center lower,text above listing]
  {How to use options (6):\par The picture may also be put above
    the listing box.}{options6}
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
\end{texexptitled}
```



Example 15.11: How to use options (6): The picture may also be put above the listing box.

```
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
```

```
\begin{texexptitled}[beamer,center lower,text outside listing,lefthand width=3.5cm]
  {How to use options (7):\par Our style is easily transformed into
    a beamerish one.}{options7}
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
\end{texexptitled}
```



Example 15.12: How to use options (7):
Our style is easily transformed into a beamerish one.

```
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
```

15.9 Creation of LaTeX Exercises

In the following, a guideline is given for the creation of LaTeX exercises with solutions. These solutions are saved to disk for application at a place of choice. Therefore, all used exercises are logged to a file `\jobname.records` for automatic processing. The solution contents themselves are saved to a subdirectory named `solutions`. Also see Section 8 on page 119.

- Before the first exercise is given, `\tcbstartrecording` → P. 119 has to be called to start recording.
- The solution is given as content of a `tcboutputlisting` → P. 289 environment. Note, that you can use this content also inside the exercise with `\tcbuselistingtext` → P. 289 in compiled form.
- After the last exercise is given (and before using the solutions), `\tcbstoprecording` → P. 119 has to be called to stop recording.
- The solutions are loaded by `\tcbinputrecords` → P. 119.

Inside the exercise text, there may be text parts which are needed as LaTeX source code and as compiled text as well. These parts can be saved by `tcbwritetemp` → P. 118 and used in compiled form by `\tcbusetemp` → P. 118 or as source code by `\tcbusetemplisting` → P. 289.

At first, we generate some a common style for the exercises and the solutions. Further, since exercises and solutions should be numbered, we force to use a label ⟨*marker*⟩. Automatically, the label `exe:`⟨*marker*⟩ is used to mark the exercise and the label `sol:`⟨*marker*⟩ is used to mark the solution.

```
\tcbset{texercisestyle/.style={arc=0.5mm, colframe=blue!25!yellow!90!white,
  colback=blue!25!yellow!5!white, coltitle=blue!25!yellow!40!black,
  fonttitle=\small\sffamily\bfseries, fontupper=\small, fontlower=\small,
  listing options={style=tcblatex,texcsstyle=*\color{red!40!black}},
}}
```

With these preparations, the kernel environment `texercise` for our exercises is created quickly:

Definition in the preamble:

```
\newtcolorbox[auto counter,number within=section,list inside=exam]{texercise}[2][]{%
  texercisestyle,
  listing file={solutions/texercise\thetcbcounter.tex},
  label={exe:#2},
  record={\string\processsol{solutions/texercise\thetcbcounter.tex}{#2}},
  title={Exercise \thetcbcounter\hfill\mdseries Solution on page \pageref{sol:#2}},
  list text={Exercise with solution on page \pageref{sol:#2}},#1}
```

The following examples demonstrate the application.

```
\tcbstartrecording
```

```
\begin{texercise}{tabular_example}
\textit{Create the following table:}\par\smallskip%
\begin{tcboutputlisting}
\begin{tabular}{|p{3cm}|p{3cm}|p{3cm}|p{3cm}|}\hline
\multicolumn{4}{|c|}{\bfseries\itshape Das alte Italien}\\\hline
\multicolumn{2}{|c|}{\bfseries Antike} &
\multicolumn{2}{c|}{\bfseries Mittelalter}\\\hline
\multicolumn{1}{|c|}{\itshape Republik}&
\multicolumn{1}{c|}{\itshape Kaiserreich}&
\multicolumn{1}{c|}{\itshape Franken}&
\multicolumn{1}{c|}{\itshape Teilstaaten}\\\hline
In den Zeiten der r\"{o}mischen Republik standen dem Staat jeweils zwei
Konsuln vor, deren Machtbefugnisse identisch waren. &
Das r\"{o}mische Kaiserreich wurde von einem Alleinherrscher, dem Kaiser,
regiert.
& In der V\"{o}lkerwanderungszeit \"{u}bernahmen die Goten und sp\"{a}ter die
Franken die Vorherrschaft.
& Im sp\"{a}teren Mittelalter regierten F\"{u}rsten einen Fleckenteppich
von Einzelstaaten.\\\hline
\end{tabular}
\end{tcboutputlisting}
\tcbuselistingtext%
\end{texercise}
```

Exercise 15.1 Solution on page 324

Create the following table:

Das alte Italien			
Antike		**Mittelalter**	
Republik	*Kaiserreich*	*Franken*	*Teilstaaten*
In den Zeiten der römischen Republik standen dem Staat jeweils zwei Konsuln vor, deren Machtbefugnisse identisch waren.	Das römische Kaiserreich wurde von einem Alleinherrscher, dem Kaiser, regiert.	In der Völkerwanderungszeit übernahmen die Goten und später die Franken die Vorherrschaft.	Im späteren Mittelalter regierten Fürsten einen Fleckenteppich von Einzelstaaten.

```
\begin{texercise}{macro_oneparam}
\begin{tcboutputlisting}
\newcommand{\headingline}[1]{%
  \begin{center}\Large\bfseries #1\end{center}}
\end{tcboutputlisting}
\tcbuselistingtext%

Create a new macro \verb+\headingline+ which produces the
following output:\par\smallskip
\begin{tcbwritetemp}
\headingline{Very important heading}
\end{tcbwritetemp}
\tcbusetemplisting\tcbusetemp%
\end{texercise}
```

Exercise 15.2 Solution on page 324

Create a new macro `\headingline` which produces the following output:

```
\headingline{Very important heading}
```

Very important heading

```
\begin{texercise}{macro_twoparam}
\begin{tcboutputlisting}
\newcommand{\minitable}[2]{%
  \begin{center}\begin{tabular}{p{10cm}}\hline%
  \multicolumn{1}{c}{\bfseries#1}\\\hline%
  #2\\\hline%
  \end{tabular}\end{center}}
\end{tcboutputlisting}
\tcbuselistingtext%
Create a new macro \verb+\minitable+ which produces the
following output:\par\smallskip
\begin{tcbwritetemp}
\minitable{My heading}{In this tiny tabular, there is only a heading
  and some text below which has a width of ten centimeters.}
\end{tcbwritetemp}
\tcbusetemplisting\par\smallskip\tcbusetemp%
\end{texercise}
```

Exercise 15.3 Solution on page 324

Create a new macro `\minitable` which produces the following output:

```
\minitable{My heading}{In this tiny tabular, there is only a heading
  and some text below which has a width of ten centimeters.}
```

My heading
In this tiny tabular, there is only a heading and some text below which has a width of ten centimeters.

```
\begin{texercise}{macro_threeparam}
\begin{tcboutputlisting}
\newcommand{\synop}[3]{%
  \begin{tabular}{@{}p{(\linewidth-\tabcolsep*2-\arrayrulewidth)/2}|%
    p{(\linewidth-\tabcolsep*2-\arrayrulewidth)/2}@{}}\hline
  \multicolumn{2}{c}{\bfseries #1}\\\hline
  \multicolumn{1}{c|}{\itshape English}&
  \multicolumn{1}{c}{\itshape German}\\\hline
  #2 & #3
  \end{tabular}}
\end{tcboutputlisting}
\tcbuselistingtext%
Create a new macro \verb+\synop+ which typesets a synoptic text according
to the following example. Base your macro on a tabular which takes the
total line width.\par\smallskip
\begin{tcbwritetemp}
\synop{Neil Armstrong}%
{That's one small step for a man, one giant leap for mankind.}%
{Das ist ein kleiner Schritt f\"{u}r einen Mann,
   ein riesiger Sprung f\"{u}r die Menschheit.}
\end{tcbwritetemp}
\tcbusetemplisting\par\smallskip\tcbusetemp%
\end{texercise}
```

Exercise 15.4 Solution on page 325

Create a new macro `\synop` which typesets a synoptic text according to the following example. Base your macro on a tabular which takes the total line width.

```
\synop{Neil Armstrong}%
{That's one small step for a man, one giant leap for mankind.}%
{Das ist ein kleiner Schritt f\"{u}r einen Mann,
   ein riesiger Sprung f\"{u}r die Menschheit.}
```

Neil Armstrong	
English	*German*
That's one small step for a man, one giant leap for mankind.	Das ist ein kleiner Schritt für einen Mann, ein riesiger Sprung für die Menschheit.

```
\tcbstoprecording
```

Now, we give a list of all exercises with:

```
\tcblistof[\subsection]{exam}{List of Exercises%
  \label{listofexercises}}
```

15.10 List of Exercises

15.1 Exercise with solution on page 324 321
15.2 Exercise with solution on page 324 322
15.3 Exercise with solution on page 324 322
15.4 Exercise with solution on page 325 323

15.11 Solutions for the given LaTeX Exercises

For all solutions, a macro `\processsol` was written to the file `\jobname.records`. Now, we need a definition for this macro to use the solutions.

```
% \usepackage{hyperref} % for phantomlabel
\newtcbinputlisting{\processsol}[2]{%
  texercisestyle,
  listing only,
  listing file={#1},
  phantomlabel={sol:#2},%
  title={Solution for Exercise \ref{exe:#2} on page \pageref{exe:#2}},
}
```

The loading of all solutions is done by:

```
\tcbinputrecords
```

With this, we get:

Solution for Exercise 15.1 on page 321

```
\begin{tabular}{|p{3cm}|p{3cm}|p{3cm}|p{3cm}|}\hline
\multicolumn{4}{|c|}{\bfseries\itshape Das alte Italien}\\\hline
\multicolumn{2}{|c|}{\bfseries Antike} &
\multicolumn{2}{c|}{\bfseries Mittelalter}\\\hline
\multicolumn{1}{|c|}{\itshape Republik}&
\multicolumn{1}{c|}{\itshape Kaiserreich}&
\multicolumn{1}{c|}{\itshape Franken}&
\multicolumn{1}{c|}{\itshape Teilstaaten}\\\hline
In den Zeiten der r\"{o}mischen Republik standen dem Staat jeweils zwei
Konsuln vor, deren Machtbefugnisse identisch waren. &
Das r\"{o}mische Kaiserreich wurde von einem Alleinherrscher, dem Kaiser,
regiert.
& In der V\"{o}lkerwanderungszeit \"{u}bernahmen die Goten und sp\"{a}ter die
Franken die Vorherrschaft.
& Im sp\"{a}teren Mittelalter regierten F\"{u}rsten einen Fleckenteppich
von Einzelstaaten.\\\hline
\end{tabular}
```

Solution for Exercise 15.2 on page 322

```
\newcommand{\headingline}[1]{%
  \begin{center}\Large\bfseries #1\end{center}}
```

Solution for Exercise 15.3 on page 322

```
\newcommand{\minitable}[2]{%
  \begin{center}\begin{tabular}{p{10cm}}\hline%
  \multicolumn{1}{c}{\bfseries#1}\\\hline%
  #2\\\hline%
  \end{tabular}\end{center}}
```

Solution for Exercise 15.4 on page 323

```
\newcommand{\synop}[3]{%
  \begin{tabular}{@{}p{(\linewidth-\tabcolsep*2-\arrayrulewidth)/2}|%
    p{(\linewidth-\tabcolsep*2-\arrayrulewidth)/2}@{}}\hline
  \multicolumn{2}{c}{\bfseries #1}\\\hline
  \multicolumn{1}{c|}{\itshape English}&
  \multicolumn{1}{c}{\itshape German}\\\hline
  #2 & #3
  \end{tabular}}
```

16 Library LIB theorems

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{theorems}
```

This also loads the package `amsmath`.

16.1 Macros of the Library

`\newtcbtheorem`[⟨*init options*⟩]{⟨*name*⟩}{⟨*display name*⟩}{⟨*options*⟩}{⟨*prefix*⟩}

Creates new environments ⟨*name*⟩ and ⟨*name*⟩`*` based on `tcolorbox` to frame a (mathematical) theorem. The ⟨*display name*⟩ is used in the title line with a number, e. g. «Theorem 5.1». The ⟨*options*⟩ are given to the underlying `tcolorbox` to control the appearance. The ⟨*init options*⟩ allow setting up automatic numbering, see Section 5 on page 100. The new environment ⟨*name*⟩ takes one optional and two mandatory parameters. The optional parameter supplements the options and should be used only in rare cases. The first mandatory parameter is the title text for the theorem and the second mandatory parameter is a ⟨*marker*⟩. The theorem is automatically labeled with ⟨*prefix*⟩⟨*separator*⟩⟨*marker*⟩ where ⟨*separator*⟩ is predefined as ':', see `/tcb/label separator` → P. 333.
The new environment ⟨*name*⟩`*` takes one optional and one mandatory parameter and represents an unnumbered variant of the environment ⟨*name*⟩. This variant is not labeled and not listed in lists of theorems.

Definition in the preamble:

```
\newtcbtheorem[number within=section]{mytheo}{My Theorem}%
  {colback=green!5,colframe=green!35!black,fonttitle=\bfseries}{th}
```

```
\begin{mytheo}{This is my title}{theoexample}
  This is the text of the theorem. The counter is automatically assigned and,
  in this example, prefixed with the section number. This theorem is numbered with
  \ref{th:theoexample} and is given on page \pageref{th:theoexample}.
\end{mytheo}
```

My Theorem 16.1: This is my title

This is the text of the theorem. The counter is automatically assigned and, in this example, prefixed with the section number. This theorem is numbered with 16.1 and is given on page 326.

```
\begin{mytheo}[label=myownlabel]{This is my title}{}
  The label parameter can be left empty without \LaTeX\ error.
  Or you may use an own label to reference Theorem \ref{myownlabel}.
\end{mytheo}
```

My Theorem 16.2: This is my title

The label parameter can be left empty without LaTeX error. Or you may use an own label to reference Theorem 16.2.

```
\begin{mytheo}{}{}
  The title can also be left empty without problem. Note that the ':'
  vanished magically.
\end{mytheo}
```

My Theorem 16.3

The title can also be left empty without problem. Note that the ':' vanished magically.

```
\begin{mytheo*}{Unnumbered Theorem}
  This theorem is not numbered.
\end{mytheo*}
```

My Theorem: Unnumbered Theorem

This theorem is not numbered.

```
\begin{mytheo*}{}
  This theorem has no number and no title.
\end{mytheo*}
```

My Theorem

This theorem has no number and no title.

`\renewtcbtheorem[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*display name*⟩`}{`⟨*options*⟩`}{`⟨*prefix*⟩`}`

Operates like `\newtcbtheorem` → P. 326, but based on `\renewenvironment` instead of `\newenvironment`. An existing environment is redefined.

`\tcbmaketheorem{`⟨*name*⟩`}{`⟨*display name*⟩`}{`⟨*options*⟩`}{`⟨*counter*⟩`}{`⟨*prefix*⟩`}`

! `\newtcbtheorem` → P. 326 supersedes this macro.

Creates a new environment ⟨*name*⟩ based on `tcolorbox` to frame a (mathematical) theorem. The ⟨*display name*⟩ is used in the title line with a number, e. g. «Theorem 5.1». The ⟨*options*⟩ are given to the underlying `tcolorbox` to control the appearance. The ⟨*counter*⟩ is used for automatic numbering. The new environment ⟨*name*⟩ takes one optional and two mandatory parameters. The optional parameter supplements the options and should be used only in rare cases. The first mandatory parameter is the title text for the theorem and the second mandatory parameter is a ⟨*marker*⟩. The theorem is automatically labeled with ⟨*prefix*⟩⟨*separator*⟩⟨*marker*⟩ where ⟨*separator*⟩ is predefined as ':', see `/tcb/label separator` → P. 333.

deprecated

`\tcboxmath`[⟨*options*⟩]{⟨*mathematical box content*⟩}

Creates a `tcolorbox` → P. 12 which is fitted to the width of the given ⟨*mathematical box content*⟩. This box is intended to be applied as part of a larger formula and may be used as replacement for the `\boxed` macro of `amsmath`.

```
\begin{equation}
\tcbset{fonttitle=\scriptsize}
\tcboxmath[colback=LightBlue!25!white,colframe=blue]{ a^2 = 16 }
\quad \Rightarrow \quad
\tcboxmath[colback=Salmon!25!white,colframe=red,title=Implication]%
  { a = 4 ~\vee~ a=-4. }
\end{equation}
```

$$a^2 = 16 \quad \Rightarrow \quad \underset{\text{Implication}}{a = 4 \ \vee \ a = -4.} \tag{3}$$

`\tcbhighmath`[⟨*options*⟩]{⟨*mathematical box content*⟩}

This is a special case of the `\tcboxmath` macro which uses the style `/tcb/highlight math` → P. 335. It is intended to provide context sensitive highlighting of formula parts. The color settings via `/tcb/highlight math style` → P. 335 may be different inside theorems or other colored areas and outside.

```
\tcbset{myformula/.style={colback=yellow!10!white,colframe=red!50!black,
  every box/.style={highlight math style={colback=LightBlue!50!white,colframe=Navy}}
  }}

\begin{align}
  \tcbhighmath{\sum\limits_{n=1}^{\infty} \frac{1}{n}} &= \infty.\\
  \int x^2 ~\text{d}x                                  &= \frac13 x^3 + c.
\end{align}

\begin{tcolorbox}[ams align,myformula]
  \tcbhighmath{\sum\limits_{n=1}^{\infty} \frac{1}{n}} &= \infty.\\
  \int x^2 ~\text{d}x                                  &= \frac13 x^3 + c.
\end{tcolorbox}
```

$$\boxed{\sum_{n=1}^{\infty} \frac{1}{n}} = \infty. \tag{4}$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c. \tag{5}$$

$$\boxed{\sum_{n=1}^{\infty} \frac{1}{n}} = \infty. \tag{6}$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c. \tag{7}$$

\tcbhighmath → P. 328 can be used in symbiosis with the **empheq** package which allows to specify own boxing commands to mark multiline formulas.

```
% \usepackage{empheq}
\begin{empheq}[box=\tcbhighmath]{align}
a&=\sin(z)\\
E&=mc^2 + \int_a^b x\, dx
\end{empheq}

\tcbset{highlight math style={enhanced,
  colframe=red!60!black,colback=yellow!50!white,arc=4pt,boxrule=1pt,
  drop fuzzy shadow}}

\begin{empheq}[box=\tcbhighmath]{align}
a&=\sin(z)\\
E&=mc^2 + \int_a^b x\, dx
\end{empheq}
```

$$a = \sin(z) \tag{8}$$
$$E = mc^2 + \int_a^b x\,dx \tag{9}$$

$$a = \sin(z) \tag{10}$$
$$E = mc^2 + \int_a^b x\,dx \tag{11}$$

Besides \tcbhighmath → P. 328, one can easily define an independent new box based on \tcbox → P. 14 which acts like \tcbhighmath → P. 328:

```
% \usepackage{empheq}
\newtcbox{\otherbox}[1][]{nobeforeafter,math upper,tcbox raise base,
  enhanced,frame hidden,boxrule=0pt,interior style={top color=green!10!white,
  bottom color=green!10!white,middle color=green!50!yellow},
  fuzzy halo=1pt with green,#1}

\begin{empheq}[box=\otherbox]{align}
a&=\sin(z)\\
E&=mc^2 + \int_a^b x\, dx
\end{empheq}

\begin{equation}
\tcbhighmath{E} = \otherbox{mc^2}
\end{equation}
```

$$a = \sin(z) \tag{12}$$
$$E = mc^2 + \int_a^b x\,dx \tag{13}$$

$$E = mc^2 \tag{14}$$

16.2 Option Keys of the Library

/tcb/separator sign=⟨*sign*⟩ (no default, initially :)

The given ⟨*sign*⟩ is used inside the title text of a theorem as separater between display name combined with number and the specific title text. It is omitted, if there is no specific title text.

```
% \usepackage{amssymb}
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   separator sign={\ $\blacktriangleright$}}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 16.4 ▶ My example

My theorem text.

/tcb/separator sign colon (style, no value, initially set)

Sets /tcb/separator sign to the default colon : sign.

/tcb/separator sign dash (style, no value)

Sets /tcb/separator sign to an en-dash sign.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   separator sign dash}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 16.5 – My example

My theorem text.

/tcb/separator sign none (style, no value)

Sets /tcb/separator sign to empty.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   separator sign none}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 16.6 My example

My theorem text.

`/tcb/description delimiters={⟨left⟩}{⟨right⟩}` (no default, initially empty)

The given ⟨*left*⟩ and ⟨*right*⟩ delimiter signs are used to frame the descriptive title text of a theorem.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   description delimiters={\flqq}{\frqq}}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 16.7: «My example»

My theorem text.

`/tcb/description delimiters parenthesis` (style, no value)

Sets `/tcb/description delimiters` to (and).

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   description delimiters parenthesis}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 16.8: (My example)

My theorem text.

`/tcb/description delimiters none` (style, no value, initially set)

Sets `/tcb/description delimiters` to the default empty texts.

`/tcb/description color=`⟨*color*⟩ (default empty, initially empty)

Sets the ⟨*color*⟩ of the descriptive title text deviating from `/tcb/coltitle` → P. 28. The color is reset to `/tcb/coltitle` → P. 28, if `description color` is used without value.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   description color=red!25!yellow}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 16.9: My example

My theorem text.

`/tcb/description font=`⟨*text*⟩ (default empty, initially empty)

Sets ⟨*text*⟩ (e. g. font settings) before the descriptive title text deviating from `/tcb/fonttitle` → P. 29. The ⟨*text*⟩ is removed, if `description font` is used without value.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   description delimiters={\glqq}{\grqq},
   description font=\mdseries\itshape}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 16.10: *„My example“*

My theorem text.

`/tcb/description formatter=`⟨*macro*⟩ (default empty, initially empty)

Sets ⟨*macro*⟩ as formatter for the descriptive title text. The ⟨*macro*⟩ has to take one mandatory argument (the description text).

Note that `/tcb/description delimiters` → P. 331, `/tcb/description color` → P. 331, and `/tcb/description font` are ignored, if this option is used.

If `description formatter` is used without value, the formatter is reset to its standard behavior.

```
\newtcbox{\formbox}{enhanced,frame empty,size=minimal,boxsep=2pt,arc=1pt,
  on line,interior style image=goldshade.png}

\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   description formatter=\formbox}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 16.11: My example

My theorem text.

`/tcb/terminator sign=`⟨*sign*⟩ (no default, initially empty)

The given ⟨*sign*⟩ is used as terminator at the end of the title text of a theorem.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   terminator sign={.}}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 16.12: My example.

My theorem text.

/tcb/terminator sign colon (style, no value, initially set)

Sets /tcb/terminator sign → P. 332 to the colon : sign.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   separator sign dash,terminator sign colon}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 16.13 – My example:

My theorem text.

/tcb/terminator sign dash (style, no value)

Sets /tcb/terminator sign → P. 332 to an en-dash sign.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   terminator sign dash}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 16.14: My example –

My theorem text.

/tcb/terminator sign none (style, no value)

Sets /tcb/terminator sign → P. 332 to the default empty text.

N 2016-04-19 **/tcb/label separator=⟨*separator*⟩** (no default, initially :)

The given ⟨*separator*⟩ is used for labels created with environments which are defined themselves by \newtcbtheorem → P. 326. This ⟨*separator*⟩ is put between ⟨*prefix*⟩ (defined by \newtcbtheorem → P. 326) and ⟨*marker*⟩ (defined by an actual theorem environment).

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   label separator=*}{theo}
\begin{sometheorem}{My example}{myex}
My theorem text.
\end{sometheorem}
See Example~\ref{theo*myex}.
```

Theorem 16.15: My example

My theorem text.

See Example 16.15.

/tcb/theorem name and number (style, no value, initially set)

Prints theorem name followed by theorem number inside the title.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   theorem name and number}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 16.16: My example

My theorem text.

/tcb/theorem number and name (style, no value)

Prints theorem number followed by theorem name inside the title.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   theorem number and name}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

16.17 Theorem: My example

My theorem text.

/tcb/theorem name (style, no value)

Prints theorem name without number inside the title.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   theorem name,enhanced,watermark text={\thetcbcounter}}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem: My example

My theorem text. 16.18

/tcb/theorem={⟨*display name*⟩}{⟨*counter*⟩}{⟨*title*⟩}{⟨*marker*⟩} (no default)

This key is internally used by \tcbmaketheorem → P. 327, but can be used directly in a tcolorbox for a more flexible approach. The ⟨*display name*⟩ is used together with the increased ⟨*counter*⟩ value and the ⟨*title*⟩ for the title line of the box. Additionally, a \label with the given ⟨*marker*⟩ is created.

```
% \newcounter{texercise}%  preamble
\begin{tcolorbox}[colback=green!10,colframe=green!50!black,arc=4mm,
                  theorem={Test}{texercise}{Direct usage}{myMarker}]
Here, we see the test \ref{myMarker}.
\end{tcolorbox}
```

Test 1: Direct usage

Here, we see the test 1.

For a common appearance inside the document, the key theorem should not be used directly as in the example above, but as part of a new environment created by hand or using \tcbmaketheorem → P. 327 or using its successor \newtcbtheorem → P. 326.

/tcb/highlight math (style, no value)

Predefined style which is used for \tcbhighmath → P. 328. It can be changed comfortable with /tcb/highlight math style.

/tcb/highlight math style=⟨*style definition*⟩ (style, no default)

Changes the definition for /tcb/highlight math to the given ⟨*style definition*⟩. See \tcbhighmath → P. 328 for another example.

```
% \tcbuselibrary{skins}
\tcbset{highlight math style={enhanced,%<-- needed for the 'remember' options
  colframe=red,colback=red!10!white,boxsep=0pt}}
\begin{align*}
\tcbhighmath[remember as=fx]{f(x)}
     &= \int\limits_{1}^{x} \frac{1}{t^2}~dt
      = \left[ -\frac{1}{t} \right]_{1}^{x}\\
     &= -\frac{1}{x} + \frac{1}{1}\\
     &=
\tcbhighmath[remember,overlay={%
    \draw[blue,very thick,->] (fx.south) to[bend right] ([yshift=2mm]frame.west);}]
  {1-\frac{1}{x}.}
\end{align*}
```

$$\boxed{f(x)} = \int_1^x \frac{1}{t^2}\,dt = \left[-\frac{1}{t}\right]_1^x$$
$$= -\frac{1}{x} + \frac{1}{1}$$
$$= \boxed{1 - \frac{1}{x}.}$$

`/tcb/math upper` (style, no value)

Sets the upper part to mathematical mode with font `\displaystyle`.

`/tcb/math lower` (style, no value)

Sets the lower part to mathematical mode with font `\displaystyle`.

`/tcb/math` (style, no value)

Sets the upper part *and* lower part to mathematical mode with font `\displaystyle`.

```
\begin{tcolorbox}[math,colback=yellow!10!white,colframe=red!50!black]
  \sum\limits_{n=1}^{\infty} \frac{1}{n} = \infty.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty.$$

! The following styles are only tested to work with the original `amsmath` environments. If e.g. the `equation` environment is redefined as `gather`, then `/tcb/ams equation` should / could not be used. Obviously, you are encouraged to use `/tcb/ams gather` → P. 338 in this case.

U 2014-10-30 `/tcb/ams equation upper` (style, no value)

Adds an `amsmath` `equation` environment to the start and end of the upper part.

U 2014-10-30 `/tcb/ams equation lower` (style, no value)

Adds an `amsmath` `equation` environment to the start and end of the lower part.

U 2014-10-30 `/tcb/ams equation` (style, no value)

Adds an `amsmath` `equation` environment to the start and end of the upper *and* lower part.

```
\begin{tcolorbox}[ams equation,colback=yellow!10!white,colframe=red!50!black]
  \sum\limits_{n=1}^{\infty} \frac{1}{n} = \infty.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty. \tag{15}$$

U 2014-10-30 `/tcb/ams equation* upper` (style, no value)

Adds an `amsmath` `equation*` environment to the start and end of the upper part.

U 2014-10-30 `/tcb/ams equation* lower` (style, no value)

Adds an `amsmath` `equation*` environment to the start and end of the lower part.

U 2014-10-30 `/tcb/ams equation*` (style, no value)

Adds an `amsmath` `equation*` environment to the start and end of the upper *and* lower part.

```
\begin{tcolorbox}[ams equation*,colback=yellow!10!white,colframe=red!50!black]
  \sum\limits_{n=1}^{\infty} \frac{1}{n} = \infty.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty.$$

/tcb/ams align upper (style, no value)

Adds an amsmath align environment to the start and end of the upper part.

/tcb/ams align lower (style, no value)

Adds an amsmath align environment to the start and end of the lower part.

/tcb/ams align (style, no value)

Adds an amsmath align environment to the start and end of the upper *and* lower part.

```
\begin{tcolorbox}[ams align,colback=yellow!10!white,colframe=red!50!black]
  \sum\limits_{n=1}^{\infty} \frac{1}{n} &= \infty.\\
  \int x^2 ~\text{d}x                    &= \frac13 x^3 + c.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty. \tag{16}$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c. \tag{17}$$

/tcb/ams align* upper (style, no value)

Adds an amsmath align* environment to the start and end of the upper part.

/tcb/ams align* lower (style, no value)

Adds an amsmath align* environment to the start and end of the lower part.

/tcb/ams align* (style, no value)

Adds an amsmath align* environment to the start and end of the upper *and* lower part.

```
\begin{tcolorbox}[ams align*,colback=yellow!10!white,colframe=red!50!black]
  \sum\limits_{n=1}^{\infty} \frac{1}{n} &= \infty.\\
  \int x^2 ~\text{d}x                    &= \frac13 x^3 + c.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty.$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c.$$

/tcb/ams gather upper (style, no value)

Adds an amsmath gather environment to the start and end of the upper part.

/tcb/ams gather lower (style, no value)

Adds an amsmath gather environment to the start and end of the lower part.

/tcb/ams gather (style, no value)

Adds an amsmath gather environment to the start and end of the upper *and* lower part.

```
\begin{tcolorbox}[ams gather,colback=yellow!10!white,colframe=red!50!black]
  \sum\limits_{n=1}^{\infty} \frac{1}{n} = \infty.\\
  \int x^2 ~\text{d}x = \frac13 x^3 + c.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty. \tag{18}$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c. \tag{19}$$

/tcb/ams gather* upper (style, no value)

Adds an amsmath gather* environment to the start and end of the upper part.

/tcb/ams gather* lower (style, no value)

Adds an amsmath gather* environment to the start and end of the lower part.

/tcb/ams gather* (style, no value)

Adds an amsmath gather* environment to the start and end of the upper *and* lower part.

```
\begin{tcolorbox}[ams gather*,colback=yellow!10!white,colframe=red!50!black]
  \sum\limits_{n=1}^{\infty} \frac{1}{n} = \infty.\\
  \int x^2 ~\text{d}x = \frac13 x^3 + c.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty.$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c.$$

`/tcb/ams nodisplayskip upper` (style, no value)

Neutralizes the `\abovedisplayskip` of a following `align` or `gather` environment for the upper part. Note that the text content has to start with such a formula.

`/tcb/ams nodisplayskip lower` (style, no value)

Neutralizes the `\abovedisplayskip` of a following `align` or `gather` environment for the lower part. Note that the text content has to start with such a formula.

`/tcb/ams nodisplayskip` (style, no value)

Neutralizes the `\abovedisplayskip` of a following `align` or `gather` environment for the upper part *and* lower part. Note that the text content has to start with such a formula.

```
\begin{tcolorbox}[ams nodisplayskip,colback=yellow!10!white,colframe=red!50!black]
  \begin{gather}
  \sum\limits_{n=1}^{\infty} \frac{1}{n} = \infty.\\
  \int x^2 ~\text{d}x = \frac13 x^3 + c.
  \end{gather}
  And now for something completely different.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty. \tag{20}$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c. \tag{21}$$

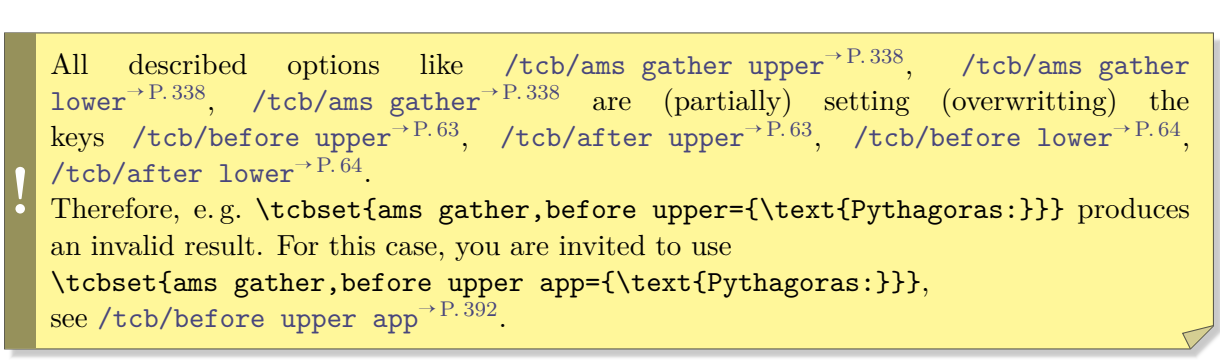
And now for something completely different.

New colored mathematical environments are easily created using `\newtcolorbox` → P. 15:

```
\newtcolorbox{mymath}{ams gather*,colback=yellow!10!white,colframe=red!50!black}

\begin{mymath}
  \sum\limits_{n=1}^{\infty} \frac{1}{n} = \infty.\\
  \int x^2 ~\text{d}x = \frac13 x^3 + c.
\end{mymath}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty.$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c.$$



All described options like `/tcb/ams gather upper` → P. 338, `/tcb/ams gather lower` → P. 338, `/tcb/ams gather` → P. 338 are (partially) setting (overwritting) the keys `/tcb/before upper` → P. 63, `/tcb/after upper` → P. 63, `/tcb/before lower` → P. 64, `/tcb/after lower` → P. 64.
Therefore, e. g. `\tcbset{ams gather,before upper={\text{Pythagoras:}}}` produces an invalid result. For this case, you are invited to use
`\tcbset{ams gather,before upper app={\text{Pythagoras:}}}`,
see `/tcb/before upper app` → P. 392.

`/tcb/theorem style=`⟨*name*⟩ (no default, initially `standard`)

Applies a predefined style ⟨*name*⟩ to the theorem environment. Some of the feasible ⟨*name*⟩ values resemble style names from the packages `theorem` and `ntheorem` to give convenient access to known patterns.

> **!** The styles alter `/tcb/separator sign` → P. 330, `/tcb/description delimiters` → P. 331, `/tcb/terminator sign` → P. 332, and more. Therefore, one should apply such keys *after* a theorem style.

For the following examples, we use:

Definition in the preamble:

```
\newtcbtheorem[use counter from=mytheo]{theorem}{Theorem}{%
  fonttitle=\bfseries\upshape,fontupper=\itshape,
  colframe=green!50!black,colback=green!10!white,
  colbacktitle=green!20!white,coltitle=blue!75!black}{theo}
```

The predefined styles are:

- `standard`: This is the initial value.

```
\begin{theorem}[theorem style=standard]{standard}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

Theorem 16.19: standard

This is my theorem.

$$a^2 + b^2 = c^2.$$

- `change standard`

```
\begin{theorem}[theorem style=change standard]{change standard}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

16.20 Theorem: change standard

This is my theorem.

$$a^2 + b^2 = c^2.$$

- `plain`

```
\begin{theorem}[theorem style=plain]{plain}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

Theorem 16.21 (plain): *This is my theorem.*

$$a^2 + b^2 = c^2.$$

- **break**

```
\begin{theorem}[theorem style=break]{break}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

Theorem 16.22 (break):
This is my theorem.

$$a^2 + b^2 = c^2.$$

- **plain apart**

```
\begin{theorem}[theorem style=plain apart]{plain apart}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

Theorem 16.23 (plain apart)

This is my theorem.

$$a^2 + b^2 = c^2.$$

- **change**

```
\begin{theorem}[theorem style=change]{change}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

16.24 Theorem (change): *This is my theorem.*

$$a^2 + b^2 = c^2.$$

- **change break**

```
\begin{theorem}[theorem style=change break]{change break}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

16.25 Theorem (change break):
This is my theorem.

$$a^2 + b^2 = c^2.$$

- **change apart**

```
\begin{theorem}[theorem style=change apart]{change apart}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

16.26 Theorem (change apart)

This is my theorem.

$$a^2 + b^2 = c^2.$$

- **margin**

```
\begin{theorem}[theorem style=margin,left=10mm]{margin}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
\begin{theorem}[theorem style=margin,left=10mm,oversize]{margin}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

16.27 Theorem (margin): *This is my theorem.*

$$a^2 + b^2 = c^2.$$

16.28 Theorem (margin): *This is my theorem.*

$$a^2 + b^2 = c^2.$$

- **margin break**

```
\begin{theorem}[theorem style=margin break,left=10mm]{margin break}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
\begin{theorem}[theorem style=margin break,left=10mm,oversize]{margin break}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

16.29 Theorem (margin break):
This is my theorem.

$$a^2 + b^2 = c^2.$$

16.30 Theorem (margin break):
This is my theorem.

$$a^2 + b^2 = c^2.$$

- **margin apart**

```
\begin{theorem}[theorem style=margin apart,left=10mm]{margin apart}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
\begin{theorem}[theorem style=margin apart,left=10mm,oversize]{margin apart}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

16.31 Theorem (margin apart)

This is my theorem.

$$a^2 + b^2 = c^2.$$

16.32 Theorem (margin apart)

This is my theorem.

$$a^2 + b^2 = c^2.$$

16.3 Examples for Definitions and Theorems

In the following, the application of \tcbmaketheorem → P. 327 to highlight mathematical definitions, theorems, or the like is demonstrated.

At first, additional tcb keys are created for the appearance of the colored boxes. It is assumed that theorems and corollaries should be identically colored. All following environments are numbered with a common counter, but this can be changed easily. Here, the counter output is supplemented by the subsection number. Further, the cleveref package [5] is used for clever references.

Definition in the preamble:

```
% \usepackage{cleveref}
\tcbset{
  defstyle/.style={fonttitle=\bfseries\upshape, fontupper=\slshape,
              arc=0mm, colback=blue!5!white,colframe=blue!75!black},
  theostyle/.style={fonttitle=\bfseries\upshape, fontupper=\slshape,
               colback=red!10!white,colframe=red!75!black},
}
\newtcbtheorem[number within=subsection,crefname={definition}{definitions}]%
  {Definition}{Definition}{defstyle}{def}
\newtcbtheorem[use counter from=Definition,crefname={theorem}{theorems}]%
  {Theorem}{Theorem}{theostyle}{theo}
\newtcbtheorem[use counter from=Definition,crefname={corollary}{corollaries}]%
  {Corollary}{Corollary}{theostyle}{cor}
```

By \newtcbtheorem → P. 326, commonly numbered theorem environments are created now. defstyle and theostyle are used for the appearance.

Now, everything is prepared for the following examples.

```
The following theorem is numbered as \Cref{theo:diffbarstetig} and
referenced with the marker \texttt{theo:diffbarstetig}.\bigskip

\begin{Theorem}{Differenzierbarkeit bedingt Stetigkeit, wobei diese Benennung
  zu Testzwecken ungew\"{o}hnlich lang ist}{diffbarstetig}%
  Eine Funktion $f:I\to\mathbb{R}$ ist in $x_0\in I$ stetig, wenn $f$ in
  $x_0$ differenzierbar ist.
\end{Theorem}
```

The following theorem is numbered as Theorem 16.3.1 and referenced with the marker theo:diffbarstetig.

Theorem 16.3.1: Differenzierbarkeit bedingt Stetigkeit, wobei diese Benennung zu Testzwecken ungewöhnlich lang ist

Eine Funktion $f : I \to \mathbb{R}$ ist in $x_0 \in I$ stetig, wenn f in x_0 differenzierbar ist.

```
The following definition is numbered as \Cref{def:diffbarkeit} and
referenced with the marker \texttt{def:diffbarkeit}.\bigskip

\begin{Definition}{Differenzierbarkeit}{diffbarkeit}
  Eine Funktion $f:~I\to\mathbb{R}$ auf einem Intervall $I$ hei\ss{}t in
  $x_0\in I$ differenzierbar oder linear approximierbar,
  wenn der Grenzwert
  \begin{equation*}
  \lim\limits_{x\to x_0}\frac{f(x)-f(x_0)}{x-x_0}=
  \lim\limits_{h\to 0}\frac{f(x_0+h)-f(x_0)}{h}
  \end{equation*}
  existiert. Bei Existenz hei\ss{}t dieser Grenzwert Ableitung
  oder Differentialquotient von $f$ in $x_0$ und man
  schreibt f\"{u}r ihn
  \begin{equation*}
  f'(x_0)\quad\text{oder}\quad\frac{df}{dx}(x_0).
  \end{equation*}
\end{Definition}
```

The following definition is numbered as Definition 16.3.2 and referenced with the marker `def:diffbarkeit`.

Definition 16.3.2: Differenzierbarkeit

Eine Funktion $f: I \to \mathbb{R}$ *auf einem Intervall* I *heißt in* $x_0 \in I$ *differenzierbar oder linear approximierbar, wenn der Grenzwert*

$$\lim_{x\to x_0} \frac{f(x)-f(x_0)}{x-x_0} = \lim_{h\to 0} \frac{f(x_0+h)-f(x_0)}{h}$$

existiert. Bei Existenz heißt dieser Grenzwert Ableitung oder Differentialquotient von f *in* x_0 *und man schreibt für ihn*

$$f'(x_0) \quad \text{oder} \quad \frac{df}{dx}(x_0).$$

```
The following corollary is numbered as \Cref{cor:nullstellen} and
referenced with the marker \texttt{cor:nullstellen}.\bigskip

\begin{Corollary}{Nullstellenexistenz}{nullstellen}
  Ist $f:[a,b]\to\mathbb{R}$ stetig und haben $f(a)$ und $f(b)$ entgegengesetzte
  Vorzeichen, also $f(a)f(b)<0$, so besitzt $f$ eine Nullstelle $x_0\in]a,b[$,
  also $f(x_0)=0$.
\end{Corollary}
```

The following corollary is numbered as Corollary 16.3.3 and referenced with the marker `cor:nullstellen`.

Corollary 16.3.3: Nullstellenexistenz

Ist $f:[a,b]\to\mathbb{R}$ *stetig und haben* $f(a)$ *und* $f(b)$ *entgegengesetzte Vorzeichen, also* $f(a)f(b)<0$*, so besitzt* f *eine Nullstelle* $x_0 \in]a,b[$*, also* $f(x_0)=0$.

```
\begin{Theorem}[boxrule=2mm,toptitle=-1.5mm,bottomtitle=-1.5mm]{%
    Hinreichende Bedingung f\"{u}r Wendepunkte}{wendehinreichend}%
  $f$ sei eine auf einem Intervall $]a,b[$ dreimal stetig differenzierbare Funktion.
  Ist $f''(x_0)=0$ in $x_0\in]a,b[$ und $f'''(x_0)\ne 0$, so ist
  $(x_0,f(x_0))$ ein Wendepunkt von $f$.
\end{Theorem}
```

Theorem 16.3.4: Hinreichende Bedingung für Wendepunkte

f sei eine auf einem Intervall $]a,b[$ dreimal stetig differenzierbare Funktion. Ist $f''(x_0)=0$ in $x_0\in]a,b[$ und $f'''(x_0)\neq 0$, so ist $(x_0,f(x_0))$ ein Wendepunkt von f.

```
% \usepackage{varioref}
% \usepackage{cleveref}
% \tcbuselibrary{skins}
\newtcbtheorem[use counter from=Definition]{YetAnotherTheorem}{Theorem}%
  {theorem style=plain apart,label type=theorem,enhanced,frame hidden,
  boxrule=2mm,titlerule=0mm,toptitle=1mm,bottomtitle=1mm,
  fonttitle=\bfseries\large,fontupper=\normalsize,
  coltitle=green!35!black,colbacktitle=green!15!white,
  colback=green!50!yellow!15!white,borderline={1pt}{0pt}{green!25!blue},
  }{theo}

\begin{YetAnotherTheorem}{Mittelwertsatz f\"{u}r $n$ Variable}{meanvaluetheorem}%
  Es sei $n\in\mathbb{N}$, $D\subseteq\mathbb{R}^n$ eine offene Menge und
  $f\in C^{1}(D,\mathbb{R})$. Dann gibt es auf jeder Strecke
  $[x_0,x]\subset D$ einen Punkt $\xi\in[x_0,x]$, so dass gilt
  \begin{equation*}
  f(x)-f(x_0) = \operatorname{grad} f(\xi)^{\top}(x-x_0)
  \end{equation*}
\end{YetAnotherTheorem}

\medskip
Here, |cleveref| support is used to reference \Cref{theo:meanvaluetheorem}
on \Cpageref{theo:meanvaluetheorem}. This \namecref{theo:meanvaluetheorem}
can also be referenced by |\Vref| resulting in \Vref{theo:meanvaluetheorem}.
```

Theorem 16.3.5 (Mittelwertsatz für n Variable)

Es sei $n \in \mathbb{N}$, $D \subseteq \mathbb{R}^n$ eine offene Menge und $f \in C^1(D, \mathbb{R})$. Dann gibt es auf jeder Strecke $[x_0, x] \subset D$ einen Punkt $\xi \in [x_0, x]$, so dass gilt

$$f(x) - f(x_0) = \operatorname{grad} f(\xi)^{\top}(x - x_0)$$

Here, `cleveref` support is used to reference Theorem 16.3.5 on Page 345. This theorem can also be referenced by `\Vref` resulting in Theorem 16.3.5.

Note that `/tcb/label type` → P. 94 was used in the example above to feed `cleveref` [5] with the needed name information.

```
Here, using |\Vref| resulting in \Vref{theo:meanvaluetheorem} is more interesting\ldots
```

Here, using \Vref resulting in Theorem 16.3.5 on page 345 is more interesting...

```
% \tcbuselibrary{skins}
\newtcbtheorem[use counter from=Definition]{YetAnotherTheorem}{Theorem}%
  {theorem style=change apart,enhanced,arc=0mm,outer arc=0mm,
  boxrule=0mm,toprule=1mm,bottomrule=1mm,left=1mm,right=1mm,
  titlerule=0mm,toptitle=0mm,bottomtitle=1mm,top=0mm,
  colframe=red!50!black,colback=red!5!white,coltitle=red!50!black,
  title style={top color=yellow!50!white,bottom color=red!5!white,
    middle color=yellow!50!white},
  fonttitle=\bfseries\sffamily\normalsize,fontupper=\normalsize\itshape,
  }{theo}

\begin{YetAnotherTheorem}{Mittelwertsatz f\"{u}r $n$ Variable}{mittelwertsatz_n2}%
  Es sei $n\in\mathbb{N}$, $D\subseteq\mathbb{R}^n$ eine offene Menge und
  $f\in C^{1}(D,\mathbb{R})$. Dann gibt es auf jeder Strecke
  $[x_0,x]\subset D$ einen Punkt $\xi\in[x_0,x]$, so dass gilt
  \begin{equation*}
  f(x)-f(x_0) = \operatorname{grad} f(\xi)^{\top}(x-x_0)
  \end{equation*}
\end{YetAnotherTheorem}
```

16.3.6 Theorem (Mittelwertsatz für n Variable)

Es sei $n \in \mathbb{N}$, $D \subseteq \mathbb{R}^n$ eine offene Menge und $f \in C^1(D, \mathbb{R})$. Dann gibt es auf jeder Strecke $[x_0, x] \subset D$ einen Punkt $\xi \in [x_0, x]$, so dass gilt

$$f(x) - f(x_0) = \operatorname{grad} f(\xi)^\top (x - x_0)$$

```
% \usepackage{varwidth}   \tcbuselibrary{skins}
\newtcbtheorem[use counter from=Definition]{YetAnotherTheorem}{Theorem}%
{enhanced,frame empty,interior empty,colframe=ForestGreen!50!white,
  coltitle=ForestGreen!50!black,fonttitle=\bfseries,colbacktitle=ForestGreen!15!white,
  borderline={0.5mm}{0mm}{ForestGreen!15!white},
  borderline={0.5mm}{0mm}{ForestGreen!50!white,dashed},
  attach boxed title to top center={yshift=-2mm},
  boxed title style={boxrule=0.4pt},varwidth boxed title}{theo}

\begin{YetAnotherTheorem}{Mittelwertsatz f\"{u}r $n$ Variable}{mittelwertsatz_n3}%
  Es sei $n\in\mathbb{N}$, $D\subseteq\mathbb{R}^n$ eine offene Menge und
  $f\in C^{1}(D,\mathbb{R})$. Dann gibt es auf jeder Strecke
  $[x_0,x]\subset D$ einen Punkt $\xi\in[x_0,x]$, so dass gilt
  \begin{equation*}
  f(x)-f(x_0) = \operatorname{grad} f(\xi)^{\top}(x-x_0)
  \end{equation*}
\end{YetAnotherTheorem}
```

Theorem 16.3.7: Mittelwertsatz für n Variable

Es sei $n \in \mathbb{N}$, $D \subseteq \mathbb{R}^n$ eine offene Menge und $f \in C^1(D, \mathbb{R})$. Dann gibt es auf jeder Strecke $[x_0, x] \subset D$ einen Punkt $\xi \in [x_0, x]$, so dass gilt

$$f(x) - f(x_0) = \operatorname{grad} f(\xi)^\top (x - x_0)$$

You need more attention for your theorems? Here, you are ...

```
% tcbuselibrary{skins}  % preamble
\begin{Theorem}[enhanced,
    fuzzy halo=3mm with yellow,
    fuzzy halo=2mm with red,
    fuzzy halo=1mm with yellow,
    watermark color=red!35!white,
    watermark text={Overacting\\Fundamental Theorem}]%
  {Fundamental Theorem of Theorems}{fundamental}%
  \lipsum[1-2]
\end{Theorem}
```

Theorem 16.3.8: Fundamental Theorem of Theorems

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Overacting
Fundamental Theorem

Let's try a more conservative approach:

```
% \tcbuselibrary{skins}
\newtcbtheorem[use counter from=Definition]{YetAnotherTheorem}{Theorem}%
  {theorem style=plain,enhanced,colframe=blue!50!black,colback=yellow!20!white,
   coltitle=red!50!black,fonttitle=\upshape\bfseries,fontupper=\itshape,
   drop fuzzy shadow=blue!50!black!50!white,boxrule=0.4pt}{theo}

\begin{YetAnotherTheorem}{Mittelwertsatz f\"{u}r $n$ Variable}{mittelwertsatz_n4}%
  Es sei $n\in\mathbb{N}$, $D\subseteq\mathbb{R}^n$ eine offene Menge und
  $f\in C^{1}(D,\mathbb{R})$. Dann gibt es auf jeder Strecke
  $[x_0,x]\subset D$ einen Punkt $\xi\in[x_0,x]$, so dass gilt
  \begin{equation*}
  f(x)-f(x_0) = \operatorname{grad} f(\xi)^{\top}(x-x_0)
  \end{equation*}
\end{YetAnotherTheorem}
```

Theorem 16.3.9 (Mittelwertsatz für n Variable): *Es sei $n \in \mathbb{N}$, $D \subseteq \mathbb{R}^n$ eine offene Menge und $f \in C^1(D, \mathbb{R})$. Dann gibt es auf jeder Strecke $[x_0, x] \subset D$ einen Punkt $\xi \in [x_0, x]$, so dass gilt*

$$f(x) - f(x_0) = \operatorname{grad} f(\xi)^\top (x - x_0)$$

16.4 Using other theorem environments with `tcolorbox`

Instead of creating theorem environments with the methods described before, environments from other packages can be boxed with a `tcolorbox`.

Environments may be created e.g. by methods from the `theorem` package or the `amsthm` package. `\tcolorboxenvironment` → P. 17 can be used to put a box around these environments.

Definition in the preamble:

```
\usepackage{amsthm}

\theoremstyle{plain}%    from 'amsthm'
\newtheorem{lem}{Lemma}% from 'amsthm'

\tcolorboxenvironment{lem}{
  enhanced jigsaw,colframe=cyan,interior hidden,
  breakable,before skip=10pt,after skip=10pt }

\tcolorboxenvironment{proof}{% 'proof' from 'amsthm'
  blanker,breakable,left=5mm,
  before skip=10pt,after skip=10pt,
  borderline west={1mm}{0pt}{red}}
```

```
\begin{lem}
  \lipsum[2]
\end{lem}

\lipsum[3]

\begin{proof}
  \lipsum*[4]
\end{proof}
```

Lemma 1. *Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.*

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Proof. Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. □

17 Library LIB breakable

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{breakable}
```

17.1 Technical Overview

The library LIB breakable supports the automatic breaking of a `tcolorbox`. This feature is enabled by `/tcb/breakable` → P. 351 and disabled by `/tcb/unbreakable` → P. 352.

If a `tcolorbox` is set to be `/tcb/breakable` → P. 351, then the following algorithm is executed:

1. The box content is read to a box register similar but not identical to the unbreakable case.
2. If the total box fits into the current page, it is shipped out visibly unbroken and the algorithm stops.

   Unbroken Box

   The box. unbroken

3. Otherwise, it is checked if at least `/tcb/lines before break` → P. 352 of the upper box can be placed on the current page. If not, a page break is inserted and the algorithm goes back to Step 2.
4. Now, the *break sequence* starts. The upper box part or the lower box part is split such that it fits into the current page. The fitting part is named *first part* of the *break sequence* and shipped out.

   Broken Box

   The box. first

5. If the remaining content of the total box fits into the current page, the algorithm continues with Step 7, else with Step 6.
6. The upper box part or the lower box part is split such that it fits into the current page. The fitting part is named *middle part* of the *break sequence* and shipped out. Then, the algorithm goes back to Step 5.

   The box. middle

7. The remaining part is named *last part* of the *break sequence* and shipped out. The algorithm stops.

   The box. last

The algorithm takes care that the optional segmentation line never appears at the end of a box. The optional lower box part is also checked to have at least `/tcb/lines before break` → P. 352.

In principle, all boxes of the *break sequence* share the same geometric parameters. The differences are:

- The given /tcb/before → P. 76 and /tcb/after → P. 76 values are used only before the *first* and after the *last* part of the *break sequence*.
- A special behavior between the parts of the *break sequence* can be given by /tcb/toprule at break → P. 355, /tcb/bottomrule at break → P. 355, /tcb/enlarge top at break by → P. 81, and /tcb/enlarge bottom at break by → P. 81.
- The /tcb/skin → P. 125 decides *how* the *first*, *middle*, and *last* part look like. Actually, every part type has its own skin given by the options /tcb/skin first → P. 125, /tcb/skin middle → P. 125, and /tcb/skin last → P. 125. Typically, these options are set automatically by the main skin, see Subsection 17.7 from page 362.

17.2 Limitations and Known Bugs

- The maximal total height of the upper and of the lower part of normal breakable tcolorboxes is about 65536pt (ca. 2300cm) apiece. If such a part gets longer, the output will get buggy without warning. For very oversized boxes which are longer than 65536pt, use the unlimited value for /tcb/breakable → P. 351. With the unlimited setting, the applied algorithm has (virtually) no height limit for boxes, but very likely the compiler memory will have to be increased for boxes longer than 300 pages (depending on compiler settings and box content). But it is recommended to use unlimited for critical large boxes only.
- You can nest an unbreakable tcolorbox inside another tcolorbox, even inside a breakable one. But you cannot not nest a breakable box inside a breakable box. The /tcb/breakable → P. 351 key for a nested box is ignored automatically[3], i. e. inner boxes are always unbreakable.

  After all, in the unlikely case you really want to have the nested box to be breakable, use /tcb/enforce breakable → P. 352 for the nested box[4]. **But, a breakable box inside a breakable box will usually give a mess.**
- If your text content contains some text color changing commands, your color will not survive the break to the next box. But, with the fontspec package and xelatex or lualatex, you can use \addfontfeatures{Color=mycolor} to add a font color which survives the break.

N 2014-10-30

- The perpage option of the footmisc package is deliberately deactivated inside a breakable box since all footnotes are placed at the end of the box (possibly far away from the reference point).

N 2016-02-15

- Making a box /tcb/breakable → P. 351 which actually is not broken creates a box which acts *almost* like an unbreakable box. Visual differences are kept as indiscernible as possible, but can appear with certain /tcb/before → P. 76 and /tcb/after → P. 76 settings, especially, if there is an automatic page break before the box.

[3] Until tcolorbox 3.04, the /tcb/breakable → P. 351 key was not ignored for nested boxes.
[4] /tcb/enforce breakable → P. 352 acts like /tcb/breakable → P. 351 until tcolorbox 3.04.

17.3 Main Option Keys

`/tcb/breakable=true|false|unlimited` (default `true`, initially `false`)

Allows the `tcolorbox` to be breakable. If the box is larger than the available space at the current page, the box is automatically broken and continued to the next next page. All sorts of `tcolorbox` can be made breakable. It depends on the skin how the breaking looks like. If you do not know better, use `/tcb/enhanced` → P. 195 for breaking a box. The parts of the *break sequence* are numbered by the counter `tcbbreakpart`.

- `false`: Sets the `tcolorbox` to be unbreakable.
- `true`: Breaks the `tcolorbox` from one page to another. The maximal total height of the upper and of the lower part is about 65536pt (ca. 2300cm or ca. 90 pages) apiece.
- `unlimited`: Experimental code for unlimited total height of breakable boxes. For boxes longer than 300 pages (or even shorter ones) the compiler memory will have to be increased.

```
% \usepackage{lipsum}  % preamble
\tcbset{enhanced jigsaw,colback=red!5!white,colframe=red!75!black,
  watermark color=yellow!25!white,watermark text=\arabic{tcbbreakpart},
  fonttitle=\bfseries}

\begin{tcolorbox}[breakable,title=My breakable box]
\lipsum[1-6]
\end{tcolorbox}
```

My breakable box

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea

dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.
Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.
Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

`/tcb/unbreakable` (no value, initially set)

Sets the `tcolorbox` to be unbreakable.

`/tcb/enforce breakable` (no value)

A `tcolorbox` inside a `tcolorbox` is automatically set to be unbreakable. Using `/tcb/breakable` → P. 351 on such an inner box has no effect. If one *really* wants the inner box to be breakable, use `/tcb/enforce breakable`. **This will usually give a mess of shattered boxes. You are advised to not use this option.**

Note that `/tcb/enforce breakable` has the functionality that `/tcb/breakable` → P. 351 had until package version 3.04 and exists for backward compatibility.

`/tcb/title after break=`⟨*text*⟩ (no default, initially empty)

The `/tcb/title` → P. 18 is used only for the *first* part of a *break sequence*. Use `title after break` to create a heading line with ⟨*text*⟩ as content for all following parts.

`/tcb/notitle after break` (no value, initially set)

Removes the title line or following parts in a *break sequence* if set before.

`/tcb/adjusted title after break=`⟨*text*⟩ (style, no default, initially unset)

Works like `/tcb/adjusted title` → P. 18 but applied to `/tcb/title after break`.

`/tcb/lines before break=`⟨*number*⟩ (no default, initially 2)

Assures that the given ⟨*number*⟩ of lines of the upper box part or the lower box part are placed before a break happens.

`/tcb/break at`=⟨*length*⟩/⟨*length*⟩/.../⟨*length*⟩ (no default, initially `0pt`)

Defines break points at the given ⟨*length*⟩ values. The first ⟨*length*⟩ defines the (maximal) height of the first partial box, the second ⟨*length*⟩ defines the (maximal) height of the second partial box, and so on. The last ⟨*length*⟩ value is applied to all following partial boxes if any. Setting a length to `0pt` means that the naturally available space is used for breaking.

```
% \usepackage{multicol,lipsum}
\begin{multicols}{3}\footnotesize
Breakable boxes inside a |multicols| environment need special attendance.
They are broken by default at |\textheight|.
The |break at| option can be used to insert better break points by hand.
\begin{tcolorbox}[enhanced jigsaw,size=small,vfill before first,
  colframe=red,colback=yellow!10!white,before title=\raggedright,
  title={Broken box inside a |multicols| environment},fonttitle=\bfseries,
  enforce breakable,% use only breakable in the real world!
  pad at break=1mm,break at=3cm/6.3cm ]
\lipsum[1]
\end{tcolorbox}
\refKey{/tcb/height fixed for} may also be considered for |multicols| environments.
\end{multicols}
```

Breakable boxes inside a `multicols` environment need special attendance. They are broken by default at `\textheight`. The `break at` option can be used to insert better break points by hand.

Broken box inside a `multicols` environment

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

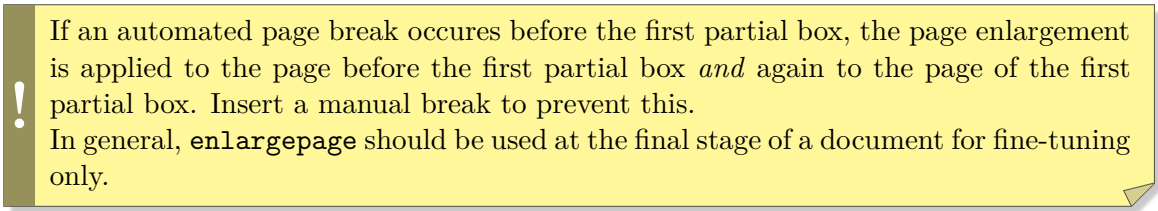
`/tcb/height fixed for` → P. 356 may also be considered for `multicols` environments.

`/tcb/enlargepage`=⟨*length*⟩/⟨*length*⟩/.../⟨*length*⟩ (no default, initially `0pt`)

Inserts a `\enlargethispage{`⟨*length*⟩`}` to the pages of the break sequence, i. e. allows one to enlarge (or shrink) partial boxes. The first ⟨*length*⟩ is applied to the first partial box, the second ⟨*length*⟩ is applied to the second partial box, and so on. The last ⟨*length*⟩ value is applied to all following partial boxes if any. Note that floating boxes will not be enlarged.

```
\begin{tcolorbox}[breakable,enlargepage=0mm/\baselineskip/2\baselineskip/0mm,...
```

The example code enlarged the second partial box by one line, the third partial box by two lines, and all following parts are not enlarged.



If an automated page break occures before the first partial box, the page enlargement is applied to the page before the first partial box *and* again to the page of the first partial box. Insert a manual break to prevent this.
In general, `enlargepage` should be used at the final stage of a document for fine-tuning only.

`/tcb/enlargepage flexible=`⟨*length*⟩ (no default, initially `0pt`)

This allows an automated page enlargement for up to ⟨*length*⟩. The algorithm can use this to avoid breaking a box, if there is anough room after enlargement. Also, the *last* partial box of a break sequence may be enlarged to avoid further breaking.
Note that this potential enlargement is *additive* to settings of `/tcb/enlargepage` → P. 353. But `/tcb/enlargepage flexible` overwrites settings of `/tcb/pad before break*` → P. 355 or `/tcb/pad at break*` → P. 355.

```
% The following setting hinders orphan lines for the last partial box
\tcbset{enlargepage flexible=\baselineskip}
```

N 2014-12-15 `/tcb/compress page=`⟨*option*⟩ (default `all`, initially `baselineskip`)

This option controls the space management on the page which contains the unbroken box or the first part of a *break sequence*. Feasible ⟨*option*⟩ values are:

- `all` (default value): All shrinkable glue on the page is potentially used for the unbroken box or the first part of a *break sequence*. Thus, all vertical spaces on the page will potentially be reduced to their minimal values.
- `baselineskip` (initial value): Shrinkable glue up to one `\baselineskip` on the page is potentially used for the unbroken box or the first part of a *break sequence*.
- `none`: The break algorithm respects the target size of the given glue values on the page. This was the inital value before version `3.34`.

Note that the box *content* is not influenced by this option.

`/tcb/shrink break goal=`⟨*length*⟩ (no default, initially `0pt`)

This is an emergency parameter if the break algorithm produces unpleasant breaks. It shrinks the goal height of the current box part by ⟨*length*⟩ which may result in smaller boxes. Never use negative values. *Usually, this option will never be needed at all.*

17.4 Option Keys for the Break Appearance

/tcb/toprule at break=⟨*length*⟩ (no default, initially `0.5mm`)
Sets the line width of the top rule to ⟨*length*⟩ *if* the box is /tcb/breakable → P. 351. In this case, it is applied to *middle* and *last* parts in a break sequence. Note that /tcb/toprule → P. 35 overwrites this value if used afterwards.

/tcb/bottomrule at break=⟨*length*⟩ (no default, initially `0.5mm`)
Sets the line width of the bottom rule to ⟨*length*⟩ *if* the box is /tcb/breakable → P. 351. In this case, it is applied to *first* and *middle* parts in a break sequence. Note that /tcb/bottomrule → P. 35 overwrites this value if used afterwards.

/tcb/topsep at break=⟨*length*⟩ (no default, initially `0mm`)
Additional vertical space of ⟨*length*⟩ which is added at the top of *middle* and *last* parts in a break sequence. In general, it is not advisable to change this value if these parts start with a rule or a title.

/tcb/bottomsep at break=⟨*length*⟩ (no default, initially `0mm`)
Additional vertical space of ⟨*length*⟩ which is added at the bottom of *first* and *middle* parts in a break sequence. In general, it is not advisable to change this value if these parts end with a rule.

/tcb/pad before break=⟨*length*⟩ (style, no default, initially `3.5mm`)
Sets the total amount of vertical space after the text content and before the break point to ⟨*length*⟩. This style sets /tcb/toprule at break to `0pt` and changes /tcb/topsep at break as required. In general, it is not advisable to change this value if the *middle* and *last* parts in a break sequence start with a rule or a title.

/tcb/pad before break*=⟨*length*⟩ (style, no default)
Sets /tcb/pad before break to ⟨*length*⟩ and /tcb/enlargepage flexible → P. 354 to an appropriate value such that empty closing frames are avoided.

/tcb/pad after break=⟨*length*⟩ (style, no default, initially `3.5mm`)
Sets the total amount of vertical space after the break point and before the text content to ⟨*length*⟩. This style sets /tcb/bottomrule at break to `0pt` and changes /tcb/bottomsep at break as required. In general, it is not advisable to change this value if the *first* and *middle* parts in a break sequence end with a rule.

/tcb/pad at break=⟨*length*⟩ (style, no default, initially `3.5mm`)
Abbreviation for setting ⟨*length*⟩ to /tcb/pad before break and /tcb/pad after break.

/tcb/pad at break*=⟨*length*⟩ (style, no default)
Sets /tcb/pad at break to ⟨*length*⟩ and /tcb/enlargepage flexible → P. 354 to an appropriate value such that empty closing frames are avoided.

```
% \usepackage{lipsum}  % preamble
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced jigsaw,breakable,pad at break*=0mm,
  title={For this box, the pad space at the break point is set to 0mm}]
  \lipsum[1-2]
\end{tcolorbox}
```

For this box, the pad space at the break point is set to 0mm

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/pad at break`→ P. 355 or `/tcb/pad at break*`→ P. 355 should be used as very last option in an option list, because they adapt other settings.

Also see `/tcb/enlarge top at break by`→ P. 81 and `/tcb/enlarge bottom at break by`→ P. 81.

`/tcb/height fixed for=`⟨*part*⟩ (no default, initially `none`)

When certain amount of space is available for a partial box of a break sequence, the partial box typically is smaller than this space (depending on the box content). For given ⟨*part*⟩(s), the height can be set to all available space.

- `none`: Every partial `tcolorbox` is set with its natural height.
- `first`: The *first* partial box is set to a height which matches the available space.
- `middle`: All *middle* partial boxes are set to a height which matches the available space.
- `last`: The *last* partial box is set to a height which matches the available space.
- `first and middle`: The *first* and all *middle* partial boxes are set to a height which matches the available space.
- `middle and last`: All *middle* partial boxes and the *last* partial box are set to a height which matches the available space.
- `all`: All partial boxes are set to a height which matches the available space.

If the box keeps unbroken, this option is not applied. See `/tcb/height`→ P. 52 for setting a fixed height for unbroken boxes. See `/tcb/height fill`→ P. 55 for giving unbroken boxes maximum height.

`/tcb/vfill before first=true|false` (default `true`, initially `false`)

Inserts a `\vfill` at the begin of the *first* partial box to move this partial box to the end of the current page. This may be used as an alternative to `/tcb/height fixed for=first` to get justified columns or pages. The `\vfill` is not inserted, if the box gets not actually broken.

17.5 Extra Options for Partial Boxes

N 2015-07-16 **/tcb/extras**={⟨*options*⟩} (no default, initially unset)

Adds `tcolorbox` ⟨*options*⟩ to every box of a *break sequence* after skin settings are done. This is quite late in box processing. Geometry and break settings should *not be used* here, because they will either be ignored or have unexpected negative results. But it is possible to change most colors, skin effects, shadows, borders, frame code, etc. Note that using /tcb/extras for every box is very seldom an advantage over setting the options directly. Usually, /tcb/extras first, /tcb/extras middle, etc. are sensible to apply.

N 2015-07-16 **/tcb/no extras** (style, no default, initially set)

Removes all extras if set before.

N 2015-07-16 **/tcb/extras broken**={⟨*options*⟩} (no default, initially unset)

If the box is set to be /tcb/breakable → P. 351 and *is* broken actually, then the ⟨*options*⟩ are added to every box of the *break sequence*. /tcb/extras overwrites this key.

N 2015-07-16 **/tcb/extras unbroken**={⟨*options*⟩} (no default, initially unset)

If the box is set to be /tcb/breakable → P. 351 but *is not* broken actually or if the box is set to be /tcb/unbreakable → P. 352, then the ⟨*options*⟩ are added to the box. /tcb/extras overwrites this key.

N 2015-07-16 **/tcb/no extras unbroken** (style, no default, initially set)

Removes the unbroken extras if set before.

N 2015-07-16 **/tcb/extras first**={⟨*options*⟩} (no default, initially unset)

If the box is set to be /tcb/breakable → P. 351 and *is* broken actually, then the ⟨*options*⟩ are added to the *first* box of the break sequence. /tcb/extras overwrites this key.

N 2015-07-16 **/tcb/no extras first** (style, no default, initially set)

Removes the first extras if set before.

N 2015-07-16 **/tcb/extras middle**={⟨*options*⟩} (no default, initially unset)

If the box is set to be /tcb/breakable → P. 351 and *is* broken actually, then the ⟨*options*⟩ are added to every *middle* box (if any) of the break sequence. /tcb/extras overwrites this key.

N 2015-07-16 **/tcb/no extras middle** (style, no default, initially set)

Removes the middle extras if set before.

N 2015-07-16 **/tcb/extras last**={⟨*options*⟩} (no default, initially unset)

If the box is set to be /tcb/breakable → P. 351 and *is* broken actually, then the ⟨*options*⟩ are added to the *last* box of the break sequence. /tcb/extras overwrites this key.

N 2015-07-16 **/tcb/no extras last** (style, no default, initially set)

Removes the last extras if set before.

N 2015-07-16 **/tcb/extras unbroken and first**={⟨*options*⟩} (no default, initially unset)

This is an abbreviation for setting /tcb/extras unbroken and /tcb/extras first together. /tcb/extras overwrites this key.

N 2015-07-16 **/tcb/extras middle and last**={⟨*options*⟩} (no default, initially unset)

This is an abbreviation for setting /tcb/extras middle and /tcb/extras last together. /tcb/extras overwrites this key.

N 2015-07-16 **/tcb/extras unbroken and last**={⟨*options*⟩} (no default, initially unset)

This is an abbreviation for setting /tcb/extras unbroken and /tcb/extras last together. /tcb/extras overwrites this key.

N 2015-07-16 **/tcb/extras first and middle**={⟨*options*⟩} (no default, initially unset)

This is an abbreviation for setting /tcb/extras first and /tcb/extras middle together. /tcb/extras overwrites this key.

```
% \usepackage{lipsum,multicol}
% \usetikzlibrary{decorations.pathmorphing}
% \tcbuselibrary{skins}
\newtcolorbox{mybox}[1][]{
  tile,
  colback=green!7,coltitle=blue!50!black,colbacktitle=blue!5,
  center title,
  toprule=1.25mm,bottomrule=1.25mm,
  extras unbroken and first={
    borderline north={0.25mm}{0.5mm}{blue,decoration={zigzag,amplitude=0.5mm},decorate}},
  extras unbroken and last={
    borderline south={0.25mm}{0.5mm}{blue,decoration={zigzag,amplitude=0.5mm},decorate}},
  #1
}

\begin{mybox}[title=My unbroken box]
\lipsum[1]
\end{mybox}

\begin{multicols}{3}
  \begin{mybox}[title=My broken box,
    enforce breakable,% use only breakable in the real world!
    break at=4.2cm,pad at break=2mm,
    height fixed for=first and middle,  ]
  \lipsum[2]
  \end{mybox}
\end{multicols}
```

My unbroken box

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

My broken box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

17.6 Breakable boxes and the multicol package

Unbreakable tcolorboxes can be used without special care inside a multicols environment from the multicol package [9].
Since version 3.10, a breakable tcolorbox detects, if it is used inside a multicols environment. But choosing break points for a breakable box cannot be done by the balancing routine of multicols. By default, boxes will break at \textheight. To get pleasant results, use the /tcb/break at → P. 353 and /tcb/height fixed for → P. 356 options.

```
% \usepackage{lipsum,multicol}  % preamble
\small
\begin{multicols}{2}
  \lipsum[1]
  \begin{tcolorbox}[enhanced jigsaw,breakable,size=title,
    colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
    title=My breakable box,pad at break=1mm, break at=7.5cm/0pt ]
  \lipsum[2-4]
  \end{tcolorbox}
  \lipsum[4]
\end{multicols}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

My breakable box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.
Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

This example is already set inside a `multicols` environment. This time, a *middle* part has full `\textheight`. `/tcb/height fixed for` → P. 356 is used to spread this box part over the full height toi align with neighboring columns.

```
% \usepackage{lipsum,multicol}
\lipsum[1]
\begin{tcolorbox}[enhanced ↴
      ↳jigsaw,breakable,
  size=title,
  colback=red!5!white,
  colframe=red!75!black,
  fonttitle=\bfseries,
  title=My breakable box,
  pad at break=2mm,
  break at=8.2cm/0pt,
  height fixed for=middle ]
\lipsum[2-7]
\end{tcolorbox}
\lipsum[8]
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

My breakable box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi

quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

The following example has a `\tcolorbox` which fills the `\multicols` environment completely. Here, `/tcb/height fixed for` → P. 356 is used to give all three columns the full height. Note that the appropriate `/tcb/break at` → P. 353 value is not computed automatically but set manually.

```
% \usepackage{lipsum,multicol}  % preamble
\small
\begin{multicols}{3}
  \begin{tcolorbox}[enhanced jigsaw,breakable,size=small,
    colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
    title=My breakable box,pad at break=2mm,drop fuzzy shadow,
    height fixed for=all, break at=11.4cm ]
  \lipsum[1-3]
  \end{tcolorbox}
\end{multicols}
```

My breakable box

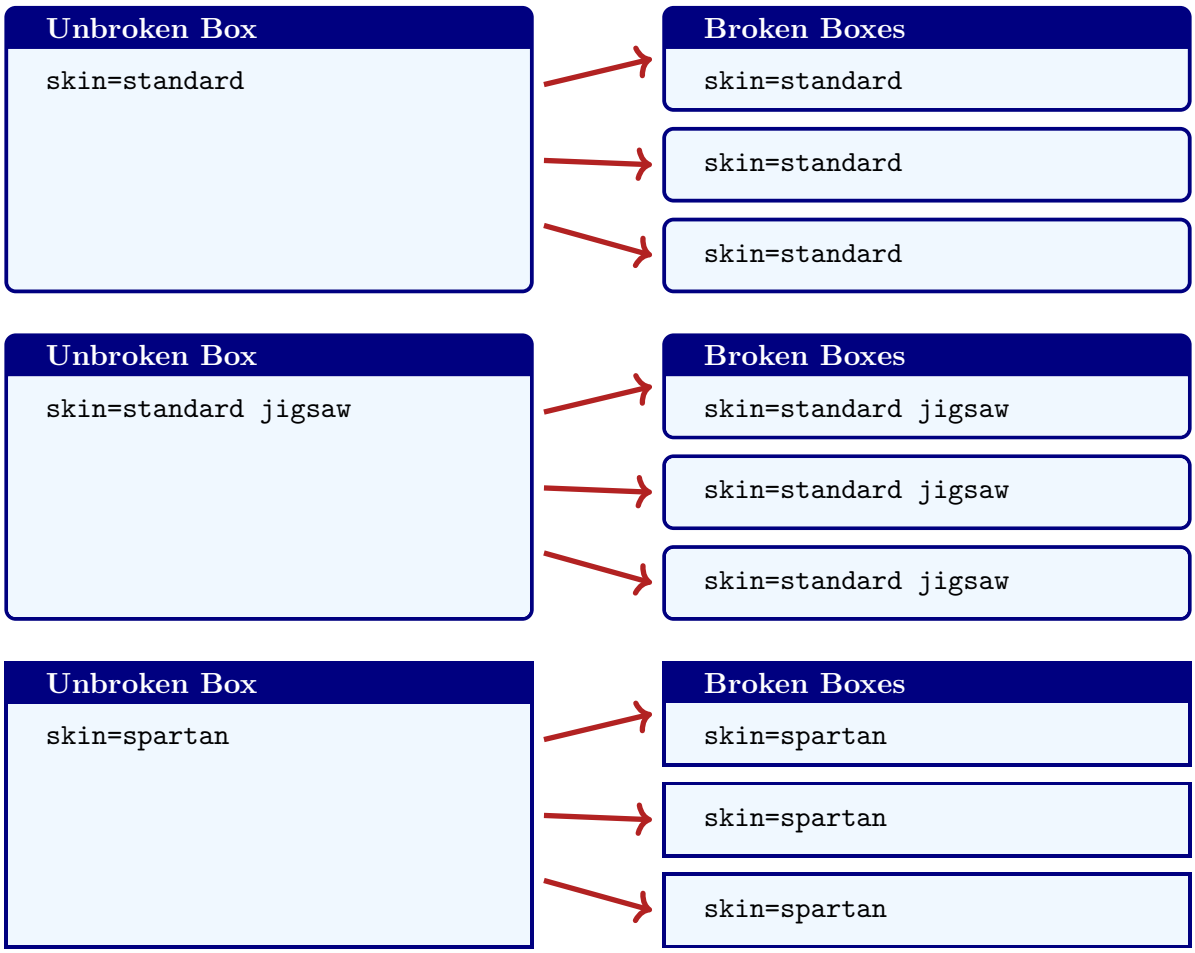
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

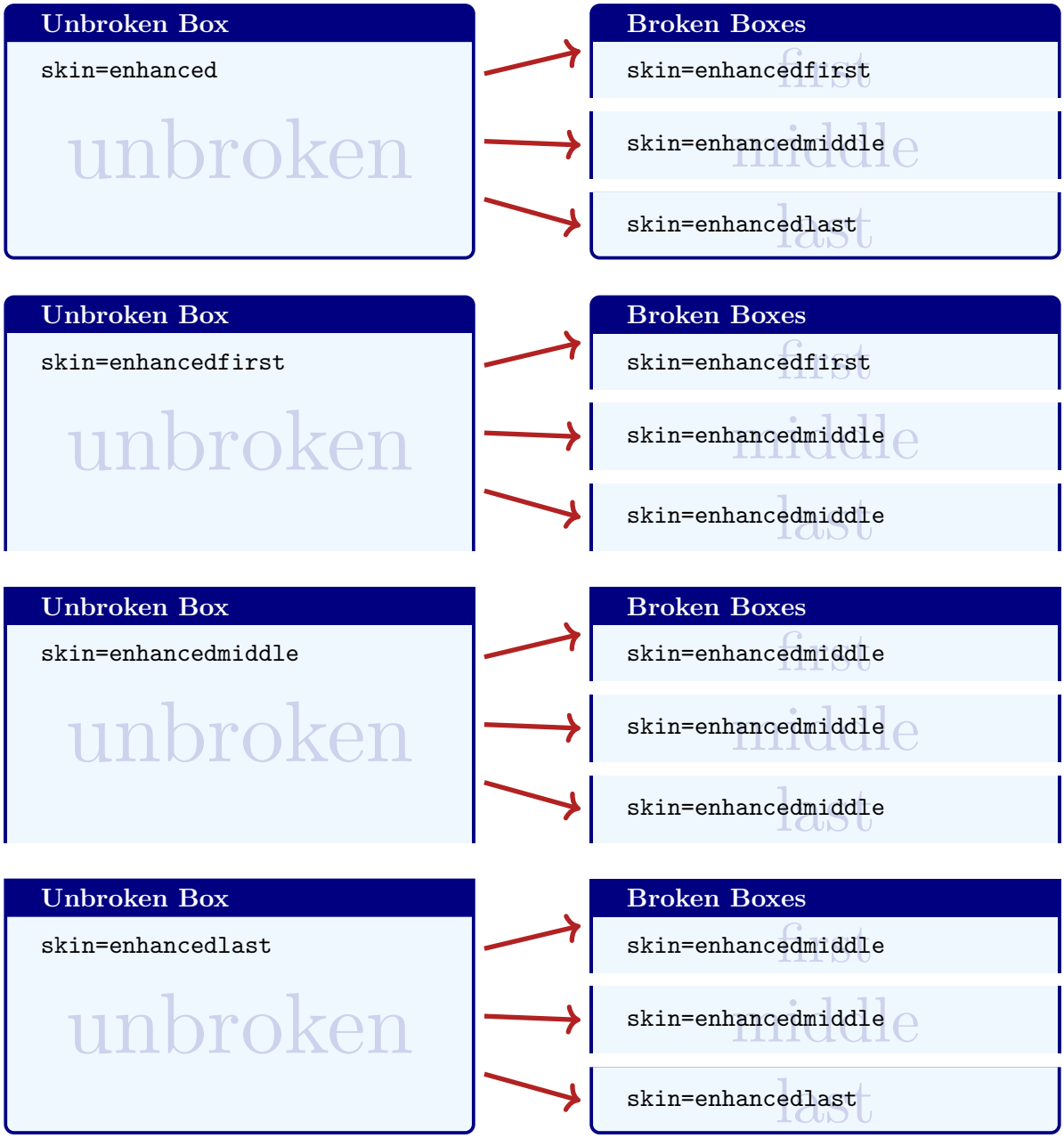
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

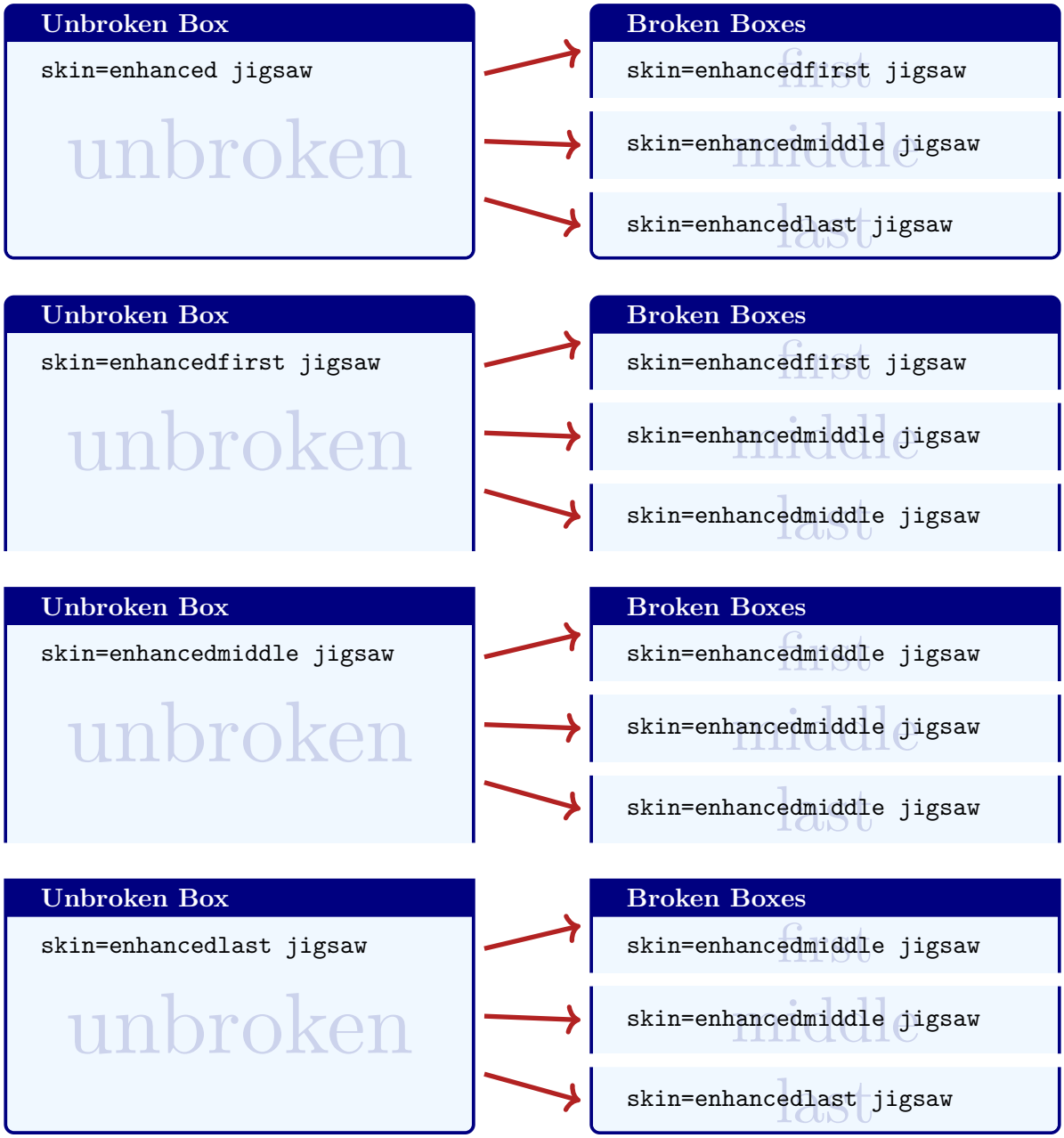
Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

17.7 Break Sequence for the Skins

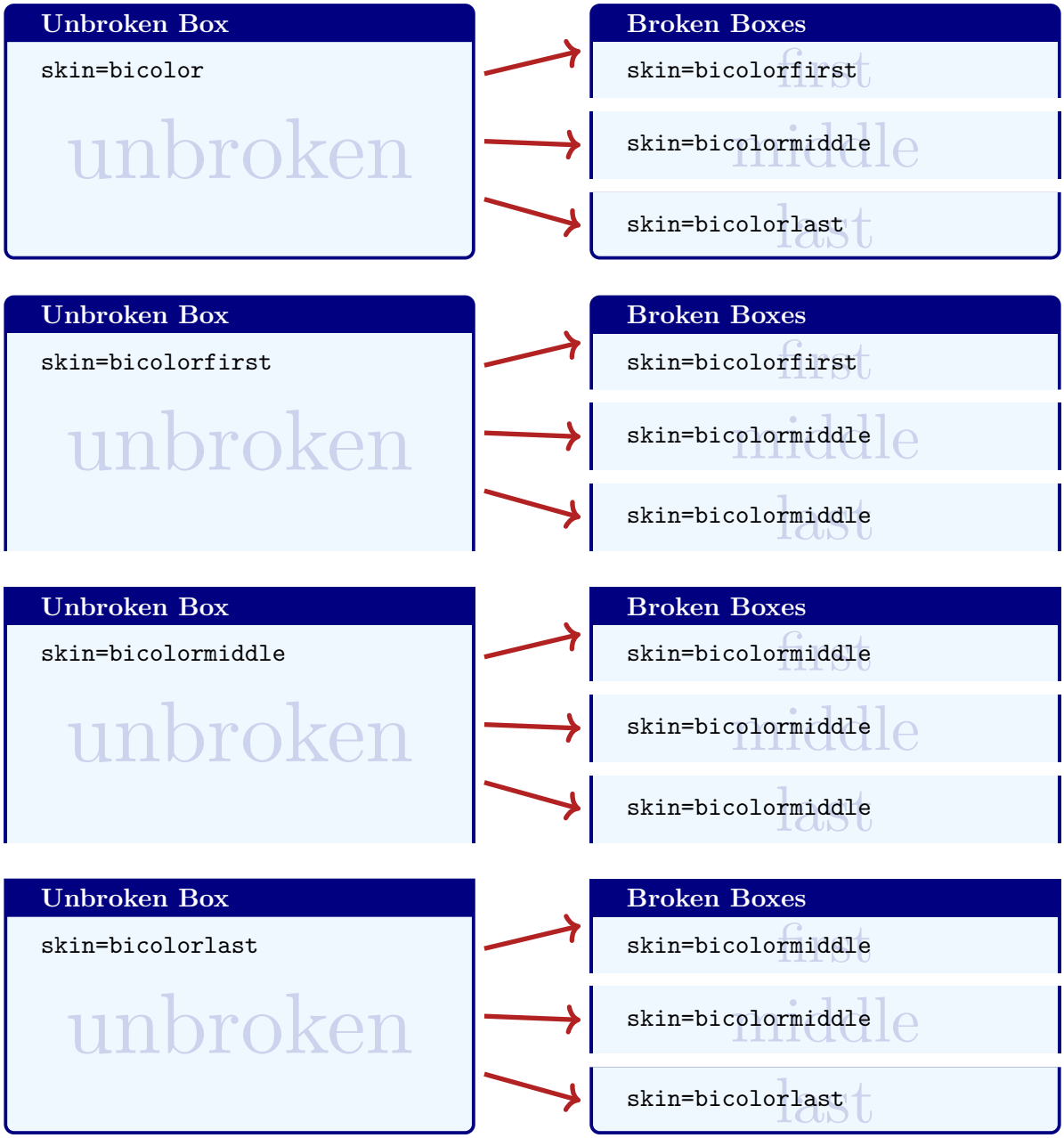
The following diagrams document the *break sequence* for different skins. Depending on the main skin of a `tcolorbox`, the actual skins of the *break sequence* parts are displayed.

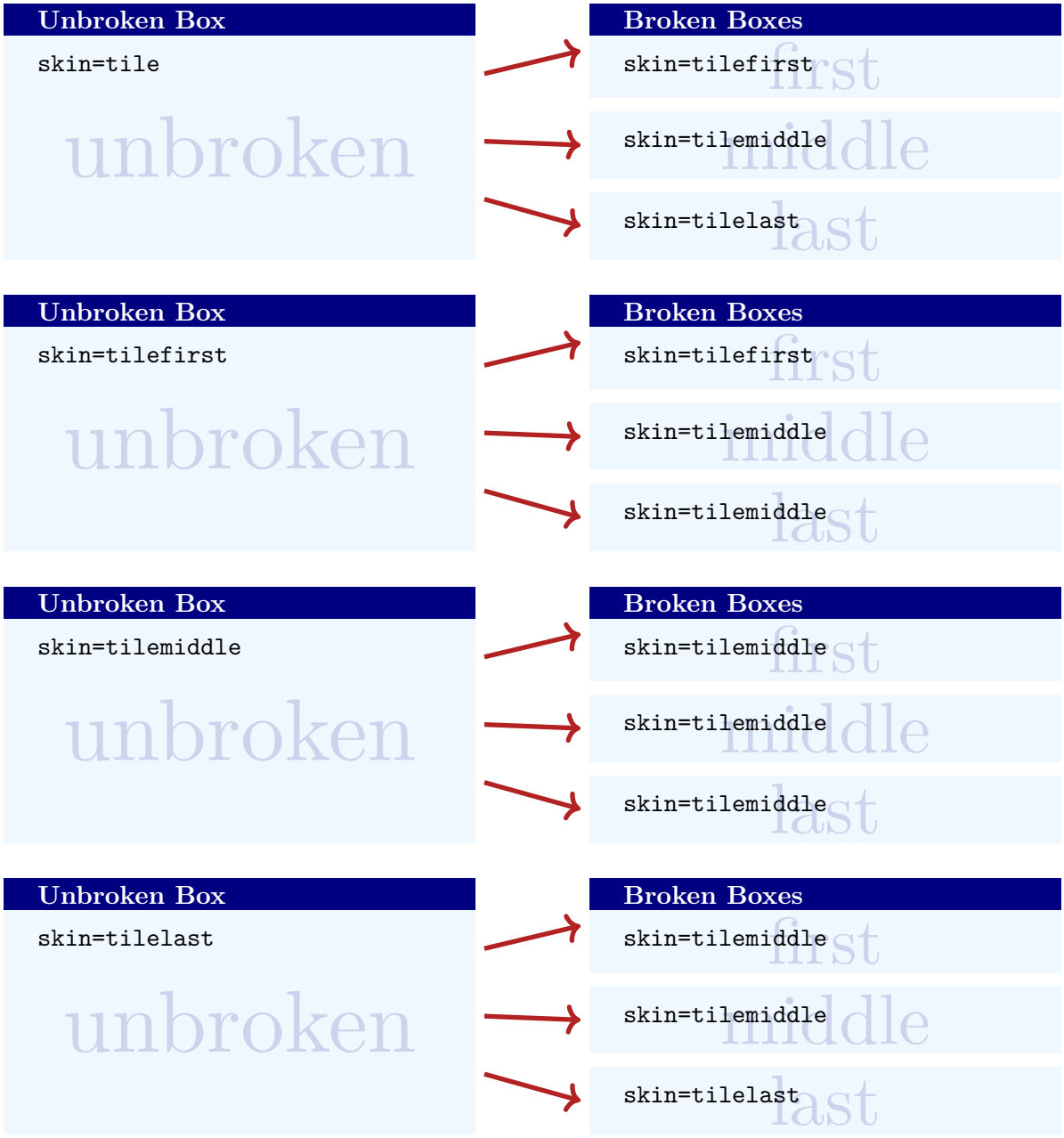


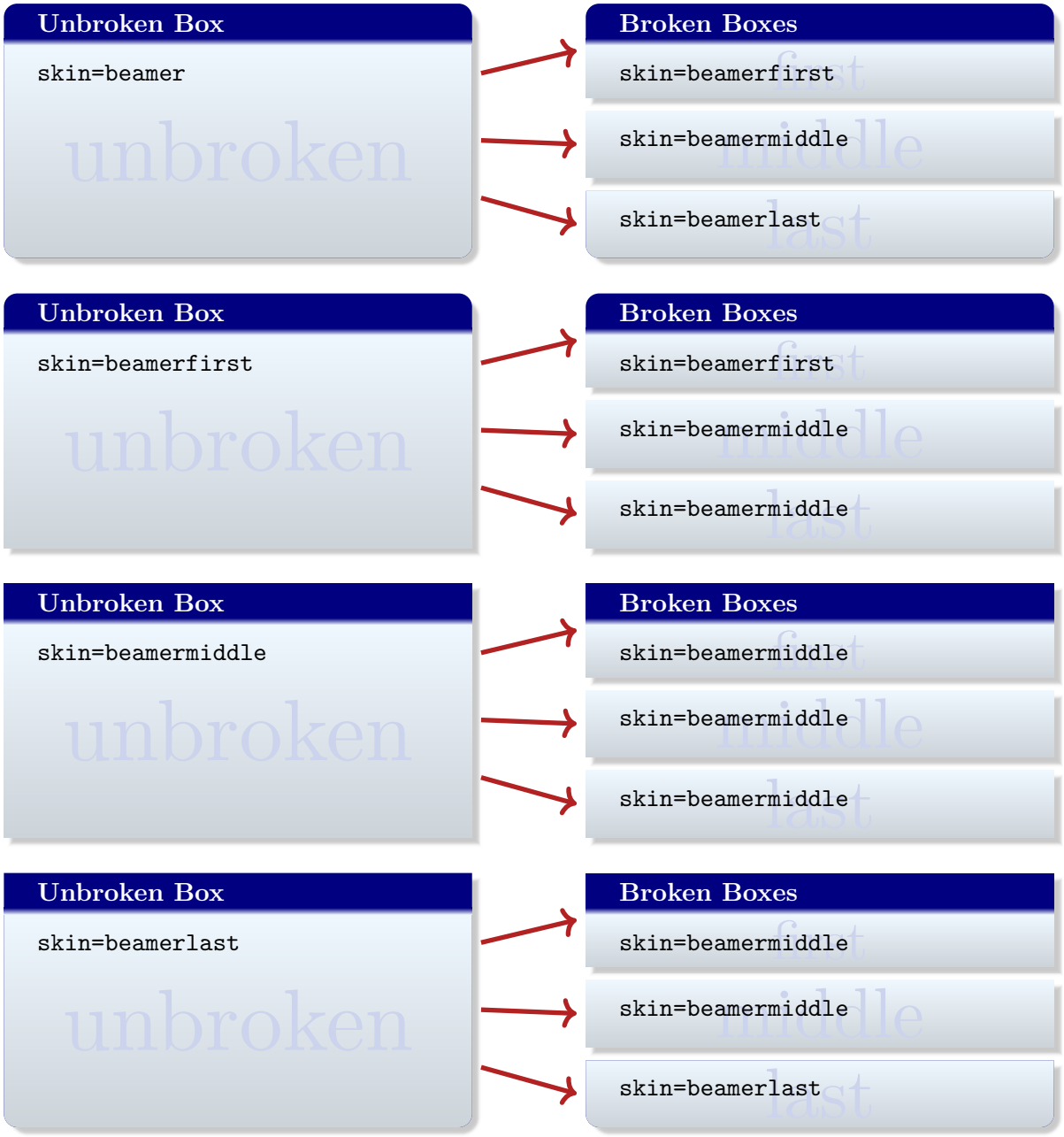

Unbroken Box
skin=enhanced
unbroken
Broken Boxes
skin=enhancedfirst
first
skin=enhancedmiddle
middle
skin=enhancedlast
last
Unbroken Box
skin=enhancedfirst
unbroken
Broken Boxes
skin=enhancedfirst
first
skin=enhancedmiddle
middle
skin=enhancedmiddle
last
Unbroken Box
skin=enhancedmiddle
unbroken
Broken Boxes
skin=enhancedmiddle
first
skin=enhancedmiddle
middle
skin=enhancedmiddle
last
Unbroken Box
skin=enhancedlast
unbroken
Broken Boxes
skin=enhancedmiddle
first
skin=enhancedmiddle
middle
skin=enhancedlast
last

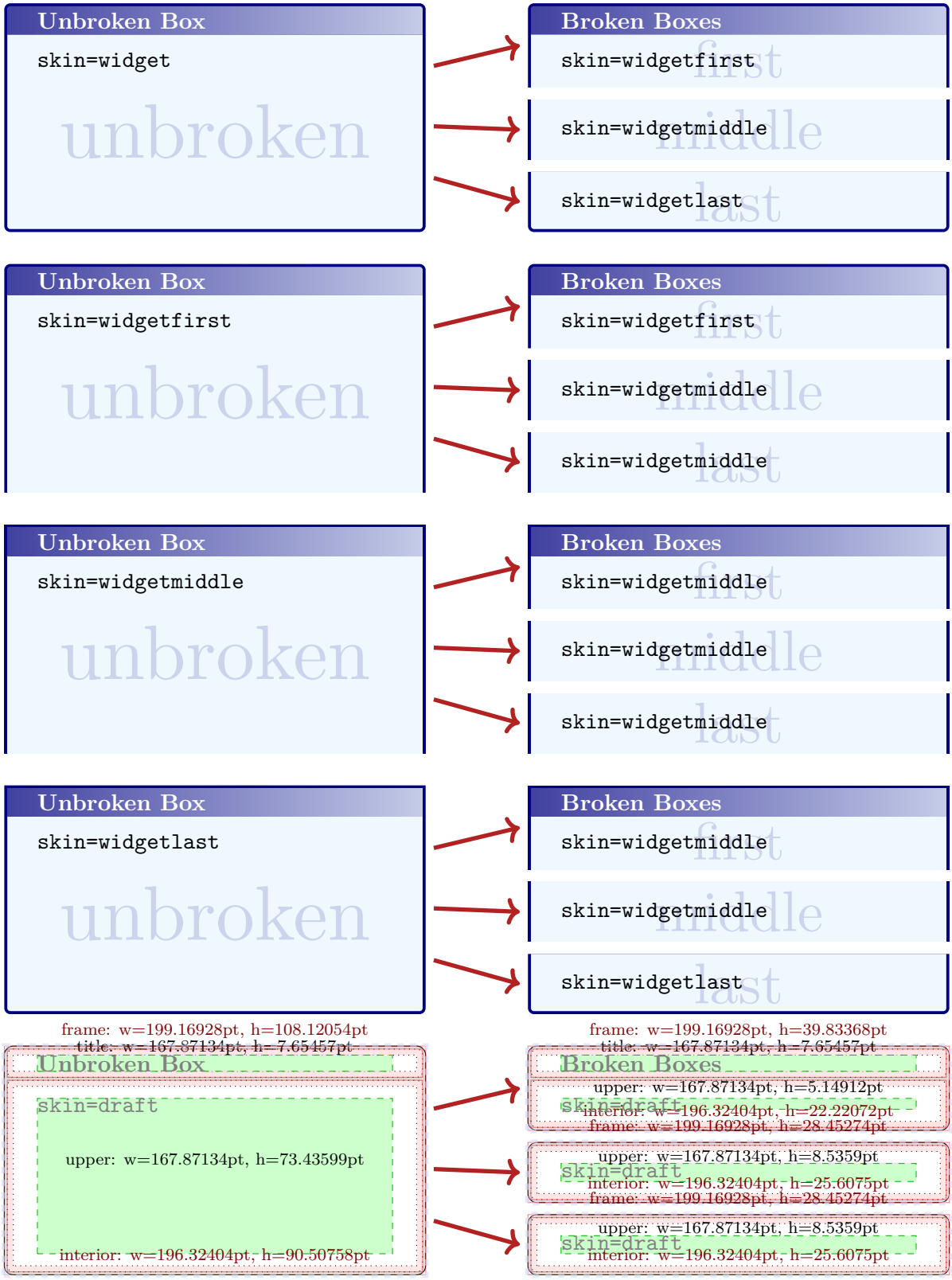

Unbroken Box
skin=enhanced jigsaw
unbroken
Broken Boxes
skin=enhancedfirst jigsaw
skin=enhancedmiddle jigsaw
skin=enhancedlast jigsaw
Unbroken Box
skin=enhancedfirst jigsaw
unbroken
Broken Boxes
skin=enhancedfirst jigsaw
skin=enhancedmiddle jigsaw
skin=enhancedmiddle jigsaw
Unbroken Box
skin=enhancedmiddle jigsaw
unbroken
Broken Boxes
skin=enhancedmiddle jigsaw
skin=enhancedmiddle jigsaw
skin=enhancedmiddle jigsaw
Unbroken Box
skin=enhancedlast jigsaw
unbroken
Broken Boxes
skin=enhancedmiddle jigsaw
skin=enhancedmiddle jigsaw
skin=enhancedlast jigsaw

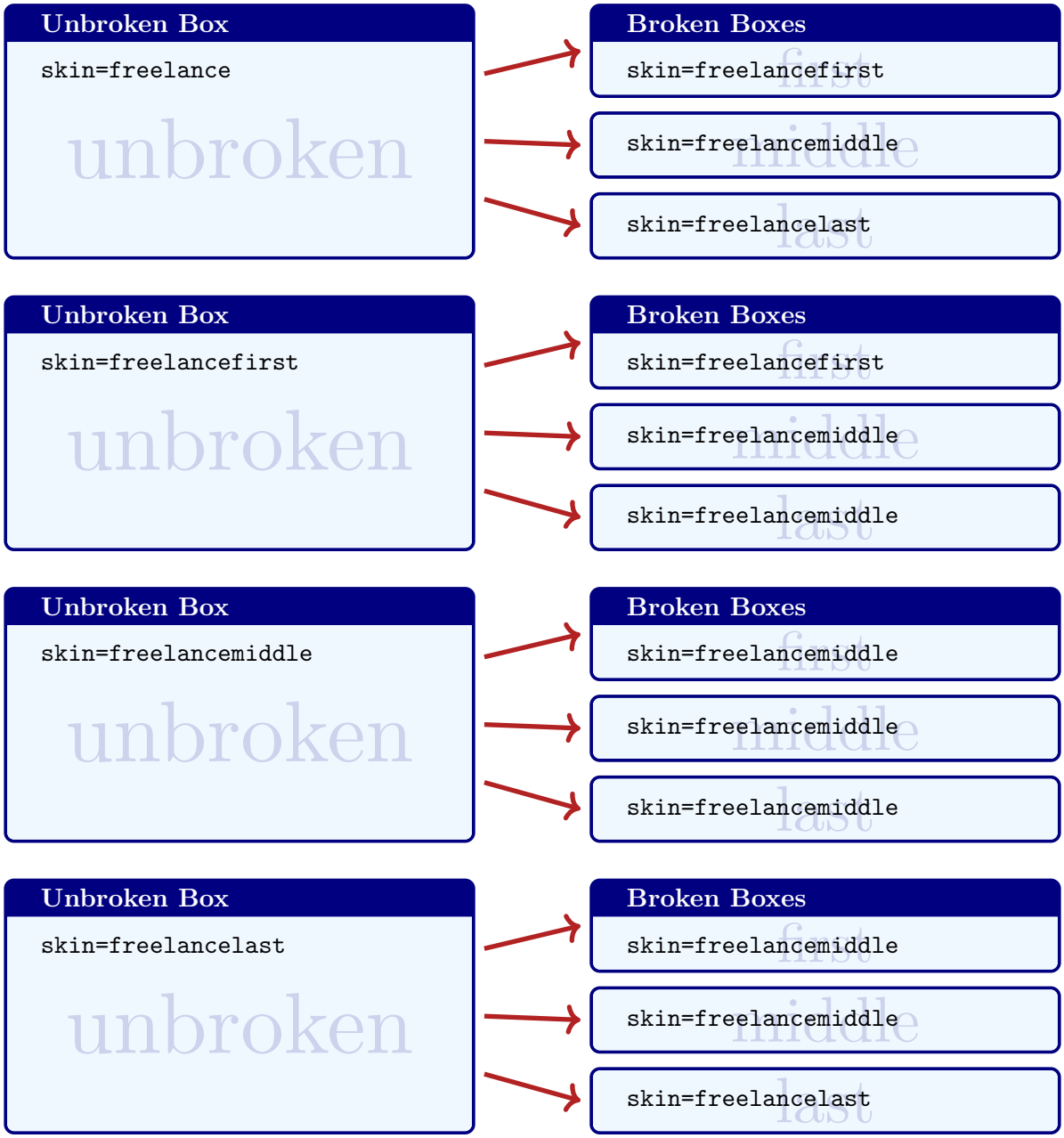
skin=empty
unbroken
skin=emptyfirst
first
skin=emptymiddle
middle
skin=emptylast
last
skin=emptyfirst
unbroken
skin=emptyfirst
first
skin=emptymiddle
middle
skin=emptymiddle
last
skin=emptymiddle
unbroken
skin=emptymiddle
first
skin=emptymiddle
middle
skin=emptymiddle
last
skin=emptylast
unbroken
skin=emptymiddle
first
skin=emptymiddle
middle
skin=emptylast
last


Unbroken Box
skin=bicolor
unbroken
Broken Boxes
skin=bicolorfirst
first
skin=bicolormiddle
middle
skin=bicolorlast
last
Unbroken Box
skin=bicolorfirst
unbroken
Broken Boxes
skin=bicolorfirst
first
skin=bicolormiddle
middle
skin=bicolormiddle
last
Unbroken Box
skin=bicolormiddle
unbroken
Broken Boxes
skin=bicolormiddle
first
skin=bicolormiddle
middle
skin=bicolormiddle
last
Unbroken Box
skin=bicolorlast
unbroken
Broken Boxes
skin=bicolormiddle
first
skin=bicolormiddle
middle
skin=bicolorlast
last


Unbroken Box
skin=tile
unbroken
Broken Boxes
skin=tilefirst
first
skin=tilemiddle
middle
skin=tilelast
last
Unbroken Box
skin=tilefirst
unbroken
Broken Boxes
skin=tilefirst
first
skin=tilemiddle
middle
skin=tilemiddle
last
Unbroken Box
skin=tilemiddle
unbroken
Broken Boxes
skin=tilemiddle
first
skin=tilemiddle
middle
skin=tilemiddle
last
Unbroken Box
skin=tilelast
unbroken
Broken Boxes
skin=tilemiddle
first
skin=tilemiddle
middle
skin=tilelast
last


Unbroken Box
skin=beamer
unbroken
Broken Boxes
skin=beamerfirst
first
skin=beamermiddle
middle
skin=beamerlast
last
Unbroken Box
skin=beamerfirst
unbroken
Broken Boxes
skin=beamerfirst
first
skin=beamermiddle
middle
skin=beamermiddle
last
Unbroken Box
skin=beamermiddle
unbroken
Broken Boxes
skin=beamermiddle
first
skin=beamermiddle
middle
skin=beamermiddle
last
Unbroken Box
skin=beamerlast
unbroken
Broken Boxes
skin=beamermiddle
first
skin=beamermiddle
middle
skin=beamerlast
last


Unbroken Box
skin=widget
unbroken
Broken Boxes
skin=widgetfirst
first
skin=widgetmiddle
middle
skin=widgetlast
last
Unbroken Box
skin=widgetfirst
unbroken
Broken Boxes
skin=widgetfirst
first
skin=widgetmiddle
middle
skin=widgetmiddle
last
Unbroken Box
skin=widgetmiddle
unbroken
Broken Boxes
skin=widgetmiddle
first
skin=widgetmiddle
middle
skin=widgetmiddle
last
Unbroken Box
skin=widgetlast
unbroken
Broken Boxes
skin=widgetmiddle
first
skin=widgetmiddle
middle
skin=widgetlast
last
frame: w=199.16928pt, h=108.12054pt
title: w=167.87134pt, h=7.65457pt
Unbroken Box
skin=draft
upper: w=167.87134pt, h=73.43599pt
interior: w=196.32404pt, h=90.50758pt
frame: w=199.16928pt, h=39.83368pt
title: w=167.87134pt, h=7.65457pt
Broken Boxes
upper: w=167.87134pt, h=5.14912pt
skin=draft
interior: w=196.32404pt, h=22.22072pt
frame: w=199.16928pt, h=28.45274pt
upper: w=167.87134pt, h=8.5359pt
skin=draft
interior: w=196.32404pt, h=25.6075pt
frame: w=199.16928pt, h=28.45274pt
upper: w=167.87134pt, h=8.5359pt
skin=draft
interior: w=196.32404pt, h=25.6075pt


Unbroken Box
skin=freelance
unbroken
Broken Boxes
skin=freelancefirst
first
skin=freelancemiddle
middle
skin=freelancelast
last
Unbroken Box
skin=freelancefirst
unbroken
Broken Boxes
skin=freelancefirst
first
skin=freelancemiddle
middle
skin=freelancemiddle
last
Unbroken Box
skin=freelancemiddle
unbroken
Broken Boxes
skin=freelancemiddle
first
skin=freelancemiddle
middle
skin=freelancemiddle
last
Unbroken Box
skin=freelancelast
unbroken
Broken Boxes
skin=freelancemiddle
first
skin=freelancemiddle
middle
skin=freelancelast
last

17.8 Break by Hand (Faked Break)

See Section 17.6 on page 359 for *real* column breaks.

Since the appearance of broken boxes is done by skins, it is quite easy to 'fake a break'. For this, you actually don't need the LIB breakable library at all.

```
\tcbset{enhanced,equal height group=fakedbreak,
  colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-6mm)/3,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}[title=My broken box,skin=enhancedfirst]
This is a box which breaks from one column to another
\end{tcolorbox}\hfill
\begin{tcolorbox}[skin=enhancedmiddle]
column. I am sorry to say that this is a trick.
Nevertheless, you may use this trick for your
\end{tcolorbox}\hfill
\begin{tcolorbox}[skin=enhancedlast]
own purposes.
\end{tcolorbox}
```

My broken box

This is a box which breaks from one column to another

column. I am sorry to say that this is a trick. Nevertheless, you may use this trick for your

own purposes.

18 Library magazine

The main purpose of this library is to store a `tcolorbox` into an array of box registers for later usage.

If the `tcolorbox` is not breakable, there is not much add-on compared to usual TEX/LATEX box storage and usage (and you do not really need this library for that use case). For a breakable `tcolorbox`, this library allows to capture all partial boxes into a sequence of registers. The partial boxes can be used anywhere in arbitrary order.

Example Article

This is an example for an article which starts right here and is continued to the following pages. The body text for the article is written inside a single `tcolorbox`. This box is split into parts using the tools from this section,

— *continued on page 375* —

The name of this library indicates *magazine* in the sense of storage, but also in the sense of a journal where an article often is *continued on page x*. An example for this kind of application is given throughout this section starting on the right hand side. The creation of this library was motivated by Ulrike Fischer and Steven B. Segletes.

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{magazine}
```

This also loads the library breakable, see Section 17 on page 349.

> **!** The box register operations of this library are global. TEX grouping will not clear the registers when leaving the current group. Also be aware that extensive use of large box arrays may eat up TEX's available memory and registers.

18.1 Creation and Resetting of Box Arrays

N 2015-07-13 `\newboxarray{`⟨*name*⟩`}`

This creates a new box array called ⟨*name*⟩. There already is a box array available with name `default` which can be used directly. Note that the creation is a global operation.

```
\newboxarray{myarray}
```

N 2015-07-13 `\boxarrayreset[`⟨*name*⟩`]`

Resets the size counter of a box array ⟨*name*⟩ to zero. If ⟨*name*⟩ is not provided, `default` is used as name. Use this or `/tcb/reset box array` before you apply `/tcb/store to box array` → P. 373. Otherwise, all boxes would be appended to the already existing boxes. This command does not clear box registers.

```
\boxarrayreset           % resets 'default'
\boxarrayreset{myarray}  % resets 'myarray'
```

N 2015-07-13 `/tcb/reset box array=`⟨*name*⟩ (default `default`, initially unset)

Resets the size counter of a box array ⟨*name*⟩ to zero. Use this or `\boxarrayreset` (which does the same) before you apply `/tcb/store to box array` → P. 373.

```
\tcbset{
  reset box array,          % resets 'default'
  reset box array=myarray, % resets 'myarray'
}
```

N 2015-07-13 `\boxarrayclear[⟨name⟩]`

Works like `\boxarrayreset` → P. 372 to reset the size counter of a box array ⟨*name*⟩ to zero. Additionally, all allocated box registers of the box array are cleared of their content. Note that the allocated box registers stay allocated. So, this may be useful to clear memory, but not to free registers for other applications. If `\consumeboxarray` → P. 376 or `\consumetcboxarray` → P. 376 was used to apply the stored boxes, there is no advantage in using `\boxarrayclear`.

```
\boxarrayclear            % clears 'default'
\boxarrayclear{myarray}   % clears 'myarray'
```

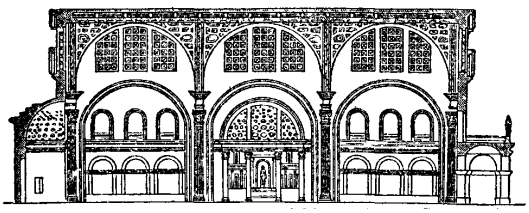
18.2 Storing Content

N 2015-07-13 `/tcb/store to box array=`⟨*name*⟩ (default `default`, initially unset)

Stores a `tcolorbox` or all parts of a break sequence of a `tcolorbox` into a box array ⟨*name*⟩. If no ⟨*name*⟩ is given, the already existing `default` box array is used. Otherwise, the box array has to be created beforehand with `\newboxarray` → P. 372. Note that the box has to be `/tcb/breakable` → P. 351, if the box shall break into several parts. Typically, manual break points are additionally defined by `/tcb/break at` → P. 353. Otherwise, the box parts will have a length of about `\textheight`. For most use cases, a `/tcb/reset box array` → P. 372 should be applied to reset the box array counter.

```
% \usepackage{lipsum}
\begin{tcolorbox}[enhanced jigsaw,size=fbox,width=4cm,
  colback=yellow!10,colframe=yellow!10!black,
  enforce breakable,% use only breakable in the real world!
  break at=7cm/4cm,
  height fixed for=all,
  watermark text=\arabic{tcbbreakpart},
  reset box array,
  store to box array
  ]
\lipsum[1]
\end{tcolorbox}

\useboxarray{1}\hfill
\begin{tabular}[b]{cc}
\multicolumn{2}{c}{\includegraphics[width=7cm]{Basilica_5.png}}\\
\useboxarray{2} & \useboxarray{3}
\end{tabular}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhon-



cus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

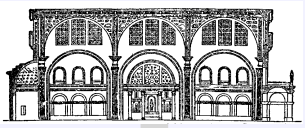
If the first box part should fill the rest of the available space of the current page, you can use `\pagegoal-\pagetotal` minus some distance for the first element of `/tcb/break at` → P. 353. You may want to have some additional distance to the preceeding text.

```
% \usepackage{lipsum}
\begin{tcolorbox}[enhanced,breakable,
  reset box array,
  store to box array,
  break at=\pagegoal-\pagetotal-5mm/0pt,
  height fixed for=first and middle]
\lipsum[1-15]
\end{tcolorbox}%
%
\consumetcboxarray{1}{blanker,before=\par\vfill\noindent}
```

```
\begin{tcolorbox}[blanker,width=4cm,
  fontupper=\footnotesize,
  enforce breakable,% use only breakable in the real world!
  break at=4cm,
  height fixed for=all,
  watermark text=\arabic{tcbbreakpart},
  reset box array,
  store to box array
  ]
\includegraphics[width=\linewidth]{Basilica_5.png}\par
\lipsum[1-2]
\end{tcolorbox}

\begin{tcbitemize}[raster columns=3,raster equal height,
  size=small,halign=center,sharp corners,colback=blue!5]
\tcbitem\consumeboxarray{5}
\tcbitem\consumeboxarray{6}
\tcbitem\consumeboxarray{1}
\tcbitem\consumeboxarray{2}
\tcbitem\consumeboxarray{3}
\tcbitem\consumeboxarray{4}
\end{tcbitemize}
```

lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec

ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.



Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam

arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus

vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at,

mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum. Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor

N 2015-07-13 **/tcb/reset and store to box array=**⟨*name*⟩ (style, default **default**, initially unset)

Combination of /tcb/reset box array → P. 372 and /tcb/store to box array → P. 373.

N 2015-07-13 **/tcb/do not store to box array** (style, no default, initially set)

Disables the /tcb/store to box array → P. 373 option, if set before.

N 2015-07-13 **\begin{boxarraystore}**{⟨*name*⟩}
⟨*environment content*⟩
\end{boxarraystore}

Stores the environment content into a box array ⟨*name*⟩. This corresponds to the standard LaTeX environment **lrbox**, but the storage operation is global. As long as \boxarrayreset → P. 372 is not used, every new boxarraystore adds a further box to the array.

```
\boxarrayreset
\begin{boxarraystore}{default}\fbox{Mary}\end{boxarraystore}
\begin{boxarraystore}{default}\fbox{Had}\end{boxarraystore}
\begin{boxarraystore}{default}\fbox{a}\end{boxarraystore}
\begin{boxarraystore}{default}\fbox{Little}\end{boxarraystore}
\begin{boxarraystore}{default}\fbox{Lamb}\end{boxarraystore}
\useboxarray{5}\useboxarray{4}\useboxarray{3}\useboxarray{2}\useboxarray{1}\hfill
\useboxarray{1}\useboxarray{5}
```

Lamb Little a Had Mary Mary Lamb

18.3 Retrieving Content

N 2015-07-13 **\boxarraygetsize[**⟨*name*⟩**]{**⟨*macro*⟩**}**

Stores the current size of a box array ⟨*name*⟩ into a given ⟨*macro*⟩. If no ⟨*name*⟩ is given, the already existing **default** box array is used.

```
\boxarraygetsize{\mysize}
Current size of the default box array:
\mysize.
```

Current size of the default box array: 5.

— continued from page 372 —
namely /tcb/reset and store to box array → P. 375 with a new box array **myarticle** which was created by **\newboxarray{myarticle}**.
The resulting parts are distributed throughout this Section 18 on page 372 using \consumetcboxarray → P. 376 at
— continued on page 376 —

N 2015-07-13 **\useboxarray[**⟨*name*⟩**]{**⟨*index*⟩**}**

Typesets the box with the given ⟨*index*⟩ number from the box array ⟨*name*⟩. If no ⟨*name*⟩ is given, the already existing **default** box array is used. It is considered an error, if a not existing box array ⟨*name*⟩ is used. It is silently ignored, if the ⟨*index*⟩ is out of range. Note that \useboxarray corresponds to the standard **\usebox** macro, respectively, **\copy**.

```
\boxarraygetsize{\mysize}
\foreach \n in  {1,...,\mysize} { \useboxarray{\n} }
```

Mary Had a Little Lamb

N 2015-07-13 `\usetcboxarray[⟨name⟩]{⟨index⟩}{⟨options⟩}`

Typesets the box with the given ⟨*index*⟩ number from the box array ⟨*name*⟩ using `\useboxarray` → P. 375 as content of a `\tcbox` → P. 14. If no ⟨*name*⟩ is given, the already existing `default` box array is used. It is considered an error, if a not existing box array ⟨*name*⟩ is used. It is silently ignored, if the ⟨*index*⟩ is out of range. The `\tcbox` → P. 14 can be customized by `tcolorbox` ⟨*options*⟩.

```
\boxarraygetsize{\mysize}
\foreach \n in  {1,...,\mysize} { \usetcboxarray{\n}{on line,colframe=yellow,
  colback=yellow!10} }
```

Mary Had a Little Lamb

N 2015-07-13 `\consumeboxarray[⟨name⟩]{⟨index⟩}`

Typesets the box with the given ⟨*index*⟩ number from the box array ⟨*name*⟩. If no ⟨*name*⟩ is given, the already existing `default` box array is used. It is considered an error, if a not existing box array ⟨*name*⟩ is used. It is silently ignored, if the ⟨*index*⟩ is out of range. In contrast to `\useboxarray` → P. 375, `\consumeboxarray` corresponds to the standard `\box` macro, i.e. after typesetting the box register is cleared and cannot be used again.

```
\boxarraygetsize{\mysize}
First run: \foreach \n in  {1,...,\mysize} { \consumeboxarray{\n} }
\par
Second run: \foreach \n in  {1,...,\mysize} { \consumeboxarray{\n} }
```

First run: Mary Had a Little Lamb
Second run:

N 2015-07-13 `\consumetcboxarray[⟨name⟩]{⟨index⟩}{⟨options⟩}`

Typesets the box with the given ⟨*index*⟩ number from the box array ⟨*name*⟩ using `\consumeboxarray` as content of a `\tcbox` → P. 14. If no ⟨*name*⟩ is given, the already existing `default` box array is used. It is considered an error, if a not existing box array ⟨*name*⟩ is used. It is silently ignored, if the ⟨*index*⟩ is out of range. The `\tcbox` → P. 14 can be customized by `tcolorbox` ⟨*options*⟩. After typesetting the box register is cleared and cannot be used again.

— continued from page 375 —
the appropriate places you see. The linking texts like *continued on page x* are created by `/tcb/finish` → P. 188 commands for the embedding `\tcbox` → P. 14. To label the box parts, `/tcb/phantomlabel` → P. 94 is used.
These quite small partial boxes are
— continued on page 379 —

```
% \usepackage{lipsum}
\begin{tcolorbox}[enhanced jigsaw,size=fbox,width=6cm,
  colback=yellow!10,colframe=yellow!10!black,
  enforce breakable,% use only breakable in the real world!
  break at=5cm,
  watermark text=\arabic{tcbbreakpart},
  reset and store to box array
  ]
\lipsum[1]
\end{tcolorbox}

\consumeboxarray{2} \hfill \consumeboxarray{1} \hfill \consumeboxarray{1}
```

lus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tel-

N 2015-07-13 **\boxarraygetbox**[⟨*name*⟩]{⟨*macro*⟩}{⟨*index*⟩}

Assigns the box with the given ⟨*index*⟩ number from the box array ⟨*name*⟩ to a ⟨*macro*⟩. If no ⟨*name*⟩ is given, the already existing `default` box array is used. It is considered an error, if a not existing box array ⟨*name*⟩ is used. If the ⟨*index*⟩ is out of range, the ⟨*macro*⟩ will be undefined.

```
\tcbox[size=small,colframe=blue!20,colback=yellow!5,on line,
  reset and store to box array]{Test}

\boxarraygetsize{\mysize} Array size: \mysize

\boxarraygetbox{\mybox}{1}
Box width: \the\wd\mybox
\quad\usebox{\mybox}
```

Array size: 1
Box width: 30.35799pt Test

18.4 Box Dimensions

N 2015-07-13 `\boxarraygetwidth[`⟨*name*⟩`]{`⟨*macro*⟩`}{`⟨*index*⟩`}`

Assigns the width of the box with the given ⟨*index*⟩ number from the box array ⟨*name*⟩ to a ⟨*macro*⟩. If no ⟨*name*⟩ is given, the already existing `default` box array is used. It is considered an error, if a not existing box array ⟨*name*⟩ is used. If the ⟨*index*⟩ is out of range, the ⟨*macro*⟩ will be set to `0pt`.

```
\tcbox[size=small,colframe=blue!20,colback=yellow!5,on line,
  reset and store to box array]{Test}

\begin{tabular}{ll}
\useboxarray{1} & width of box 1: \boxarraygetwidth{\mylen}{1} \mylen\\
\useboxarray{2} & width of box 2: \boxarraygetwidth{\mylen}{2} \mylen
\end{tabular}
```

Test width of box 1: 30.35799pt
width of box 2: 0pt

N 2015-07-13 `\boxarraygetheight[`⟨*name*⟩`]{`⟨*macro*⟩`}{`⟨*index*⟩`}`

Assigns the height of the box with the given ⟨*index*⟩ number from the box array ⟨*name*⟩ to a ⟨*macro*⟩. If no ⟨*name*⟩ is given, the already existing `default` box array is used. It is considered an error, if a not existing box array ⟨*name*⟩ is used. If the ⟨*index*⟩ is out of range, the ⟨*macro*⟩ will be set to `0pt`.

```
\tcbox[size=small,colframe=blue!20,colback=yellow!5,on line,
  reset and store to box array]{Test}

\begin{tabular}{ll}
\useboxarray{1} & height of box 1: \boxarraygetheight{\mylen}{1} \mylen\\
\useboxarray{2} & height of box 2: \boxarraygetheight{\mylen}{2} \mylen
\end{tabular}
```

Test height of box 1: 9.89883pt
height of box 2: 0pt

N 2015-07-13 `\boxarraygetdepth[`⟨*name*⟩`]{`⟨*macro*⟩`}{`⟨*index*⟩`}`

Assigns the depth of the box with the given ⟨*index*⟩ number from the box array ⟨*name*⟩ to a ⟨*macro*⟩. If no ⟨*name*⟩ is given, the already existing `default` box array is used. It is considered an error, if a not existing box array ⟨*name*⟩ is used. If the ⟨*index*⟩ is out of range, the ⟨*macro*⟩ will be set to `0pt`.

```
\tcbox[size=small,colframe=blue!20,colback=yellow!5,on line,
  reset and store to box array]{Test}

\begin{tabular}{ll}
\useboxarray{1} & depth of box 1: \boxarraygetdepth{\mylen}{1} \mylen\\
\useboxarray{2} & depth of box 2: \boxarraygetdepth{\mylen}{2} \mylen
\end{tabular}
```

Test depth of box 1: 3.69884pt
depth of box 2: 0pt

N 2015-07-13 `\boxarraygettotalheight[⟨name⟩]{⟨macro⟩}{⟨index⟩}`

Assigns the total height of the box with the given ⟨*index*⟩ number from the box array ⟨*name*⟩ to a ⟨*macro*⟩. If no ⟨*name*⟩ is given, the already existing `default` box array is used. It is considered an error, if a not existing box array ⟨*name*⟩ is used. If the ⟨*index*⟩ is out of range, the ⟨*macro*⟩ will be set to `0pt`.

— continued from page 376 —

for demonstration purposes. With the tools of this section, a magazine type document could be created, but this still needs a lot of manual control.

```
\boxarrayreset
\tcbox[size=small,colframe=blue!20,colback=yellow!5,on line,
  store to box array]{Test}

\begin{tabular}{ll}
\useboxarray{1} & total height of box 1: \boxarraygettotalheight{\mylen}{1} \mylen\\
\useboxarray{2} & total height of box 2: \boxarraygettotalheight{\mylen}{2} \mylen
\end{tabular}
```

Test total height of box 1: 13.59767pt
total height of box 2: 0pt

19 Library fitting

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{fitting}
```

19.1 Macros of the Library

`\tcboxfit[`⟨*options*⟩`]{`⟨*box content*⟩`}`

Creates a colored box where the given ⟨*box content*⟩ is fitted to the width and height of the box. A `tcboxfit` has to have a fixed height. If no fixed height is given, a square box is constructed. In principle, most ⟨*options*⟩ for a `tcolorbox` → P. 12 can be used for `\tcboxfit` with some restrictions. A `\tcboxfit` cannot have a lower part and cannot be broken.

```
% \usepackage{lipsum} \tcbuselibrary{raster}
\tcbset{colframe=blue!50!black,colback=red!10!white,
    boxsep=0pt,top=1mm,bottom=1mm,left=1mm,right=1mm,
    fit algorithm=hybrid*,raster equal skip=1mm}
\begin{tcbraster}[raster columns=3,raster valign=bottom]
  \tcboxfit[height=8cm]{\lipsum[1]}
  \tcboxfit[height=4cm]{\lipsum[1]}
  \tcboxfit[height=2cm]{\lipsum[1]}
\end{tcbraster}
\begin{tcbraster}[colback=green!10!white,boxsep=1mm]
  \tcboxfit[height=4cm]{\lipsum[2]}
  \tcboxfit[height=4cm,title=With a title]{\lipsum[2]}
\end{tcbraster}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

With a title

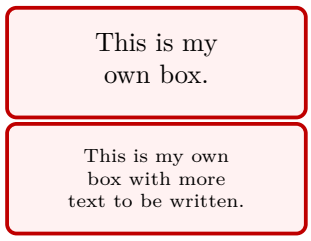
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

See Section 21.6 on page 413 for more elaborate methods to create new commands.

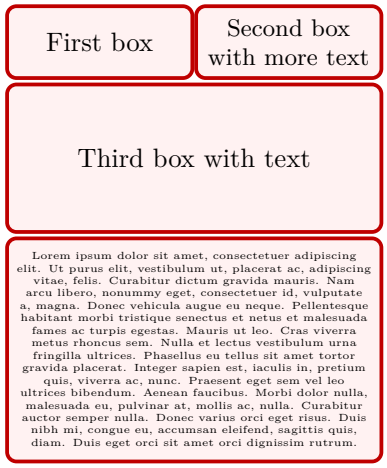
\newtcboxfit[⟨*init options*⟩]{\⟨*name*⟩}[⟨*number*⟩][⟨*default*⟩]{⟨*options*⟩}

Creates a new macro \⟨*name*⟩ based on \tcboxfit→ P. 380. Basically, **\newtcboxfit** operates like **\newcommand**. The new macro \⟨*name*⟩ optionally takes ⟨*number*⟩+1 arguments, where ⟨*default*⟩ is the default value for the optional first argument. The ⟨*options*⟩ are given to the underlying **tcboxfit**. The ⟨*init options*⟩ allow setting up automatic numbering, see Section 5 from page 100.

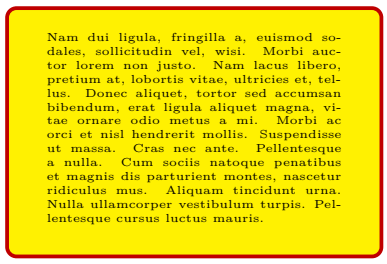
```
\newtcboxfit{\mybox}{colback=red!5!white,
  colframe=red!75!black,width=4cm,
  height=1.5cm,halign=center}

\mybox{This is my own box.}\par
\mybox{This is my own box with more text
       to be written.}
```




```
% \usepackage{lipsum}
\newtcboxfit{\mybox}[2]{colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries,
  boxsep=1mm,left=0mm,right=0mm,top=0mm,
  bottom=0mm,halign=center,valign=center,
  nobeforeafter,width=#1,height=#2}

\mybox{2.5cm}{1cm}{First box}%
\mybox{2.5cm}{1cm}{Second box with more text}\\
\mybox{5cm}{2cm}{Third box with text}\\
\mybox{5cm}{3cm}{\lipsum[1]}
```




```
% \usepackage{lipsum}
\newtcboxfit{\mybox}[2][]{colback=red!5!white,
  colframe=red!75!black,
  width=#2,height=#2/3*2,#1}

\mybox[colback=yellow]{5cm}%
  {\lipsum[2]}
```



\renewtcboxfit[⟨*init options*⟩]{\⟨*name*⟩}[⟨*number*⟩][⟨*default*⟩]{⟨*options*⟩}

Operates like \newtcboxfit, but based on **\renewcommand** instead of **\newcommand**. An existing macro is redefined.

\tcbfontsize{⟨*factor*⟩}

Selects a font size inside a tcolorbox which is scaled with the given ⟨*factor*⟩ relative to **\tcbfitdim**.

```
\tcbset{colback=red!5!white,colframe=red!75!black}
\begin{tcolorbox}[fit basedim=10pt]
  {\tcbfontsize{0.25} Very tiny,}\\
  {\tcbfontsize{0.5} Small,}\\
  {\tcbfontsize{1} Normal,}\\
  {\tcbfontsize{2} Large,}\\
  {\tcbfontsize{4} Huge.}
\end{tcolorbox}
```

19.2 Option Keys of the Library

The font size for the content of a box with fixed width and fixed height can be adjusted automatically. This is called the *fitbox capture mode*. Note that the fit control algorithm constructs a series of versions for the box and selects the 'best'. Therefore, the compilation time is quite longer than for a normal box. The algorithm will fail, if a different selected font size does not change the overall size of the box content. The `\tcboxfit` → P. 380 macro uses this algorithm by default.

> **!** The fit control keys are only applicable to unbreakable boxes without a lower part. The box content should not change counters.

`/tcb/fit` (style, initially unset)

Sets the `/tcb/capture` → P. 90 mode to `fitbox`, i. e. enables the font size adjustment algorithm. Thereby, a `tcolorbox` → P. 12 acts like `\tcboxfit` → P. 380 where the given ⟨*box content*⟩ is fitted to the width and height of the box. Therefore, the box has to have a fixed height. If no fixed height is given, a square box is constructed. The font dimension **`\tcbfitdim`** can also be used to adjust the margins of the box since a box with a tiny font may not need large margins. The number of constructed boxes is saved to the macro **`\tcbfitsteps`** for analysis.

```
% \usepackage{lipsum}
% \tcbuselibrary{skins}
\newtcolorbox{fitting}[2][]{fit,height=#2,boxsep=1pt,valign=center,opacityupper=0.5,
  top=0.4\tcbfitdim,bottom=0.4\tcbfitdim,left=0.75\tcbfitdim,right=0.75\tcbfitdim,
  enhanced,watermark text={\tcbfitsteps},colframe=blue!75!black,colback=white,#1}

\begin{fitting}{4cm}
\lipsum[1]
\end{fitting}

\begin{fitting}{2cm}
\lipsum[2]
\end{fitting}

\begin{fitting}{1cm}
\lipsum[3]
\end{fitting}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

/tcb/fit to=⟨*width*⟩ **and** ⟨*height*⟩ (style, initially unset)

Shortcut for using /tcb/fit → P. 382 and setting the ⟨*width*⟩ and ⟨*height*⟩ values separately.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[fit to=3cm and 2cm]
  This box content is fitted to the given
  dimensions.
\end{tcolorbox}
```

This box content is fitted to the given dimensions.

/tcb/fit to height=⟨*height*⟩ (style, initially unset)

Shortcut for using /tcb/fit → P. 382 and setting the ⟨*height*⟩ value separately.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[fit to height=2cm]
  This box content is fitted to the given
  height.
\end{tcolorbox}
```

This box content is fitted to the given height.

/tcb/fit basedim=⟨*length*⟩ (no default, initially 10pt)

Sets the starting font dimension for the font size adjustment algorithm to ⟨*length*⟩. The algorithm never enlarges this dimension.

```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[fit to=4cm and 2cm]
  Too few words for the box.
\end{tcolorbox}

\begin{tcolorbox}[fit to=4cm and 2cm,
  fit basedim=50pt]
  Enough words for the box.
\end{tcolorbox}
```

Too few words for the box.

Enough words for the box.

/tcb/fit skip=⟨*real value*⟩ (no default, initially 1.2)

Sets the skip value of the selected font to ⟨*real value*⟩ times \tcbfitdim.

```
% \usepackage{lipsum}
\tcbset{colback=red!5!white,
  colframe=red!75!black,left=1mm,
  right=1mm,boxsep=0mm}

\begin{tcolorbox}[fit to=5cm and 4cm,
  fit skip=1.0  ]
  \lipsum[1]
\end{tcolorbox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

`/tcb/fit fontsize macros` (style, initially unset)

Redefines the standard LaTeX font size macros `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge`, and `\Huge`, to set font sizes relative to the current `\tcbfitdim`. Note that the display skip values for mathematical formulas are respected by the redefined macros.

```
% \usepackage{lipsum}
\tcbset{colback=red!5!white,
  colframe=red!75!black,left=1mm,
  right=1mm,boxsep=0mm}

\begin{tcolorbox}[fit to height=4cm]
  {\Large\bfseries This text is
              not adapted:\par}
  \lipsum[2]
\end{tcolorbox}

\begin{tcolorbox}[fit to height=4cm,
  fit fontsize macros ]
  {\Large\bfseries This text is adapted:\par}
  \lipsum[2]
\end{tcolorbox}
```

This text is not adapted:

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

This text is adapted:

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
\tcbset{colback=red!5!white,
  colframe=red!75!black,left=1mm,
  right=1mm,boxsep=0mm}

\let\realHuge=\Huge

\begin{tcolorbox}[fit basedim=7pt,
  fontupper=\normalsize,
  fit fontsize macros]
The relative relative font size macros
are also usable without the
\textit{fit} algorithm.\par
{\Huge Adapted Huge} ---
{\realHuge Original Huge}
\end{tcolorbox}
```

The relative relative font size macros are also usable without the *fit* algorithm.

Adapted Huge — Original Huge

```
\tcbset{size=fbox,colback=red!5!white,
  colframe=red!75!black}

\tcboxfit[height=5cm,
  fit fontsize macros,
  fonttitle=\normalsize\bfseries,
  title=Adapted title]
{\lipsum[2]}
```

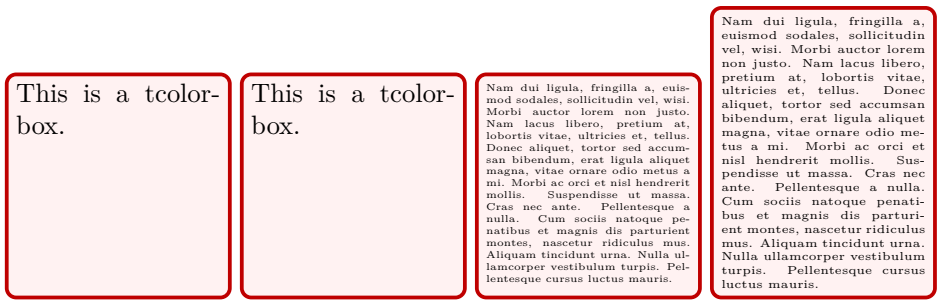
Adapted title

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/fit height plus=`⟨*dimension*⟩ (no default, initially `0pt`)

The box is allowed to enlarge the fixed height up to the given ⟨*dimension*⟩, before a font size fit is applied. An optional `/tcb/fit width plus` is tried after the height adaption.

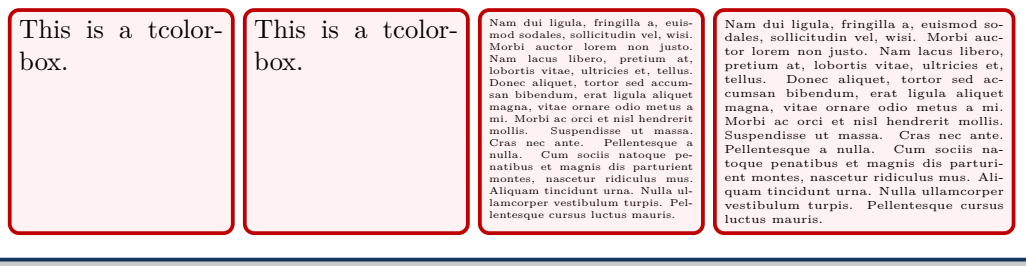
```
% \usepackage{lipsum}
\tcbset{colback=red!5!white,colframe=red!75!black,left=1mm,top=1mm,bottom=1mm,
  right=1mm,boxsep=0mm,width=3cm,height=3cm,nobeforeafter}

\begin{tcolorbox}[fit]
This is a tcolorbox.
\end{tcolorbox}
\begin{tcolorbox}[fit,fit height plus=1cm]
This is a tcolorbox.
\end{tcolorbox}
\begin{tcolorbox}[fit]
\lipsum[2]
\end{tcolorbox}
\begin{tcolorbox}[fit,fit height plus=1cm]
\lipsum[2]
\end{tcolorbox}
```



`/tcb/fit width plus=`⟨*dimension*⟩ (no default, initially `0pt`)

The box is allowed to enlarge the fixed width up to the given ⟨*dimension*⟩, before a font size fit is applied. An optional `/tcb/fit height plus` is tried before the width adaption.

```
% \usepackage{lipsum}
\tcbset{colback=red!5!white,colframe=red!75!black,left=1mm,top=1mm,bottom=1mm,
  right=1mm,boxsep=0mm,width=3cm,height=3cm,nobeforeafter}

\begin{tcolorbox}[fit]
This is a tcolorbox.
\end{tcolorbox}
\begin{tcolorbox}[fit,fit width plus=1cm]
This is a tcolorbox.
\end{tcolorbox}
\begin{tcolorbox}[fit]
\lipsum[2]
\end{tcolorbox}
\begin{tcolorbox}[fit,fit width plus=1cm]
\lipsum[2]
\end{tcolorbox}
```



Typically but not necessarily, the optional title of a `tcolorbox` is not part of the fit operation. If a `/tcb/fit width plus` is applied, the title is also adapted to the new width. If counters are increased inside the title text, they may be increased more than one time. To avoid this, you are encouraged to use `/tcb/phantom` → P. 94 or `/tcb/step and label` → P. 94 to set counters or use automatic numbering, see Subsection 5.1 from page 100.

`/tcb/fit width from=`⟨*min*⟩ `to` ⟨*max*⟩ (style, no default)

Sets the box width to ⟨*min*⟩ and allows the width to grow up to ⟨*max*⟩.

```
% \usepackage{lipsum}
\tcbset{colback=red!5!white,colframe=red!75!black,left=1mm,top=1mm,bottom=1mm,
  right=1mm,boxsep=0mm,height=4cm}

\begin{tcolorbox}[fit,width=\linewidth/2]
\lipsum[2]
\end{tcolorbox}\par
\begin{tcolorbox}[fit width from=\linewidth/2 to \linewidth]
\lipsum[2]
\end{tcolorbox}\par
```

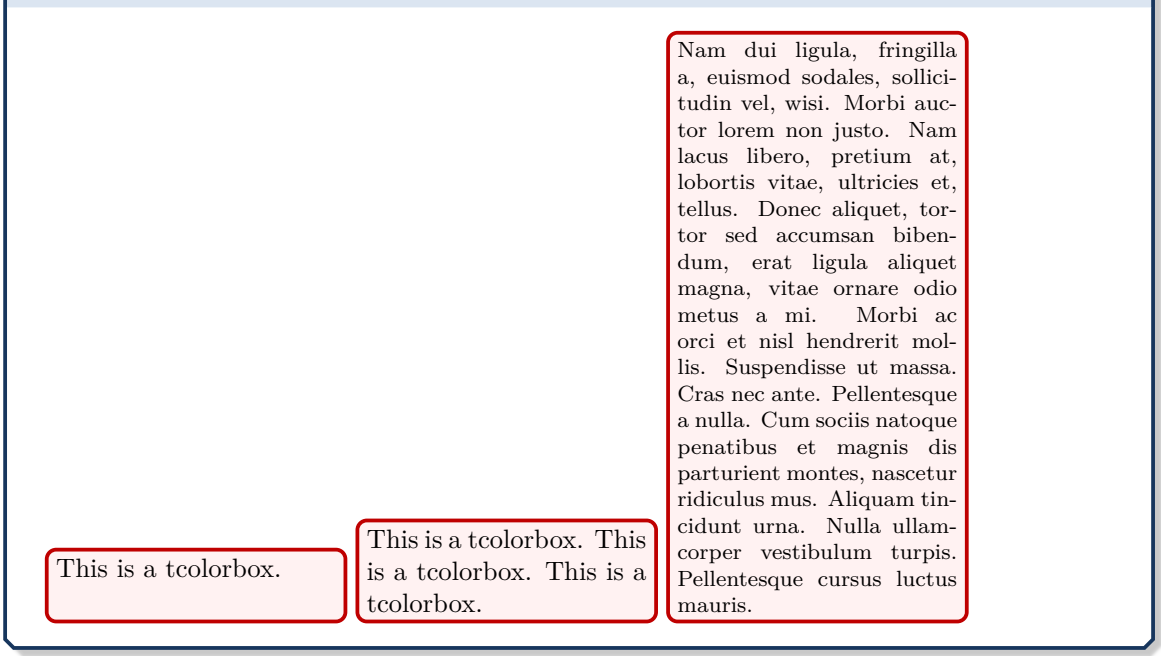
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/fit height from=`⟨*min*⟩ `to` ⟨*max*⟩ (style, no default)

Sets the box height to ⟨*min*⟩ and allows the height to grow up to ⟨*max*⟩.

```
% \usepackage{lipsum}
\newtcolorbox{mybox}{colback=red!5!white,colframe=red!75!black,left=1mm,top=1mm,
  bottom=1mm,right=1mm,boxsep=0mm,width=4cm,nobeforeafter,
  fit height from=1cm to 8cm}

\begin{mybox}
This is a tcolorbox.
\end{mybox}
\begin{mybox}
This is a tcolorbox. This is a tcolorbox. This is a tcolorbox.
\end{mybox}
\begin{mybox}
\lipsum[2]
\end{mybox}
```



`/tcb/fit algorithm=`⟨*name*⟩ (no default, initially `fontsize`)

Sets the algorithm for the fitting process *after* optionally width and height are adapted. Feasible values for ⟨*name*⟩ are:

- `fontsize` (initial): The algorithm is a bisection method that adapts the font size until certain stop conditions are fulfilled. This is the most time-consuming method but it is robust and gives pleasant results.

  ! The used font has to be freely scalable for this method! Other content than text is not scaled down. The aspect ratio is fully garanteed.

N 2014-10-29

- `fontsize*`: First, the `fontsize` algorithm is applied. If the font was scaled down and the resulting height is too small, the box is squeezed to fit the area.

  ! The used font has to be freely scalable for this method! Other content than text may be slightly rescaled. The aspect ratio cannot be fully garanteed.

- `areasize`: The algorithm calculates the area size for the text without scaling the font. The text box is shaped for the needed aspect ratio in one or two steps. Finally, it is scaled down with a standard `\resizebox` macro.

  ! The used font has not to be scalable. Every box content is scaled down. The aspect ratio cannot be fully garanteed.

N 2014-10-29

- `areasize*`: The `areasize` algorithm is applied, but if the content was scaled down and the resulting height is too small, the box is squeezed to fit the area.

  ! The used font has not to be scalable. Every box content is scaled down. The aspect ratio cannot be fully garanteed.

- `hybrid`: First, this algorithm estimates the needed font size in one or two steps. Then an `areasize` fitting as above is a applied.

  ! The used font has to be freely scalable for this method! Other content than text may be slightly rescaled. The aspect ratio cannot be fully garanteed.

N 2014-10-29

- `hybrid*`: First, this algorithm estimates the needed font size in one or two steps. Then an `areasize*` fitting as above is a applied.

  ! The used font has to be freely scalable for this method! Other content than text may be slightly rescaled. The aspect ratio cannot be fully garanteed.

- `squeeze`: The text box is brutally scaled down to fit.

  ! The aspect ratio is very likely to be horrible. You should not use this method for final documents.

```
% \usepackage{lipsum}
\newtcboxfit{mybox}[1]{colback=red!5!white,colframe=red!75!black,left=1mm,top=1mm,
  bottom=1mm,right=1mm,boxsep=0mm,width=3.5cm,height=7cm,nobeforeafter,
  before upper=\textcolor{blue}{\rule{5mm}{5mm}}\ ,
  enhanced,watermark text={\tcbfitsteps},
  fonttitle=\bfseries,adjusted title=#1,fit algorithm=#1}

\mybox{fontsize}{\lipsum[2]}\hfill
\mybox{hybrid}{\lipsum[2]}\hfill
\mybox{areasize}{\lipsum[2]}\hfill
\mybox{squeeze}{\lipsum[2]}

Quality \dotfill versus \dotfill Speed
```

fontsize

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

hybrid

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

areasize

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

squeeze

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Quality . . . . . versus . . . . . Speed

```
% \usepackage{lipsum}
\newtcboxfit{mybox}[2]{colback=red!5!white,colframe=red!75!black,left=1mm,top=1mm,
  size=tight,width=7.2cm,height=5cm,nobeforeafter,
  before upper=\textcolor{blue}{\rule{5mm}{5mm}}\ ,
  enhanced,fonttitle=\bfseries,adjusted title=#2,fit algorithm=#1}

\mybox{hybrid}{hybrid (possible gap at end)}{\lipsum[1]}\hfill
\mybox{hybrid*}{hybrid* (no gap but possibly squeezed)}{\lipsum[1]}
```

hybrid (possible gap at end)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

hybrid* (no gap but possibly squeezed)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

! The following options set control parameters for the fit algorithm. Mainly, they apply to the `fontsize` variant, see `/tcb/fit algorithm` → P. 388. The options should be seen as experimental and are likely to change in future versions, if necessary.

`/tcb/fit maxstep`=⟨*number*⟩ (no default, initially `20`)
Sets the maximal step size for the font size adjustment algorithm. In normal situations, the algorithm stops before reaching the intial value of 20 steps. If the box content does not shrink, this value prevents an endless loop.

`/tcb/fit maxfontdiff`=⟨*dimension*⟩ (no default, initially `0.1pt`)
The algorithm stops, if the font size is determined within a deviation of ⟨*dimension*⟩.

`/tcb/fit maxfontdiffgap`=⟨*dimension*⟩ (no default, initially `1pt`)
The algorithm stops, if the number of lines is determined and the font size is determined within a deviation of ⟨*dimension*⟩.

`/tcb/fit maxwidthdiff`=⟨*dimension*⟩ (no default, initially `1pt`)
The algorithm stops, if the (optionally) flexible box width is determined within a deviation of ⟨*dimension*⟩.

`/tcb/fit maxwidthdiffgap`=⟨*dimension*⟩ (no default, initially `10pt`)
The algorithm stops, if the number of lines is determined and the (optionally) flexible box width is determined within a deviation of ⟨*dimension*⟩.

`/tcb/fit warning`=⟨*value*⟩ (no default, initially `off`)
Typically, the fit control algorithm constructs several auxiliary boxes to determine the optimal one. If not switched off, the construction of the auxiliary boxes may produce many `hbox` warnings. This option key changes the `\hbadness` value.

- `off`: Most of `'Underfull \hbox'` and `'Overfull \hbox'` warnings are switched off (including the ones for the finally used box).
- `on`: All warnings for all auxiliary boxes are displayed.
- `final`: Only warnings for the finally used box are displayed. Note that an additional box has to be contructed for theses messages.

20 Library LIB hooks

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{hooks}
```

For the skin related options, the library LIB skins has to be loaded separately.

20.1 Concept of Hooks

A hook is a placeholder in some LaTeX code where additional code can be added. For example, the LaTeX macro `\AtBeginDocument` adds code to a hook which is placed at the beginning of every document.

Several option keys of `tcolorbox` allow providing some code which is added to specific places of a colored box. For example, `/tcb/before upper` → P. 63 places code before the content of the upper part. A following usage of this key overwrites any prior settings.

The library LIB hooks extends `/tcb/before upper` → P. 63 and several more existing keys to 'hookable' versions, e. g. `/tcb/before upper app` → P. 392 and `/tcb/before upper pre` → P. 392. The 'hookable' keys don't overwrite prior settings but either *app*end or *pre*pend the newly given code to the existing code.

The general naming convention (with some small exceptions) is:

- ⟨*option key*⟩ `app`: works like ⟨*option key*⟩ but *app*ends its code to the existing code.
- ⟨*option key*⟩ `pre`: works like ⟨*option key*⟩ but *pre*pends its code to the existing code.

If the original ⟨*option key*⟩ is used (again), all code will be overwritten. Therefore, the order of the option key usage is crucial.

```
% \usepackage{array,tabularx}
\newcolumntype{Y}{>{\raggedleft\arraybackslash}X}% see tabularx
\tcbset{enhanced,fonttitle=\bfseries\large,fontupper=\normalsize\sffamily,
  colback=yellow!10!white,colframe=red!50!black,colbacktitle=Salmon!30!white,
  coltitle=black,center title,
  tabularx={X||Y|Y|Y|Y||Y},% this sets 'before upper' and 'after upper'
  before upper app={Group & One & Two & Three & Four & Sum\\\hline\hline}  }

\begin{tcolorbox}[title=My table]
Red   & 1000.00 & 2000.00 &  3000.00 &  4000.00 & 10000.00\\\hline
Green & 2000.00 & 3000.00 &  4000.00 &  5000.00 & 14000.00\\\hline
Blue  & 3000.00 & 4000.00 &  5000.00 &  6000.00 & 18000.00\\\hline\hline
Sum   & 6000.00 & 9000.00 & 12000.00 & 15000.00 & 42000.00
\end{tcolorbox}
```

My table

Group	One	Two	Three	Four	Sum
Red	1000.00	2000.00	3000.00	4000.00	10000.00
Green	2000.00	3000.00	4000.00	5000.00	14000.00
Blue	3000.00	4000.00	5000.00	6000.00	18000.00
Sum	6000.00	9000.00	12000.00	15000.00	42000.00

20.2 Box Content Additions

The following option keys extend the options given in Subsection 4.11 from page 62.

/tcb/before title app=⟨*code*⟩ (no default)
Appends the given ⟨*code*⟩ to /tcb/before title→ P. 62 *after* the color and font settings and *before* the content of the title.

/tcb/before title pre=⟨*code*⟩ (no default)
Prepends the given ⟨*code*⟩ to /tcb/before title→ P. 62 *after* the color and font settings and *before* the content of the title.

/tcb/after title app=⟨*code*⟩ (no default)
Appends the given ⟨*code*⟩ to /tcb/after title→ P. 62 *after* the content of the title.

/tcb/after title pre=⟨*code*⟩ (no default)
Prepends the given ⟨*code*⟩ to /tcb/after title→ P. 62 *after* the content of the title.

/tcb/before upper app=⟨*code*⟩ (no default)
Appends the given ⟨*code*⟩ to /tcb/before upper→ P. 63 *after* the color and font settings and *before* the content of the upper part.

/tcb/before upper pre=⟨*code*⟩ (no default)
Prepends the given ⟨*code*⟩ to /tcb/before upper→ P. 63 *after* the color and font settings and *before* the content of the upper part.

/tcb/after upper app=⟨*code*⟩ (no default)
Appends the given ⟨*code*⟩ to /tcb/after upper→ P. 63 *after* the content of the upper part.

/tcb/after upper pre=⟨*code*⟩ (no default)
Prepends the given ⟨*code*⟩ to /tcb/after upper→ P. 63 *after* the content of the upper part.

```
% \tcbuselibrary{theorems}
\tcbset{ams align,%  this sets 'before upper' and 'after upper'
  colback=yellow!10!white,colframe=red!50!black,
  before upper app={\frac{2}{\sqrt{2}}&=\sqrt{2}.\\},
  after upper pre={\\\sin\left(\frac{\pi}{2}\right)&=1.},
}

\begin{tcolorbox}
  \sum\limits_{n=1}^{\infty} \frac{1}{n} &= \infty.\\
  \int x^2 ~\text{d}x &= \frac13 x^3 + c.
\end{tcolorbox}
```

$$\frac{2}{\sqrt{2}} = \sqrt{2}. \tag{22}$$

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty. \tag{23}$$

$$\int x^2 \, \mathrm{d}x = \frac{1}{3}x^3 + c. \tag{24}$$

$$\sin\left(\frac{\pi}{2}\right) = 1. \tag{25}$$

`/tcb/before lower app=`⟨*code*⟩ (no default)

Appends the given ⟨*code*⟩ to `/tcb/before lower` → P. 64 *after* the color and font settings and *before* the content of the lower part.

`/tcb/before lower pre=`⟨*code*⟩ (no default)

Prepends the given ⟨*code*⟩ to `/tcb/before lower` → P. 64 *after* the color and font settings and *before* the content of the lower part.

`/tcb/after lower app=`⟨*code*⟩ (no default)

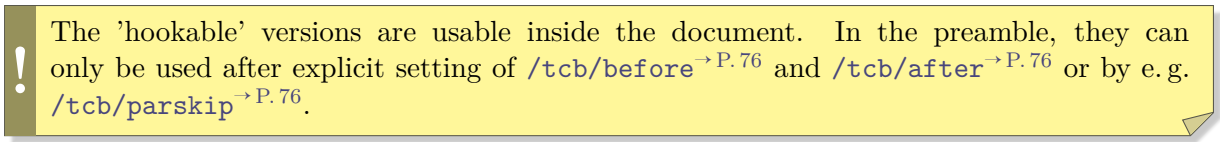
Appends the given ⟨*code*⟩ to `/tcb/after lower` → P. 64 *after* the content of the lower part.

`/tcb/after lower pre=`⟨*code*⟩ (no default)

Prepends the given ⟨*code*⟩ to `/tcb/after lower` → P. 64 *after* the content of the lower part.

20.3 Embedding into the Surroundings

The following option keys extend the options given in Subsection 4.14 from page 76.



The 'hookable' versions are usable inside the document. In the preamble, they can only be used after explicit setting of `/tcb/before` → P. 76 and `/tcb/after` → P. 76 or by e. g. `/tcb/parskip` → P. 76.

`/tcb/before app=`⟨*code*⟩ (no default)

Appends the given ⟨*code*⟩ to `/tcb/before` → P. 76 before the colored box.

`/tcb/before pre=`⟨*code*⟩ (no default)

Prepends the given ⟨*code*⟩ to `/tcb/before` → P. 76 before the colored box.

`/tcb/after app=`⟨*code*⟩ (no default)

Appends the given ⟨*code*⟩ to `/tcb/after` → P. 76 after the colored box.

`/tcb/after pre=`⟨*code*⟩ (no default)

Prepends the given ⟨*code*⟩ to `/tcb/after` → P. 76 after the colored box.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title,before app={The box follows:\\[4pt]},
  after app={This is the end.}]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

The box follows:

My title

This is a **tcolorbox**.

This is the end.

20.4 Overlays

The following option keys extend the options given in Subsection 4.12 from page 69.

`/tcb/overlay app=`⟨*graphical code*⟩ (no default)

Appends the given ⟨*graphical code*⟩ to `/tcb/overlay`→ P. 69.

```
% \usetikzlibrary{patterns} % preamble
% \tcbuselibrary{skins} % preamble
\tcbset{frogbox/.style={enhanced,colback=green!10,colframe=green!65!black,
  enlarge top by=5.5mm,
  overlay={\foreach \x in {2cm,3.5cm} {
    \begin{scope}[shift={([xshift=\x]frame.north west)}]
      \path[draw=green!65!black,fill=green!10,line width=1mm] (0,0) arc (0:180:5mm);
      \path[fill=black] (-0.2,0) arc (0:180:1mm);
    \end{scope}}}]}}
\tcbset{ribbon/.style={overlay app={%
  \path[fill=blue!75!white,draw=blue,double=white!85!blue,
    preaction={opacity=0.6,fill=blue!75!white},
    line width=0.1mm,double distance=0.2mm,
    pattern=fivepointed stars,pattern color=white!75!blue]
    ([xshift=-0.2mm,yshift=-1.02cm]frame.north east)
    -- ++(-1,1) -- ++(-0.5,0) -- ++(1.5,-1.5) -- cycle;}}}

\begin{tcolorbox}[frogbox,title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}

\begin{tcolorbox}[frogbox,ribbon,title=My title]
This is a \textbf{tcolorbox}.\par
Here, we apply a second overlay.
\end{tcolorbox}
```

`/tcb/overlay pre=`⟨*graphical code*⟩ (no default)

Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay`→ P. 69.

`/tcb/overlay unbroken app=`⟨*graphical code*⟩ (no default)

Appends the given ⟨*graphical code*⟩ to `/tcb/overlay unbroken`→ P. 70.

`/tcb/overlay unbroken pre=`⟨*graphical code*⟩ (no default)

Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay unbroken`→ P. 70.

`/tcb/overlay first app=`⟨*graphical code*⟩ (no default)

Appends the given ⟨*graphical code*⟩ to `/tcb/overlay first`→ P. 70.

`/tcb/overlay first pre=`⟨*graphical code*⟩ (no default)

Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay first`→ P. 70.

`/tcb/overlay middle app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/overlay middle` → P. 70.

`/tcb/overlay middle pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay middle` → P. 70.

`/tcb/overlay last app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/overlay last` → P. 70.

`/tcb/overlay last pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay last` → P. 70.

`/tcb/overlay broken app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/overlay broken` → P. 70.

`/tcb/overlay broken pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay broken` → P. 70.

`/tcb/overlay unbroken and first app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/overlay unbroken and first` → P. 70.

`/tcb/overlay unbroken and first pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay unbroken and first` → P. 70.

`/tcb/overlay middle and last app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/overlay middle and last` → P. 70.

`/tcb/overlay middle and last pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay middle and last` → P. 70.

`/tcb/overlay unbroken and last app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/overlay unbroken and last` → P. 70.

`/tcb/overlay unbroken and last pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay unbroken and last` → P. 70.

N 2014-09-19 `/tcb/overlay first and middle app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/overlay first and middle` → P. 70.

N 2014-09-19 `/tcb/overlay first and middle pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay first and middle` → P. 70.

20.5 Watermarks

The following option keys extend the options given in Subsection 10.3 from page 156.

> ! Watermarks are special overlays. The LIB hooks library allows the combination of several watermarks and overlays.

`/tcb/watermark text app=`⟨*text*⟩ (no default)

Appends a `/tcb/watermark text` → P. 156 to the colored box.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,watermark graphics=Basilica_5.png,
  watermark opacity=0.25,
  watermark text app=Basilica,watermark color=Navy
  ]
\lipsum[1-2]
\tcblower
This example uses a public domain picture from\\
\url{http://commons.wikimedia.org/wiki/File:Basilica_5.png}
\end{tcolorbox}
```

My title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

This example uses a public domain picture from
http://commons.wikimedia.org/wiki/File:Basilica_5.png

`/tcb/watermark text pre=`⟨*text*⟩ (no default)

Prepends a `/tcb/watermark text` → P. 156 to the colored box.

`/tcb/watermark text app on=`⟨*part*⟩ `is` ⟨*text*⟩ (no default)

Appends a `/tcb/watermark text on` → P. 156 the named ⟨*part*⟩ of a break sequence.

`/tcb/watermark text pre on=`⟨*part*⟩ `is` ⟨*text*⟩ (no default)

Prepends a `/tcb/watermark text on` → P. 156 the named ⟨*part*⟩ of a break sequence.

`/tcb/watermark graphics app=`⟨*file name*⟩ (no default)
Appends a `/tcb/watermark graphics` → P. 157 referenced by ⟨*file name*⟩ to the colored box.

`/tcb/watermark graphics pre=`⟨*file name*⟩ (no default)
Prepends a `/tcb/watermark graphics` → P. 157 referenced by ⟨*file name*⟩ to the colored box.

`/tcb/watermark graphics app on=`⟨*part*⟩ `is` ⟨*file name*⟩ (no default)
Appends a `/tcb/watermark graphics on` → P. 157 the named ⟨*part*⟩ of a break sequence. The picture is referenced by ⟨*file name*⟩.

`/tcb/watermark graphics pre on=`⟨*part*⟩ `is` ⟨*file name*⟩ (no default)
Prepends a `/tcb/watermark graphics on` → P. 157 the named ⟨*part*⟩ of a break sequence. The picture is referenced by ⟨*file name*⟩.

`/tcb/watermark tikz app=`⟨*graphical code*⟩ (no default)
Appends a `/tcb/watermark tikz` → P. 158 with the given `tikz` ⟨*graphical code*⟩ to the colored box.

`/tcb/watermark tikz pre=`⟨*graphical code*⟩ (no default)
Prepends a `/tcb/watermark tikz` → P. 158 with the given `tikz` ⟨*graphical code*⟩ to the colored box.

```
% \usepackage{tikz}
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  watermark color=Navy,watermark opacity=0.25,
  smiley/.style={watermark tikz pre={%
    \path[fill=yellow,draw=yellow!75!red] (0,0) circle (1cm);
    \fill[red] (45:5mm) circle (1mm);
    \fill[red] (135:5mm) circle (1mm);
    \draw[line width=1mm,red] (215:5mm) arc (215:325:5mm);}}

\begin{tcolorbox}[enhanced,title=My title, watermark text=Watermark,
  smiley]
\lipsum[1-2]
\end{tcolorbox}
```

My title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Watermark

`/tcb/watermark tikz app on=`⟨*part*⟩ `is` ⟨*graphical code*⟩ (no default)
Appends a `/tcb/watermark tikz on` → P. 158 the named ⟨*part*⟩ of a break sequence.

`/tcb/watermark tikz pre on=`⟨*part*⟩ `is` ⟨*graphical code*⟩ (no default)
Prepends a `/tcb/watermark tikz on` → P. 158 the named ⟨*part*⟩ of a break sequence.

20.6 Underlays

The following option keys extend the options given in Section 10.8 on page 186. There are no `app` type keys since underlays are stackable by default.

`/tcb/underlay pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay` → P. 186.

`/tcb/underlay unbroken pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay unbroken` → P. 187.

`/tcb/underlay first pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay first` → P. 187.

`/tcb/underlay middle pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay middle` → P. 187.

`/tcb/underlay last pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay last` → P. 187.

`/tcb/underlay boxed title pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay boxed title` → P. 187.

`/tcb/underlay broken pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay broken` → P. 187.

`/tcb/underlay unbroken and first pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay unbroken and first` → P. 187.

`/tcb/underlay middle and last pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay middle and last` → P. 187.

`/tcb/underlay unbroken and last pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay unbroken and last` → P. 187.

N 2014-09-19 `/tcb/underlay first and middle pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay first and middle` → P. 187.

20.7 Finishes

The following option keys extend the options given in Section 10.9 on page 188. There are no `app` type keys since finishes are stackable by default.

`/tcb/finish pre=`⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish` → P. 188.

`/tcb/finish unbroken pre=`⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish unbroken` → P. 189.

`/tcb/finish first pre=`⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish first` → P. 189.

`/tcb/finish middle pre=`⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish middle` → P. 189.

`/tcb/finish last pre=`⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish last` → P. 189.

`/tcb/finish broken pre=`⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish broken` → P. 189.

`/tcb/finish unbroken and first pre=`⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish unbroken and first` → P. 189.

`/tcb/finish middle and last pre=`⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish middle and last` → P. 189.

`/tcb/finish unbroken and last pre=`⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish unbroken and last` → P. 189.

N 2014-09-19 `/tcb/finish first and middle pre=`⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish first and middle` → P. 189.

20.8 Skin Code

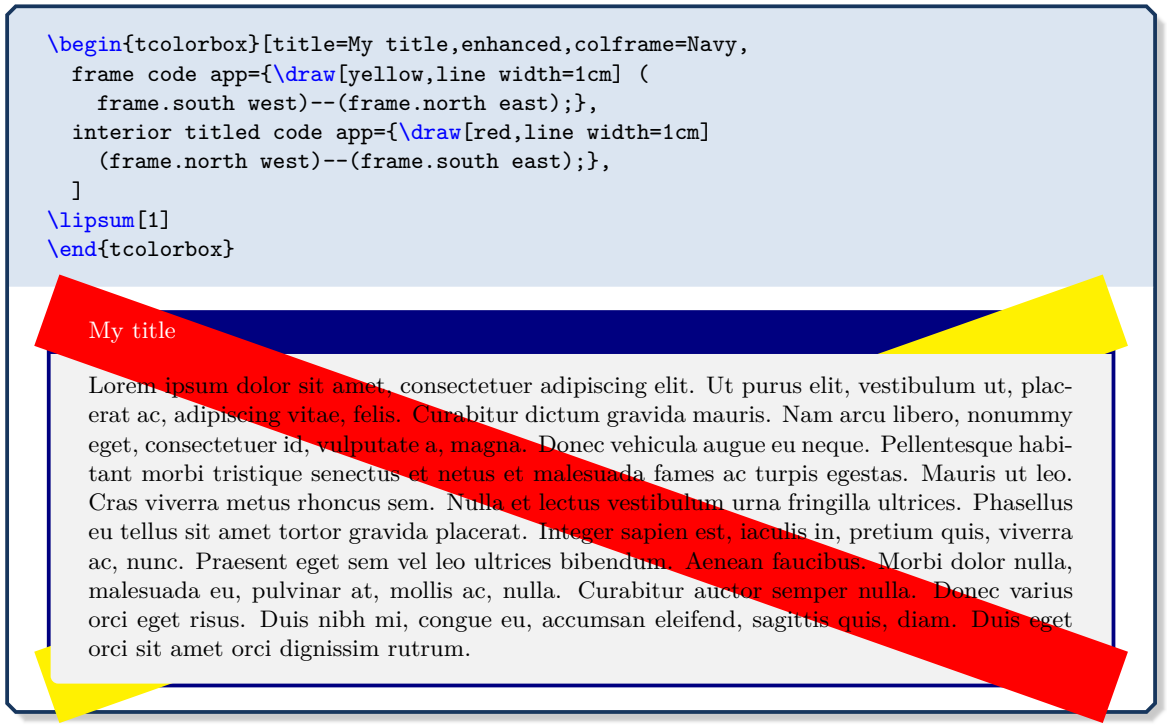
The following option keys extend the options given in Subsection 9.2 from page 129.

`/tcb/frame code app=`⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/frame code` → P. 129.

`/tcb/frame code pre=`⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/frame code` → P. 129.

`/tcb/interior titled code app=`⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/interior titled code` → P. 129.

```
\begin{tcolorbox}[title=My title,enhanced,colframe=Navy,
  frame code app={\draw[yellow,line width=1cm] (
    frame.south west)--(frame.north east);},
  interior titled code app={\draw[red,line width=1cm]
    (frame.north west)--(frame.south east);},
  ]
\lipsum[1]
\end{tcolorbox}
```



/tcb/interior titled code pre=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to /tcb/interior titled code → P. 129.

/tcb/interior code app=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to /tcb/interior code → P. 130.

/tcb/interior code pre=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to /tcb/interior code → P. 130.

/tcb/segmentation code app=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to /tcb/segmentation code → P. 130.

/tcb/segmentation code pre=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to /tcb/segmentation code → P. 130.

/tcb/title code app=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to /tcb/title code → P. 131.

/tcb/title code pre=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to /tcb/title code → P. 131.

20.9 Extras

The following option keys extend the options given in Section 17.5 on page 357. There are no app type keys since extras are stackable by default.

N 2015-07-16 **/tcb/extras pre={⟨*options*⟩}** (no default)
Prepends the given ⟨*options*⟩ to /tcb/extras→ P. 357.

N 2015-07-16 **/tcb/extras unbroken pre={⟨*options*⟩}** (no default)
Prepends the given ⟨*options*⟩ to /tcb/extras unbroken→ P. 357.

N 2015-07-16 **/tcb/extras first pre={⟨*options*⟩}** (no default)
Prepends the given ⟨*options*⟩ to /tcb/extras first→ P. 357.

N 2015-07-16 **/tcb/extras middle pre={⟨*options*⟩}** (no default)
Prepends the given ⟨*options*⟩ to /tcb/extras middle→ P. 357.

N 2015-07-16 **/tcb/extras last pre={⟨*options*⟩}** (no default)
Prepends the given ⟨*options*⟩ to /tcb/extras last→ P. 357.

N 2015-07-16 **/tcb/extras broken pre={⟨*options*⟩}** (no default)
Prepends the given ⟨*options*⟩ to /tcb/extras broken→ P. 357.

N 2015-07-16 **/tcb/extras unbroken and first pre={⟨*options*⟩}** (no default)
Prepends the given ⟨*options*⟩ to /tcb/extras unbroken and first→ P. 357.

N 2015-07-16 **/tcb/extras middle and last pre={⟨*options*⟩}** (no default)
Prepends the given ⟨*options*⟩ to /tcb/extras middle and last→ P. 357.

N 2015-07-16 **/tcb/extras unbroken and last pre={⟨*options*⟩}** (no default)
Prepends the given ⟨*options*⟩ to /tcb/extras unbroken and last→ P. 357.

N 2015-07-16 **/tcb/extras first and middle pre={⟨*options*⟩}** (no default)
Prepends the given ⟨*options*⟩ to /tcb/extras first and middle→ P. 357.

21 Library LIB xparse

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{xparse}
```

This also loads the package `xparse` [12].

The purpose of this library is to give comfortable access to the powerful document command production with `xparse` for `tcolorbox`. See the `xparse` package documentation [12] for details about the argument ⟨*specification*⟩ used in this section.

21.1 Option Keys

`/tcb/verbatim` (style, no value)

Sets options for a *verbatim* style `\tcbox` → P. 14. Since the indented boxes may contain only very few words, the dimensions are made smaller and `/tcb/nobeforeafter` → P. 76 and `/tcb/tcbox raise base` → P. 92 are set.

```
\DeclareTotalTCBox{\myverb}{ v }{verbatim,
  colframe=red!75!black,colupper=blue}{#1}

\myverb{\textbf} is a \myverb{\LaTeX} command.
```

`\textbf` is a `\LaTeX` command.

`/tcb/IfNoValueTF={`⟨*argument*⟩`}{`⟨*true options*⟩`}{`⟨*false options*⟩`}` (no default)

Wraps the `\IfNoValueTF` command of `xparse` for option setting. If the ⟨*argument*⟩ has no value, the ⟨*true options*⟩ are set. Otherwise, the ⟨*false options*⟩ are set.

```
\DeclareTColorBox{mybox}{ o }{colframe=red!75!black,
  IfNoValueTF={#1}{colback=red!5!white}{enhanced,interior style image=#1}}

\begin{mybox}
This is a tcolorbox.
\end{mybox}

\begin{mybox}[goldshade.png]
This is a tcolorbox.
\end{mybox}
```

This is a tcolorbox.

This is a tcolorbox.

`/tcb/IfValueTF={⟨argument⟩}{⟨true options⟩}{⟨false options⟩}` (no default)

Wraps the `\IfValueTF` command of `xparse` for option setting. If the ⟨*argument*⟩ has a value, the ⟨*true options*⟩ are set. Otherwise, the ⟨*false options*⟩ are set.

```
\DeclareTColorBox{mybox}{ o }{colframe=red!75!black,colback=red!5!white,
  IfValueTF={#1}{title={\flqq #1\frqq},fonttitle=\bfseries}{}}

\begin{mybox}
This is a tcolorbox.
\end{mybox}

\begin{mybox}[My title]
This is a tcolorbox.
\end{mybox}
```

This is a tcolorbox.

«My title»

This is a tcolorbox.

`/tcb/IfBooleanTF={⟨argument⟩}{⟨true options⟩}{⟨false options⟩}` (no default)

Wraps the `\IfBooleanTF` command of `xparse` for option setting. If the ⟨*argument*⟩ is `\BooleanTue`, the ⟨*true options*⟩ are set. If the ⟨*argument*⟩ is `\BooleanFalse`, the ⟨*false options*⟩ are set.

```
\DeclareTColorBox{mybox}{ s }{colframe=red!75!black,
  IfBooleanTF={#1}{colback=yellow!50!red}{colback=red!5!white}}

\begin{mybox}
This is a tcolorbox.
\end{mybox}

\begin{mybox}*
This is a tcolorbox.
\end{mybox}
```

This is a tcolorbox.

This is a tcolorbox.

21.2 Producing tcolorbox Environments and Commands

\DeclareTColorBox[⟨*init options*⟩]{⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}

Creates a new environment ⟨*name*⟩ based on tcolorbox → P. 12.

Basically, \DeclareTColorBox operates like \DeclareDocumentEnvironment. This means, the new environment ⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying tcolorbox → P. 12.

Note that /tcb/savedelimiter → P. 26 is set to the given ⟨*name*⟩ automatically.

The ⟨*init options*⟩ allow setting up automatic numbering, see Section 5 from page 100.

The new environment is always created, irrespective of an already existing environment with the same name.

```
% counter from previous example
\DeclareTColorBox[use counter from=pabox]{mybox}{ O{red} m d"" O{} }
  {enhanced,colframe=#1!75!black,colback=#1!5!white,
   fonttitle=\bfseries,title={\thetcbcounter~#2},
   IfValueTF={#3}{watermark text={#3}}{},#4}

\begin{mybox}{My title}
This is a tcolorbox.
\end{mybox}

\begin{mybox}[blue]{My title}
This is a tcolorbox.
\end{mybox}

\begin{mybox}[green]{My title}"My Watermark"
This is a tcolorbox.
\end{mybox}

\begin{mybox}[yellow]{My title}[colbacktitle=yellow!50!white,coltitle=black]
This is a tcolorbox.
\end{mybox}

\begin{mybox}[purple]{My title}"All together"[coltitle=yellow]
This is a tcolorbox.
\end{mybox}
```

21.1 My title

This is a tcolorbox.

21.2 My title

This is a tcolorbox.

21.3 My title

This is a tcolorbox. My Watermark

21.4 My title

This is a tcolorbox.

21.5 My title

This is a tcolorbox. All together

`\NewTColorBox[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTColorBox` → P. 404, but based on `\NewDocumentEnvironment` instead of `\DeclareDocumentEnvironment`. An error is issued if ⟨*name*⟩ has already been defined.

`\RenewTColorBox[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTColorBox` → P. 404, but based on `\RenewDocumentEnvironment` instead of `\DeclareDocumentEnvironment`. An existing environment is redefined.

`\ProvideTColorBox[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTColorBox` → P. 404, but based on `\ProvideDocumentEnvironment` instead of `\DeclareDocumentEnvironment`. The environment ⟨*name*⟩ is only created if it is not already defined.

`\DeclareTotalTColorBox[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}{`⟨*content*⟩`}`

Creates a new command `\`⟨*name*⟩ based on `tcolorbox` → P. 12. In contrast to `\DeclareTColorBox` → P. 404, also the ⟨*content*⟩ of the `tcolorbox` is specified.

Basically, `\DeclareTotalTColorBox` operates like `\DeclareDocumentCommand`. This means, the new command `\`⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying `tcolorbox` → P. 12 which is filled with the specified ⟨*content*⟩.

Note that `/tcb/savedelimiter` → P. 26 is set to the given ⟨*name*⟩ automatically.

The ⟨*init options*⟩ allow setting up automatic numbering, see Section 5 from page 100.

The new command is always created, irrespective of an already existing command with the same name.

```
\DeclareTotalTColorBox{\diabox}{ O{} v  m }
  { bicolor,nobeforeafter,equal height group=diabox,width=5.7cm,
    fonttitle=\bfseries\ttfamily,adjusted title={#2},center title,
    colframe=blue!20!black,leftupper=0mm,rightupper=0mm,colback=black!75!white,#1}
  { \tikz\path[fill zoom image={#2}] (0,0) rectangle (\linewidth,4cm);%
    \tcblower#3}

\diabox{blueshade.png}{Created with |GIMP|.\\\url{http://www.gimp.org}}
\diabox{goldshade.png}{Created with |GIMP|.\\\url{http://www.gimp.org}}
```

`\NewTotalTColorBox[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}{`⟨*content*⟩`}`

Operates like `\DeclareTotalTColorBox`, but based on `\NewDocumentCommand` instead of `\DeclareDocumentCommand`. An error is issued if `\`⟨*name*⟩ has already been defined.

`\RenewTotalTColorBox[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}{`⟨*content*⟩`}`

Operates like `\DeclareTotalTColorBox`, but based on `\RenewDocumentCommand` instead of `\DeclareDocumentCommand`. An existing command is redefined.

`\ProvideTotalTColorBox[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}{`⟨*content*⟩`}`

Operates like `\DeclareTotalTColorBox`, but based on `\ProvideDocumentCommand` instead of `\DeclareDocumentCommand`. The command `\`⟨*name*⟩ is only created if it is not already defined.

21.3 Producing tcbox Commands

\DeclareTCBox[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}

Creates a new command \⟨*name*⟩ based on \tcbox → P. 14. Basically, \DeclareTCBox operates like \DeclareDocumentCommand. This means, the new command \⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying \tcbox → P. 14.

Note that /tcb/savedelimiter → P. 26 is set to the given ⟨*name*⟩ automatically.

The ⟨*init options*⟩ allow setting up automatic numbering, see Section 5 from page 100.

The new command is always created, irrespective of an already existing command with the same name.

```
% counter from previous example
\DeclareTCBox[use counter from=pabox]{\mybox}{ s m s }
{ nobeforeafter,colback=red!5!white,colframe=red!75!black,
  title={#2 (Box \thetcbcounter)},fonttitle=\bfseries,
  IfBooleanTF={#1}{enhanced,drop shadow}{},
  IfBooleanTF={#3}{colbacktitle=red!50!white}{}  }

\mybox{Bird}{This is my first box.}
  \hfill
\mybox*{Tree}{This is my second box.}
  \par\bigskip
\mybox{Bike}*{This is my third box.}
  \hfill
\mybox*{City}*{This is my fourth box.}
```

Bird (Box 21.6)

This is my first box.

Tree (Box 21.7)

This is my second box.

Bike (Box 21.8)

This is my third box.

City (Box 21.9)

This is my fourth box.

\NewTCBox[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}

Operates like \DeclareTCBox, but based on \NewDocumentCommand instead of \DeclareDocumentCommand. An error is issued if \⟨*name*⟩ has already been defined.

\RenewTCBox[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}

Operates like \DeclareTCBox, but based on \RenewDocumentCommand instead of \DeclareDocumentCommand. An existing command is redefined.

\ProvideTCBox[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}

Operates like \DeclareTCBox, but based on \ProvideDocumentCommand instead of \DeclareDocumentCommand. The command \⟨*name*⟩ is only created if it is not already defined.

`\DeclareTotalTCBox`[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}{⟨*content*⟩}

Creates a new command \⟨*name*⟩ based on `\tcbox` → P. 14. In contrast to `\DeclareTCBox` → P. 407, also the ⟨*content*⟩ of the `tcbox` is specified.

Basically, `\DeclareTotalTCBox` operates like `\DeclareDocumentCommand`. This means, the new command \⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying `\tcbox` → P. 14 which is filled with the specified ⟨*content*⟩.

Note that `/tcb/savedelimiter` → P. 26 is set to the given ⟨*name*⟩ automatically.

The ⟨*init options*⟩ allow setting up automatic numbering, see Section 5 from page 100.

The new command is always created, irrespective of an already existing command with the same name.

```
\DeclareTotalTCBox{\myverb}{ O{red} v O{} }
{ fontupper=\ttfamily,nobeforeafter,tcbox raise base,arc=0pt,outer arc=0pt,
  top=0pt,bottom=0pt,left=0mm,right=0mm,
  leftrule=0pt,rightrule=0pt,toprule=0.3mm,bottomrule=0.3mm,boxsep=0.5mm,
  colback=#1!10!white,colframe=#1!50!black,#3}{#2}

To set a word \textbf{bold} in \myverb{\LaTeX}, use
\myverb[green]{\textbf{bold}}. Alternatively, write
\myverb[yellow]{{\bfseries bold}}.
In \myverb[blue]{\LaTeX}[enhanced,fuzzy halo], other font settings are
done in the same way, e.\,g. \myverb{\textit}, \myverb{\itshape}\\
or \myverb[brown]{\texttt}, \myverb[brown]{\ttfamily}.
```

To set a word **bold** in `\LaTeX`, use `\textbf{bold}`. Alternatively, write `{\bfseries bold}`. In `\LaTeX`, other font settings are done in the same way, e. g. `\textit`, `\itshape` or `\texttt`, `\ttfamily`.

The next example uses `\lstinline` from the `listings` package to typeset the verbatim content.

```
% \usepackage{listings} or \tcbuselibrary{listings}
\DeclareTotalTCBox{\commandbox}{ s v }
{verbatim,colupper=white,colback=black!75!white,colframe=black}
{\IfBooleanTF{#1}{\textcolor{red}{\ttfamily\bfseries > }}{}%
  \lstinline[language=command.com,keywordstyle=\color{blue!35!white}\bfseries]^#2^}

\commandbox*{cd "My Documents"} changes to directory \commandbox{My Documents}.

\commandbox*{dir /A} lists the directory content.

\commandbox*{copy example.txt d:\target} copies \commandbox{example.txt} to
  \commandbox{d:\target}.
```

`> cd "My␣Documents"` changes to directory `My Documents`.
`> dir /A` lists the directory content.
`> copy example.txt d:\target` copies `example.txt` to `d:\target`.

`\NewTotalTCBox[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}{`⟨*content*⟩`}`

Operates like `\DeclareTotalTCBox` → P. 408, but based on `\NewDocumentCommand` instead of `\DeclareDocumentCommand`. An error is issued if `\`⟨*name*⟩ has already been defined.

`\RenewTotalTCBox[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}{`⟨*content*⟩`}`

Operates like `\DeclareTotalTCBox` → P. 408, but based on `\RenewDocumentCommand` instead of `\DeclareDocumentCommand`. An existing command is redefined.

`\ProvideTotalTCBox[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}{`⟨*content*⟩`}`

Operates like `\DeclareTotalTCBox` → P. 408, but based on `\ProvideDocumentCommand` instead of `\DeclareDocumentCommand`. The command `\`⟨*name*⟩ is only created if it is not already defined.

`\tcboxverb[`⟨*options*⟩`]{`⟨*verbatim box content*⟩`}`

Creates a colored box based on `\tcbox` → P. 14 which is fitted to the width of the given ⟨*verbatim box content*⟩. The underlying `\tcbox` → P. 14 is styled with `/tcb/verbatim` → P. 402 plus the given ⟨*options*⟩. The difference to `\tcbox` → P. 14 is that the ⟨*verbatim box content*⟩ is interpreted *verbatim*. Therefore, `\tcboxverb` acts similar to `\verb`.

```
\tcboxverb{\LaTeX}, \tcboxverb[colback=blue!10!white,colupper=blue]{\LaTeX},
\tcboxverb[blank,fuzzy halo]{\LaTeX}, \tcboxverb[beamer]{\LaTeX},
\tcboxverb[enhanced,skin=enhancedmiddle jigsaw,colframe=red]{\LaTeX}.
```

`\LaTeX`, `\LaTeX`, `\LaTeX`, `\LaTeX`, `\LaTeX`.

21.4 Producing `tcblisting` Environments

The following commands need the LIB listings library to be included.

`\DeclareTCBListing[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Creates a new environment ⟨*name*⟩ based on `tcblisting` → P. 287.
Basically, `\DeclareTCBListing` operates like `\DeclareDocumentEnvironment`. This means, the new environment ⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying `tcblisting` → P. 287.
Note that `/tcb/savedelimiter` → P. 26 is set to the given ⟨*name*⟩ automatically.
The ⟨*init options*⟩ allow setting up automatic numbering, see Section 5 from page 100.
The new environment is always created, irrespective of an already existing environment with the same name.

```
\DeclareTCBListing{mybox}{ s O{} m }{%
  colback=red!5!white,
  colframe=red!75!black,
  fonttitle=\bfseries,
  IfBooleanTF={#1}
    {listing side text}
    {text side listing},
  title=#3,#2}

\begin{mybox}{Listing Box}
This is my
\LaTeX\ box.
\end{mybox}
\bigskip

\begin{mybox}*{Listing Box}
This is my
\LaTeX\ box.
\end{mybox}
\bigskip

\begin{mybox}[colback=yellow]
  {Listing Box}
This is my
\LaTeX\ box.
\end{mybox}
```

Listing Box

This is my LaTeX box.

This is my \LaTeX\ box.

Listing Box

This is my \LaTeX\ box.

This is my LaTeX box.

Listing Box

This is my LaTeX box.

This is my \LaTeX\ box.

`\NewTCBListing[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTCBListing`, but based on `\NewDocumentEnvironment` instead of `\DeclareDocumentEnvironment`. An error is issued if ⟨*name*⟩ has already been defined.

`\RenewTCBListing[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTCBListing`, but based on `\RenewDocumentEnvironment` instead of `\DeclareDocumentEnvironment`. An existing environment is redefined.

`\ProvideTCBListing[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTCBListing`, but based on `\ProvideDocumentEnvironment` instead of `\DeclareDocumentEnvironment`. The environment ⟨*name*⟩ is only created if it is not already defined.

Caveats of using an environment ending with an optional argument

```
\DeclareTCBListing{mybox}{ O{} }{listing only,#1}

\begin{mybox}[colframe=red]
\good
\end{mybox}

\begin{mybox}[colframe=red]\good\end{mybox}

\begin{mybox}
\good
\end{mybox}

\begin{mybox} \good\end{mybox}

\begin{mybox}\bad!\end{mybox}

\begin{mybox}
[\good]
\end{mybox}

\begin{mybox} [\good]\end{mybox}

\begin{mybox}[\bad!]\end{mybox}
```

21.5 Producing `tcbinputlisting` Commands

The following commands need the LIB listings library to be included.

`\DeclareTCBInputListing[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Creates a new command `\`⟨*name*⟩ based on `\tcbinputlisting` → P. 289. Basically, `\DeclareTCBInputListing` operates like `\DeclareDocumentCommand`. This means, the new command `\`⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying `\tcbinputlisting` → P. 289.

The ⟨*init options*⟩ allow setting up automatic numbering, see Section 5 from page 100.

The new command is always created, irrespective of an already existing command with the same name.

```
% counter from previous example
\DeclareTCBInputListing[use counter from=pabox]{\mylisting}{ O{} O{red} m }{%
  listing file={#3},title=Listing~\thetcbcounter,
  colback=#2!5!white,colframe=#2!50!black,colbacktitle=#2!75!black,
  fonttitle=\bfseries,listing only,#1}

\mylisting[before upper=\textit{This is the included file content:}]
  [blue]{\jobname.tcbtemp}
```

Listing 21.10

This is the included file content:

```
% counter from previous example
\DeclareTCBInputListing[use counter from=pabox]{\mylisting}{ O{}
    O{red} m }{%
  listing file={#3},title=Listing~\thetcbcounter,
  colback=#2!5!white,colframe=#2!50!black,colbacktitle=#2!75!black,
  fonttitle=\bfseries,listing only,#1}

\mylisting[before upper=\textit{This is the included file content:}]
  [blue]{\jobname.tcbtemp}
```

`\NewTCBInputListing[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTCBInputListing`, but based on `\NewDocumentCommand` instead of `\DeclareDocumentCommand`. An error is issued if `\`⟨*name*⟩ has already been defined.

`\RenewTCBInputListing[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTCBInputListing`, but based on `\RenewDocumentCommand` instead of `\DeclareDocumentCommand`. An existing command is redefined.

`\ProvideTCBInputListing[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTCBInputListing`, but based on `\ProvideDocumentCommand` instead of `\DeclareDocumentCommand`. The command `\`⟨*name*⟩ is only created if it is not already defined.

21.6 Producing `tboxfit` Commands

The following commands need the LIB fitting library to be included.

`\DeclareTCBoxFit`[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}

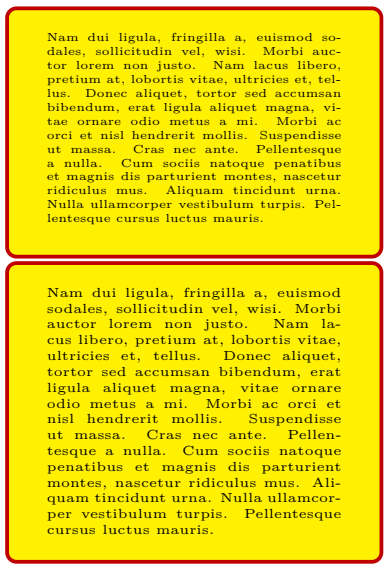
Creates a new command \⟨*name*⟩ based on `\tcboxfit` → P. 380. Basically, `\DeclareTCBoxFit` operates like `\DeclareDocumentCommand`. This means, the new command \⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying `\tcboxfit` → P. 380.

Note that `/tcb/savedelimiter` → P. 26 is set to the given ⟨*name*⟩ automatically.

The ⟨*init options*⟩ allow setting up automatic numbering, see Section 5 from page 100.

The new command is always created, irrespective of an already existing command with the same name.

```
% \usepackage{lipsum}

\DeclareTCBoxFit{\mybox}{ O{} m o }
 {colback=red!5!white,
  colframe=red!75!black,
  width=#2,height=#2/3*2,
  IfValueTF={#3}{height=#3}{},
  #1}

\mybox[colback=yellow]{5cm}%
  {\lipsum[2]}

\mybox[colback=yellow]{5cm}[4cm]{\lipsum[2]}
```



`\NewTCBoxFit`[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}

Operates like `\DeclareTCBoxFit`, but based on `\NewDocumentCommand` instead of `\DeclareDocumentCommand`. An error is issued if \⟨*name*⟩ has already been defined.

`\RenewTCBoxFit`[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}

Operates like `\DeclareTCBoxFit`, but based on `\RenewDocumentCommand` instead of `\DeclareDocumentCommand`. An existing command is redefined.

`\ProvideTCBoxFit`[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}

Operates like `\DeclareTCBoxFit`, but based on `\ProvideDocumentCommand` instead of `\DeclareDocumentCommand`. The command \⟨*name*⟩ is only created if it is not already defined.

`\DeclareTotalTCBoxFit`[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}{⟨*content*⟩}

Creates a new command \⟨*name*⟩ based on `\tcboxfit` →P. 380. In contrast to `\DeclareTCBoxFit` →P. 413, also the ⟨*content*⟩ of the `tcboxfit` is specified.

Basically, `\DeclareTotalTCBoxFit` operates like `\DeclareDocumentCommand`. This means, the new command \⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying `\tcboxfit` →P. 380 which is filled with the specified ⟨*content*⟩.

Note that `/tcb/savedelimiter` →P. 26 is set to the given ⟨*name*⟩ automatically.

The ⟨*init options*⟩ allow setting up automatic numbering, see Section 5 from page 100.

The new command is always created, irrespective of an already existing command with the same name.

```
% \usepackage{lipsum}

\DeclareTotalTCBoxFit{\multibox}{ O{} m O{10} m }
 {nobeforeafter,colback=red!5!white,colframe=red!75!black,width=#2,height=#2/3*2,
  valign=center,#1}
 { \foreach \n in {1,...,#3} { #4} }

\multibox{5cm}{I shall not repeat.}
\multibox[colframe=blue!75!white]{5cm}[20]{I shall not repeat.}\\
\multibox[colback=yellow,height=5cm]{14cm}[100]{I shall not repeat.}
```

I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat.

I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.

I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat.

22 Library LIB external

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{external}
```

The purpose of this library is to support externalization of document snippets like graphics or boxes which can be compiled stand-alone. These snippets are written to external files, compiled and the resulting pdf files are included to the main document as images. The whole procedure saves compilation time, if such a snippet is costly to compile but needs to compile just once or very seldom.

There are very good alternatives to this library. One should consider the `standalone` package or the Ti*k*Z externalization library instead. The LIB external library is something in between and can be seen as poor man variant of the Ti*k*Z externalization library.

The main differences between Ti*k*Z externalization and LIB external are:

- Ti*k*Z `external` compiles the whole original document in a sophisticated way while LIB external uses only the preamble or a part of the preamble of the original document.
- Ti*k*Z `external` can automatically externalize all `tikzpicture` environments while LIB external externalizes marked snippets only.
- Code snippets to be externalized by LIB external are not restricted to `tikzpicture` environments. But these snippets have to be stand-alone without dependencies to the rest of the document.

Why should somebody use LIB external instead of the more powerful Ti*k*Z `external`? One reason could be compilation speed, but the main reason for creating the library at all was that Ti*k*Z `external` tends to choke on complicated documents where the sophisticated mechanism stumbles. Since LIB external does not use the original document body for compilation, this cannot happen.

! Source snippets are compiled, if their `md5` checksum has changed. They are not compiled automatically, if option settings are changed or anything outside the snippet is changed. Use `/tcb/external/force remake` → P. 416 to force compilation in this case or simply delete the externalized snippets.

! To use the externalization options, the compiler has to be called with the `-shell-escape` permission to authorize potentially dangerous system calls. Be warned that this is a security risk.

22.1 Preparation of a Document for Externalization

The preamble of the main document has to contain the \tcbEXTERNALIZE command. Without this command, no externalization operation will be executed.

N 2015-03-11 **\tcbEXTERNALIZE**

It is mandatory for externalization that this command is used once in the preamble of the main document. Every setting *before* \tcbEXTERNALIZE will also be used for compiling an external snippet. Every setting *after* \tcbEXTERNALIZE will be ignored for compiling an external snippet. Place this command right before \begin{document}, if you are not absolutely sure about another place.
The main document has to look like the following:

```
\documentclass[a4paper]{book}%   for example
\usepackage{...}%                anything
% ...
% Tpyically, all or the very most settings for the document.

\tcbEXTERNALIZE% Typically, just before \begin{document}

% Additional settings which are ABSOLUTELY irrelevant for the
% stand-alone snippets.
%
\begin{document}
  % The document.
  % This also contains the marked snippets for externalization.
\end{document}
```

During compilation, a /tcb/external/runner file is dynamically created (several times). This is the actual main file for compiling an externalized snippet.

N 2015-03-11 **/tcb/external/runner=**⟨*file name*⟩ (no default, initially \jobname_run.tex)

Sets the ⟨*file name*⟩ for dynamically created runner file. This is the actual main file for a document snippet. Typically, the initial setting is not needed to be changed.

```
\tcbset{external/runner=myrunner.tex}
```

N 2015-03-11 **/tcb/external/prefix=**⟨*text*⟩ (no default, initially external/)

The ⟨*text*⟩ is prefixed to any /tcb/external/name → P. 418 for an externalization snippet. The initial setting implies saving all snippets into an external/ subdirectory. Depending on the operation system, the subdirectory may have to be created manually once.

```
% Use a 'real' prefix instead of writing into a subdirectory:
\tcbset{external/prefix=ext_}
```

N 2015-03-11 **/tcb/external/externalize=true|false** (default true, initially true)

If set to true, the marked snippets are compiled if necessary. If set to false, the marked snippets are not compiled but included as text. /tcb/external/externalize can only be used after \tcbEXTERNALIZE.

N 2015-03-11 **/tcb/external/force remake=true|false** (default true, initially false)

If set to true, the marked snippets are always compiled. If set to true, the marked snippets are compiled only if necessary. The necessity is given, if a compiled pdf file is missing or the md5 checksum of the source snippet has changed.

N 2015-03-11 **/tcb/external/!** (style)

Shortcut for setting /tcb/external/force remake to true.

N 2015-06-12 **/tcb/external/-** (style)

Shortcut for setting /tcb/external/externalize to false.

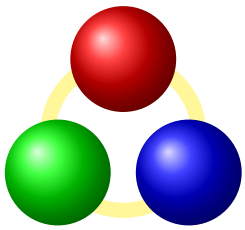
22.2 Marking Externalization Snippets

N 2015-03-11

```
\begin{tcbexternal}[⟨options⟩]{⟨name⟩}
  ⟨environment content⟩
\end{tcbexternal}
```

Marks the environment content as a snippet for externalization. Typically, the content is a `tikzpicture` or something similar. It is important to note that the snippet should not have any dependencies with the rest of the document, e.g. referencing counters or setting counters is not possible. The ⟨*name*⟩ is automatically prefixed with `/tcb/external/prefix`. In combination, this has to be a unique file name. It is advised to not use spaces or umlauts for the name. The ⟨*options*⟩ are keys from the `/tcb/external/` key tree.

```
\begin{tcbexternal}{example_tikzpicture}
  \begin{tikzpicture}
    \path[fill=yellow!50!white] (0,0) circle (11mm);
    \path[fill=white] (0,0) circle (9mm);
    \foreach \w/\c in {90/red,210/green,330/blue}
      {\path[shading=ball,ball color=\c] (\w:1cm) circle (7mm);}
  \end{tikzpicture}
\end{tcbexternal}
```



If a `tcolorbox` → P. 12 is externalized, one should use `/tcb/nobeforeafter` → P. 76 for the box. Indention and distances to the text before and after have to be given separately outside the `tcbexternal` environment.

```
\noindent%
\begin{tcbexternal}[minipage]{example_tcolorbox}
  \begin{tcolorbox}[nobeforeafter,enhanced,
      fonttitle=\bfseries,title=Externalized Box,
      colframe=red!50!black,drop fuzzy shadow,
      interior style={fill overzoom image=goldshade.png}]
    This complete tcolorbox is externalized. One cannot use numbered
    boxes here. Note the \texttt{minipage} option which tells the
    current line width to the external snippet.
  \end{tcolorbox}
\end{tcbexternal}
```

Externalized Box

This complete tcolorbox is externalized. One cannot use numbered boxes here. Note the `minipage` option which tells the current line width to the external snippet.

```
\begin{tcolorbox}[nobeforeafter,enhanced,
      fonttitle=\bfseries,title=Externalized Box,
      colframe=blue!50!black,
      interior style={fill overzoom image=blueshade.png}]
  \begin{tcbexternal}[minipage]{example_tcolorbox2}
    \color{white}%
    The interior of the tcolorbox is externalized.
    One can use numbered boxes without problems.
    Note that the text color has to be set for the text manually
    since it is converted into an image.
  \end{tcbexternal}
\end{tcolorbox}
```

Externalized Box

The interior of the tcolorbox is externalized. One can use numbered boxes without problems. Note that the text color has to be set for the text manually since it is converted into an image.

```
\begin{tcbexternal}[minipage]{example_tabularx}
  \newcolumntype{Y}{>{\raggedleft\arraybackslash}X}%
  \begin{tabularx}{\linewidth}{|l||Y|Y|Y|Y||Y|}\hline
    Group & One & Two & Three & Four & Sum\\\hline\hline
    Red & 1000.00 & 2000.00 & 3000.00 & 4000.00 & 10000.00\\\hline
    Green & 2000.00 & 3000.00 & 4000.00 & 5000.00 & 14000.00\\\hline
    Blue & 3000.00 & 4000.00 & 5000.00 & 6000.00 & 18000.00\\\hline\hline
    Sum & 6000.00 & 9000.00 & 12000.00 & 15000.00 & 42000.00\\\hline
  \end{tabularx}
\end{tcbexternal}
```

Group	One	Two	Three	Four	Sum
Red	1000.00	2000.00	3000.00	4000.00	10000.00
Green	2000.00	3000.00	4000.00	5000.00	14000.00
Blue	3000.00	4000.00	5000.00	6000.00	18000.00
Sum	6000.00	9000.00	12000.00	15000.00	42000.00

N 2015-03-11 `/tcb/external/name`=⟨*name*⟩ (no default, initially `unnamed`)

The ⟨*name*⟩ is automatically prefixed with `/tcb/external/prefix` → P. 416. In combination, this has to be a unique file name for externalization. Typically, this key is not used directly but is set indirectly as mandatory parameter, see `tcbexternal` → P. 417.

N 2015-03-11

```
\begin{extcolorbox}[⟨options⟩]{⟨name⟩}[⟨tcolorbox options⟩]
  ⟨environment content⟩
\end{extcolorbox}
```

This is an externalized version of `tcolorbox` → P. 12 created using `\newtcbexternalizetcolorbox` → P. 424:

```
\newtcbexternalizetcolorbox{extcolorbox}{tcolorbox}{}{}
```

⟨*options*⟩ and ⟨*name*⟩ are given to the underlying `tcbexternal` → P. 417 environment, while ⟨*tcolorbox options*⟩ are given to `tcolorbox` → P. 12.

! Note that you should not redefine `/tcb/before` → P. 76 and `/tcb/after` → P. 76 inside the ⟨*tcolorbox options*⟩, since the externalized version would not be identical to the non-externalized otherwise.

```
\begin{extcolorbox}[minipage]{example_extcolorbox}
  [ enhanced,colframe=red!50!black,colback=yellow!10,
    fonttitle=\bfseries,drop fuzzy shadow,
    title=My external box ]

  This box is completely externalized.

  \begin{tcolorbox}[colframe=blue,colback=blue!5,before skip=6pt]
  Inner box.
  \end{tcolorbox}
\end{extcolorbox}
```

My external box

This box is completely externalized.

Inner box.

!
- **Never** externalize numbered boxes.
- **Never** externalize boxes which contain references to other things, e.g. using `\ref` or `\cite`.
- **Never** externalize breakable boxes.

N 2015-03-11

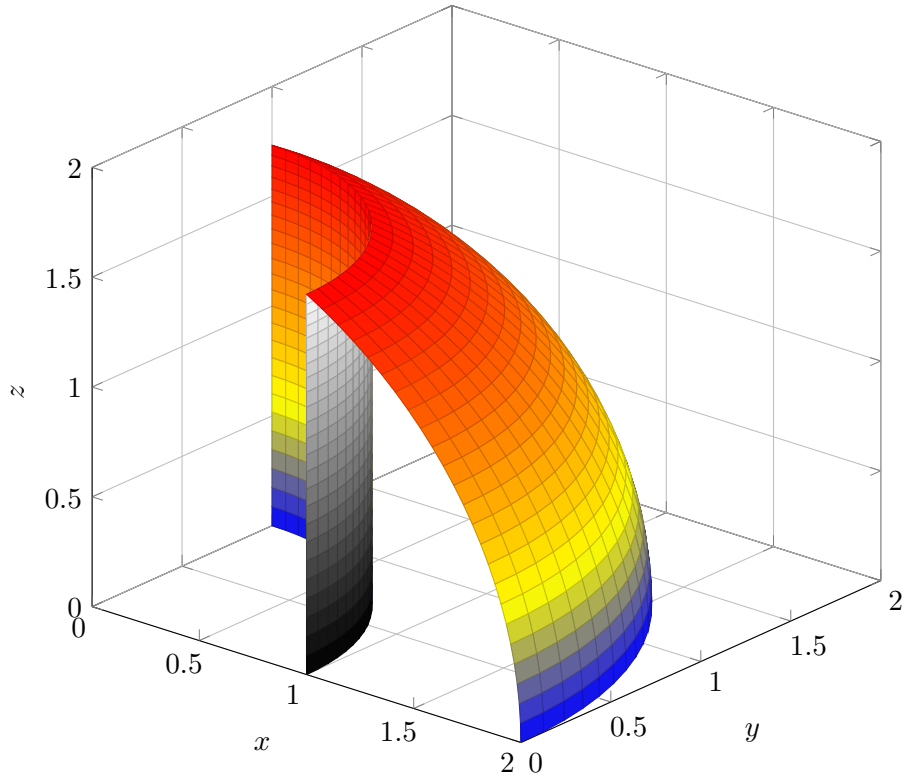
```
\begin{extikzpicture}[⟨options⟩]{⟨name⟩}[⟨tikz options⟩]
  ⟨environment content⟩
\end{extikzpicture}
```

This is an externalized version of `tikzpicture` created using `\newtcbexternalizeenvironment` → P. 424:

```
\newtcbexternalizeenvironment{extikzpicture}{tikzpicture}{}{}{}
```

⟨*options*⟩ and ⟨*name*⟩ are given to the underlying `tcbexternal` → P. 417 environment, while ⟨*tikz options*⟩ are given to `tikzpicture`.

```
\begin{center}
\begin{extikzpicture}[
  preamble={\usepackage{pgfplots}},  % add package for external graph
  input source on error=false,       % do not load source on error
]{example_pgfplots}
  \pgfplotsset{width=12cm}
  \begin{axis}[3d box=background,grid=major,
    xlabel=$x$, ylabel=$y$, zlabel=$z$, view/h=40,
    mesh/interior colormap name=hot,
    colormap/blackwhite,
    z buffer=sort,domain=0:90,y domain=0:60,
    zmin=0,zmax=2,z post scale=1.2,
    ]
  \addplot3[surf,mesh/interior colormap name=blackwhite,
    colormap/hot,] ( {cos(x)},{sin(x)}, {2*sin(y)} );
  \addplot3[surf] ( {2*cos(x)*cos(y)},{2*sin(x)*cos(y)}, {2*sin(y)} );
  \end{axis}
\end{extikzpicture}
\end{center}
```

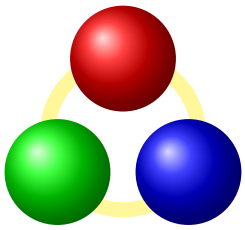


N 2015-03-11 **/tcb/externalize listing=**⟨*name*⟩ (style, no default)

The text content of a tcblisting → P. 287 is externalized with the given ⟨*name*⟩. Note that the listing part is not externalized.

```
\begin{tcblisting}{externalize listing=example_listing,
  bicolor,colback=yellow!10,colframe=yellow!50!black,
  colbacklower=white,center lower}
\begin{tikzpicture}
  \path[fill=yellow!50!white] (0,0) circle (11mm);
  \path[fill=white] (0,0) circle (9mm);
  \foreach \w/\c in {90/red,210/green,330/blue}
    {\path[shading=ball,ball color=\c] (\w:1cm) circle (7mm);}
\end{tikzpicture}
\end{tcblisting}
```

```
\begin{tikzpicture}
  \path[fill=yellow!50!white] (0,0) circle (11mm);
  \path[fill=white] (0,0) circle (9mm);
  \foreach \w/\c in {90/red,210/green,330/blue}
    {\path[shading=ball,ball color=\c] (\w:1cm) circle (7mm);}
\end{tikzpicture}
```



N 2015-03-11 **/tcb/externalize listing!=**⟨*name*⟩ (style, no default)

Combination of /tcb/externalize listing and /tcb/external/force remake → P. 416.

N 2015-03-11 **/tcb/externalize example=**⟨*name*⟩ (style, no default)

The text content of a dispExample* → P. 431 is externalized with the given ⟨*name*⟩. Note that the listing part is not externalized.

```
\begin{dispExample*}{sidebyside,externalize example=example_example}
\tikz\path[shading=ball,
  ball color=red] circle (7mm);
\end{dispExample*}
```

```
\tikz\path[shading=ball,
  ball color=red] circle (7mm);
```

N 2015-03-11 **/tcb/externalize example!=**⟨*name*⟩ (style, no default)

Combination of /tcb/externalize example and /tcb/external/force remake → P. 416.

22.3 Customization

N 2015-03-11 **/tcb/external/safety=⟨*length*⟩** (no default, initially 2mm)

The snippet box is surrounded with a safety border with a thickness of ⟨*length*⟩. This border is automatically trimmed during picture inclusion. The reason for this mechanism is to catch box content which extrudes over the bounding box. For example, shadows of a tcolorbox are painted outside the bounding box and would be lost otherwise.

N 2015-03-11 **/tcb/external/environment=⟨*env*⟩** (no default, initially unset)

Surrounds the exported snippet text with an environment ⟨*env*⟩ without parameters. Note that this option is ignored for /tcb/externalize listing → P. 421.

N 2015-05-05 **/tcb/external/environment with percent=true|false** (default true, initially true)

If set to true, the \begin and \end code of /tcb/external/environment is appended with a percent sign. For verbatim environments, this option typically has to be se to false.

N 2015-03-11 **/tcb/external/minipage=⟨*length*⟩** (default \linewidth, initially unset)

Surrounds the exported snippet text with a minipage. The optional ⟨*length*⟩ parameter sets the width of the minipage. Note that the default width is the current line width of the main document. See tcbexternal → P. 417 for examples. Note that this option is ignored for /tcb/externalize listing → P. 421.

N 2015-03-11 **/tcb/external/plain** (no value, initially set)

Removes any text which was set to surround the snippet. This removes the setting of /tcb/external/minipage, but is independent of /tcb/external/safety.

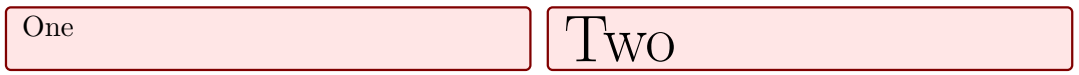
N 2015-03-11 **/tcb/external/compiler=⟨*text*⟩** (no default, initially pdflatex)

Sets the name of the compiler for the snippets. Note that this compiler has to support the \pdfmdfivesum primitive e.g. using the pdftexcmds package. This should work for xelatex and lualatex.

N 2015-03-11 **/tcb/external/runs=⟨*number*⟩** (no default, initially 1)

Sets the number of compiler runs for the snippet.

```
\begin{tcbexternal}[minipage,runs=2]{example_raster}
  \begin{tcbitemize}[raster equal height,
      size=small,colframe=red!50!black,colback=red!10!white]
    \tcbitem One
    \tcbitem \Huge Two
    \tcbitem Three
    \tcbitem Four
  \end{tcbitemize}
\end{tcbexternal}
```



N 2015-03-11 **/tcb/external/input source on error=true|false** (default true, initially true)

If set to true, the source code of the snippet is loaded instead of the failed pdf picture. Typically, this will lead to an error stop at the faulty place of the source and such helps detecting the cause. If the source input compiles without error, the document setup may be incorrect, see Section 22.1 on page 416. Maybe, the external/ subdirectory has to be created manually in this case, see /tcb/external/prefix → P. 416.

If the option is set to false, the compilation stops immediately on an error. The log file of the external snippet has to be consulted for error messages in this case.

N 2015-05-05 **/tcb/external/preclass=**⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is added before the snippet document. Typically, this means before `\documentclass`. This is not used for compilation of the main document.

N 2015-05-05 **/tcb/external/PassOptionsToPackage={**⟨*options*⟩**}{**⟨*package*⟩**}** (no default, initially unset)

The given ⟨*options*⟩ are passed to the given ⟨*package*⟩ for the snippet document. This is a shortcut for using /tcb/external/preclass with `\PassOptionsToPackage`. This not used for compilation of the main document.

N 2015-05-05 **/tcb/external/PassOptionsToClass={**⟨*options*⟩**}{**⟨*class*⟩**}** (no default, initially unset)

The given ⟨*options*⟩ are passed to the given ⟨*class*⟩ for the snippet document. This is a shortcut for using /tcb/external/preclass with `\PassOptionsToClass`. This not used for compilation of the main document.

N 2015-05-05 **/tcb/external/clear preclass** (no value)

Removes all additional /tcb/external/preclass settings.

N 2015-03-11 **/tcb/external/preamble=**⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is added to the preamble of the snippet document. This is not used for compilation of the main document.

N 2015-05-05 **/tcb/external/preamble tcbset=**⟨*options*⟩ (no default, initially unset)

The given ⟨*options*⟩ are added as parameter for `\tcbset` → P. 13 to the preamble of the snippet document. This are not used for compilation of the main document.

N 2015-03-16 **/tcb/external/clear preamble** (no value)

Removes all additional /tcb/external/preamble settings.

N 2015-03-11 **\tcbifexternal{**⟨*true*⟩**}{**⟨*false*⟩**}**

Expands to ⟨*true*⟩, if executed during snippet compilation, and to ⟨*false*⟩, if executed during main document compilation. This can be used *before* `\tcbEXTERNALIZE` → P. 416 to give different setting to snippet and main document.

```
\tcbifexternal{
  \usepackage{onlyforexternal}
}{
  \usepackage{onlyformain}
}
```

N 2015-03-11 **\newtcbexternalizeenvironment**{⟨*newenv*⟩}{⟨*env*⟩}{⟨*options*⟩}{⟨*begin*⟩}{⟨*end*⟩}

Creates a new environment ⟨*newenv*⟩ which is based on tcbexternal → P. 417. This enviroment takes *at least* one optional parameter and one mandatory parameter. These two parameters are passed to tcbexternal → P. 417. Further, the given ⟨*options*⟩ are always added to the option list of tcbexternal → P. 417.
The environment content is externalized and the external snippet is surrounded by an environment ⟨*env*⟩. All further parameters of ⟨*newenv*⟩ are given to ⟨*env*⟩ as parameters. The included image is prepended by ⟨*begin*⟩ and appended by ⟨*end*⟩.
extikzpicture → P. 420 is an example application for \newtcbexternalizeenvironment.

```
\newtcbexternalizeenvironment{extabular}{tabular}{}{\par\centering}{\par}

\begin{extabular}{example_tabular}{|l|p{6cm}|r|}\hline
A & B & C\\\hline
a & This table is externalized as snippet. Obviously,
  this only makes sense for highly complex tables.
& b\\\hline
\end{extabular}
```

A	B	C
a	This table is externalized as snippet. Obviously, this only makes sense for highly complex tables.	b

N 2015-03-11 **\renewtcbexternalizeenvironment**{⟨*newenv*⟩}{⟨*env*⟩}{⟨*options*⟩}{⟨*begin*⟩}{⟨*end*⟩}

Identical to \newtcbexternalizeenvironment, but the environment ⟨*newenv*⟩ is created by \renewenvironment instead of \newenvironment.

N 2015-03-11 **\newtcbexternalizetcolorbox**{⟨*newenv*⟩}{⟨*env*⟩}{⟨*options*⟩}{⟨*begin end options*⟩}

Creates a new environment ⟨*newenv*⟩ which is based on tcbexternal → P. 417. This enviroment takes *at least* one optional parameter and one mandatory parameter. These two parameters are passed to tcbexternal → P. 417. Further, the given ⟨*options*⟩ are always added to the option list of tcbexternal → P. 417.
The environment content is externalized and the external snippet is surrounded by an environment ⟨*env*⟩. All further parameters of ⟨*newenv*⟩ are given to ⟨*env*⟩ as parameters. **In contrast to \newtcbexternalizeenvironment, the environment ⟨*env*⟩ is intended to be based on tcolorbox → P. 12 or tcblisting → P. 287.**
The ⟨*begin end options*⟩ are options for settings the space before and after the included image using /tcb/before → P. 76, /tcb/before skip → P. 78, /tcb/after → P. 76, or /tcb/after skip → P. 78.

! Use the exact identical values for /tcb/before → P. 76 and /tcb/after → P. 76 inside ⟨*begin end options*⟩ as they where used for definition of ⟨*env*⟩! Otherwise, externalized and non-externalized version will have different spacings.

extcolorbox → P. 419 is an example application for\newtcbexternalizetcolorbox.

Definition in the preamble:

```
\newtcblisting{myownlisting}[2][]{
  enhanced,colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  colbacktitle=red!50!yellow,before skip=6pt,after skip=6pt,
  title={#2},#1}

\newtcbexternalizetcolorbox{exmyownlisting}{myownlisting}{minipage}%
  {before skip=6pt,after skip=6pt}% same values as for mylisting
```

```
\begin{exmyownlisting}{example_mylisting}% <- name for the external file
  {My externalized example box}
This is my \LaTeX\ box.
\end{exmyownlisting}
```

My externalized example box

This is my \LaTeX\ box.

This is my LaTeX box.

N 2015-03-11 `\renewtcbexternalizetcolorbox{⟨newenv⟩}{⟨env⟩}{⟨options⟩}{⟨begin end options⟩}`

Identical to `\newtcbexternalizetcolorbox` → P. 424, but the environment ⟨*newenv*⟩ is created by `\renewenvironment` instead of `\newenvironment`.

22.4 Troubleshooting and FAQ

- **I use the default settings, but the `external` subdirectory is not created.**
  Depending on operating system and compiler, an `external` subdirectory is automatically created or not. If not, create such a directory manually or add the following to your document:

```
\immediate\write18{mkdir external}
```

  or

```
\immediate\write18{mkdir -p external}
```

  If the combination of `/tcb/external/prefix` → P. 416 and chosen snippet name points to another subdirectory than `external`, this has to be adapted.

- **I use the `minted` package and I get a cache directory for every externalized snippet.**
  To avoid this problem, there are several ways.

  – If you do not need `minted` inside the snippet code, you may use `\usepackage{minted}` *after* `\tcbEXTERNALIZE` → P. 416 or use `\tcbifexternal` → P. 423 to switch `minted` off for the external code. If `minted` is already included by another package, add the following to your preamble:

```
\tcbset{external/PassOptionsToPackage={draft}{minted}}
```

  – If `minted` is needed for the snippet code, caching can be switched off by adding the following to your preamble:

```
\tcbset{external/PassOptionsToPackage={cache=false}{minted}}
```

    Alternatively, the `cachedir` option of `minted` may be used to redirect the cache.

23 Library documentation

This library has the single purpose to support LaTeX package documentations like this one. Actually, the visual nature follows the approach from Till Tantau's `pgf` [20] documentation. Typically, this library is assumed to be used in conjunction with the class `ltxdoc` or alike.

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{documentation}
```

This also loads the library listings, see Section 15 on page 286, the library skins, see Section 10 on page 139, the library xparse, see Section 21 on page 402, and a bunch of packages, namely `pifont`, `marvosym`, `makeidx`, `marginnote`, `refcount`, and `hyperref`.

N 2014-09-19

> ! The package `makeidx` is loaded only, if `\printindex` is *not* already defined. Therefore, one can include an alternative to `makeidx` like `imakeidx` *before* the library `documentation` is used.

> ! The package `marginnote` is loaded only, if `\marginnote` is *not* already defined.

For UTF-8 support, load:

```
\tcbuselibrary{listingsutf8,documentation}
```

For `minted` [11] support, load:

```
\tcbuselibrary{documentation,minted}
\tcbset{listing engine=minted}
```

23.1 Macros of the Library

`\begin{docCommand}[`⟨*options*⟩`]{`⟨*name*⟩`}{`⟨*parameters*⟩`}`
⟨*command description*⟩
`\end{docCommand}`

Documents a LaTeX macro with given ⟨*name*⟩ where ⟨*name*⟩ is written without backslash. The given ⟨*options*⟩ are set with `\tcbset` → P. 13. This macro takes mandatory or optional ⟨*parameters*⟩. It is automatically indexed and can be referenced with `\refCom` → P. 433`{`⟨*name*⟩`}`.

```
\begin{docCommand}{foomakedocSubKey}{\marg{name}\marg{key path}}
  Creates a new environment \meta{name} based on \refEnv{docKey} for the
  documentation of keys with the given \meta{key path}.
\end{docCommand}
```

`\foomakedocSubKey{`⟨*name*⟩`}{`⟨*key path*⟩`}`
Creates a new environment ⟨*name*⟩ based on `docKey` → P. 428 for the documentation of keys with the given ⟨*key path*⟩.

```
\begin{docCommand}[color definition=blue]{foomakedocSubKey*}%
    {\marg{name}\marg{key path}}
  Creates a new environment \meta{name} based on \refEnv{docKey} for the
  documentation of keys with the given \meta{key path}.
\end{docCommand}
```

\foomakedocSubKey*{⟨*name*⟩}{⟨*key path*⟩}
Creates a new environment ⟨*name*⟩ based on docKey → P. 428 for the documentation of keys with the given ⟨*key path*⟩.

\begin{docCommand*}[⟨*options*⟩]{⟨*name*⟩}{⟨*parameters*⟩}
⟨*command description*⟩
\end{docCommand*}

Identical to docCommand → P. 426, but without index entry.

\begin{docEnvironment}[⟨*options*⟩]{⟨*name*⟩}{⟨*parameters*⟩}
⟨*environment description*⟩
\end{docEnvironment}

Documents a LaTeX environment with given ⟨*name*⟩. The given ⟨*options*⟩ are set with \tcbset → P. 13. This environment takes mandatory or optional ⟨*parameters*⟩. It is automatically indexed and can be referenced with \refEnv → P. 433{⟨*name*⟩}.

```
\begin{docEnvironment}{foocolorbox}{\oarg{options}}
  This is the main environment to create an accentuated colored text box with
  rounded corners and, optionally, two parts.
\end{docEnvironment}
```

\begin{foocolorbox}[⟨*options*⟩]
⟨*environment content*⟩
\end{foocolorbox}
This is the main environment to create an accentuated colored text box with rounded corners and, optionally, two parts.

```
\begin{docEnvironment}%
    [doclang/environment content=My content text]%
    {foocolorbox*}{\oarg{options}}
  This is the main environment to create an accentuated colored text box with
  rounded corners and, optionally, two parts.
\end{docEnvironment}
```

\begin{foocolorbox*}[⟨*options*⟩]
⟨*My content text*⟩
\end{foocolorbox*}
This is the main environment to create an accentuated colored text box with rounded corners and, optionally, two parts.

\begin{docEnvironment*}[⟨*options*⟩]{⟨*name*⟩}{⟨*parameters*⟩}
⟨*environment description*⟩
\end{docEnvironment*}

Identical to docEnvironment, but without index entry.

U 2014-09-29 `\begin{docKey}[⟨key path⟩][⟨options⟩]{⟨name⟩}{⟨parameters⟩}{⟨description⟩}`
`⟨key description⟩`
`\end{docKey}`

Documents a key with given ⟨*name*⟩ and an optional ⟨*key path*⟩. The given ⟨*options*⟩ are set with `\tcbset` → P. 13. This key takes mandatory or optional ⟨*parameters*⟩ as value with a short ⟨*description*⟩. It is automatically indexed and can be referenced with `\refKey` → P. 434`{⟨name⟩}`.

```
\begin{docKey}[foo]{footitle}{=\meta{text}}{no default, initially empty}
  Creates a heading line with \meta{text} as content.
\end{docKey}
```

/foo/footitle=⟨*text*⟩ (no default, initially empty)
Creates a heading line with ⟨*text*⟩ as content.

U 2014-09-29 `\begin{docKey*}[⟨key path⟩][⟨options⟩]{⟨name⟩}{⟨parameters⟩}{⟨description⟩}`
`⟨key description⟩`
`\end{docKey*}`

Identical to `docKey`, but without index entry.

`\docValue{⟨name⟩}`

Documents a value with given ⟨*name*⟩. Typically, this is a value for a key. This value is automatically indexed.

```
A feasible value for \refKey{/foo/footitle} is \docValue{foovalue}.
```

A feasible value for /foo/footitle is foovalue.

`\docValue*{⟨name⟩}`

Identical to `\docValue`, but without index entry.

U 2014-10-09 `\docAuxCommand{⟨name⟩}`

Documents an auxiliary or minor LaTeX macro with given ⟨*name*⟩ where ⟨*name*⟩ is written without backslash. This macro is automatically indexed.

```
The macro \docAuxCommand{fooaux} holds some interesting data.
```

The macro \fooaux holds some interesting data.

U 2014-10-09 `\docAuxCommand*{⟨name⟩}`

Identical to `\docAuxCommand`, but without index entry.

`\docAuxEnvironment{⟨name⟩}`

Documents an auxiliary or minor LaTeX environment with given ⟨*name*⟩. This macro is automatically indexed.

```
The environment \docAuxEnvironment{fooauxenv} holds some interesting data.
```

The environment fooauxenv holds some interesting data.

`\docAuxEnvironment*[⟨key path⟩]{⟨name⟩}`

Identical to `\docAuxEnvironment`, but without index entry.

`\docAuxKey[`⟨*key path*⟩`]{`⟨*name*⟩`}`

Documents an auxiliary key with given ⟨*name*⟩ and an optional ⟨*key path*⟩. It is automatically indexed.

```
The key \docAuxKey[foo]{fooaux} holds some interesting data.
```

The key /foo/fooaux holds some interesting data.

`\docAuxKey*[`⟨*key path*⟩`]{`⟨*name*⟩`}`

Identical to \docAuxKey, but without index entry.

N 2015-01-08 `\docCounter{`⟨*name*⟩`}`

Documents a counter with given ⟨*name*⟩. The counter is automatically indexed.

```
The counter \docCounter{foocounter} can be used for computation.
```

The counter foocounter can be used for computation.

N 2015-01-08 `\docCounter*{`⟨*name*⟩`}`

Identical to \docCounter, but without index entry.

N 2015-01-08 `\docLength{`⟨*name*⟩`}`

Documents a counter with given ⟨*name*⟩. The counter is automatically indexed.

```
The length \docLength{foolength} can be used for computation.
```

The length \foolength can be used for computation.

N 2015-01-08 `\docLength*{`⟨*name*⟩`}`

Identical to \docLength, but without index entry.

`\docColor{`⟨*name*⟩`}`

Documents a color with given ⟨*name*⟩. The color is automatically indexed.

```
The color \docColor{foocolor} is available.
```

The color foocolor is available.

`\docColor*{`⟨*name*⟩`}`

Identical to \docColor, but without index entry.

`\cs{`⟨*name*⟩`}`

Macro from `ltxdoc` [3] to typeset a command word ⟨*name*⟩ where the backslash is prefixed. The library overwrites the original macro.

```
This is a \cs{foocommand}.
```

This is a \foocommand.

`\meta{`⟨*text*⟩`}`

Macro from `doc` [8] to typeset a meta ⟨*text*⟩. The library overwrites the original macro.

```
This is a \meta{text}.
```

This is a ⟨*text*⟩.

`\marg{`⟨*text*⟩`}`

Macro from `ltxdoc` [3] to typeset a ⟨*text*⟩ with curly brackets as a mandatory argument. The library overwrites the original macro.

```
This is a mandatory \marg{argument}.
```

This is a mandatory {⟨*argument*⟩}.

`\oarg{`⟨*text*⟩`}`

Macro from `ltxdoc` [3] to typeset a ⟨*text*⟩ with square brackets as an optional argument. The library overwrites the original macro.

```
This is an optional \oarg{argument}.
```

This is an optional [⟨*argument*⟩].

`\brackets{`⟨*text*⟩`}`

Sets the given ⟨*text*⟩ with curly brackets.

```
  Here we use \brackets{some text}.
```

Here we use {some text}.

U 2014-10-10 `\begin{dispExample}`
⟨*environment content*⟩
`\end{dispExample}`

Creates a colored box based on a `tcolorbox` → P. 12. It displays the environment content as source code in the upper part and as compiled text in the lower part of the box. The appearance is controlled by `/tcb/documentation listing style` → P. 436 and the style `/tcb/docexample` → P. 436. It may be changed by redefining this style.

```
\begin{dispExample}
This is a \LaTeX\ example.
\end{dispExample}
```

```
This is a \LaTeX\ example.
```

This is a LaTeX example.

U 2014-10-10 `\begin{dispExample*}`{⟨*options*⟩}
⟨*environment content*⟩
`\end{dispExample*}`

The starred version of `dispExample` takes `tcolorbox` → P. 12 ⟨*options*⟩ as parameter. These ⟨*options*⟩ are executed after `/tcb/docexample` → P. 436.

```
\begin{dispExample*}{sidebyside}
This is a \LaTeX\ example.
\end{dispExample*}
```

```
This is a \LaTeX\ example.
```

This is a LaTeX example.

\begin{**displListing**}
⟨*environment content*⟩
\end{**displListing**}

Creates a colored box based on a `tcolorbox` → P. 12. It displays the environment content as source code. The appearance is controlled by `/tcb/documentation listing style` → P. 436 and the style `/tcb/docexample` → P. 436. It may be changed by redefining this style.

```
\begin{displListing}
This is a \LaTeX\ example.
\end{displListing}
```

```
This is a \LaTeX\ example.
```

\begin{**displListing***}{⟨*options*⟩}
⟨*environment content*⟩
\end{**displListing***}

The starred version of `displListing` takes `tcolorbox` → P. 12 ⟨*options*⟩ as parameter. These ⟨*options*⟩ are executed after `/tcb/docexample` → P. 436.

```
\begin{displListing*}{title=My listing}
This is a \LaTeX\ example.
\end{displListing*}
```

My listing

```
This is a \LaTeX\ example.
```

\begin{**absquote**}
⟨*environment content*⟩
\end{**absquote**}

Used to typeset an abstract as quoted and small text.

```
\begin{absquote}
|tcolorbox| provides an environment for colored and framed text boxes with a
heading line. Optionally, such a box can be split in an upper and a lower part.
\end{absquote}
```

> `tcolorbox` provides an environment for colored and framed text boxes with a heading line. Optionally, such a box can be split in an upper and a lower part.

`\tcbmakedocSubKey{`⟨*name*⟩`}{`⟨*key path*⟩`}`

Creates a new environment ⟨*name*⟩ based on docKey → P. 428 for the documentation of keys with the given ⟨*key path*⟩ as default. The new environment ⟨*name*⟩ takes the same parameters as docKey → P. 428 itself. A second starred environment ⟨*name*⟩ is also created, which is identical to ⟨*name*⟩ but without index entry.

```
\tcbmakedocSubKey{docFooKey}{foo}

\begin{docFooKey}{foodummy}{=\meta{nothing}}{no default, initially empty}
Some key.
\end{docFooKey}

\begin{docFooKey*}{foo another dummy}{=\meta{nothing}}{no default, initially empty}
Some key (not indexed).
\end{docFooKey*}
```

/foo/foodummy=⟨*nothing*⟩ (no default, initially empty)
Some key.

/foo/foo another dummy=⟨*nothing*⟩ (no default, initially empty)
Some key (not indexed).

`\refCom{`⟨*name*⟩`}`

References a documented LaTeX macro with given ⟨*name*⟩ where ⟨*name*⟩ is written without backslash. The page reference is suppressed if it links to the same page.

```
We have created \refCom{foomakedocSubKey} as an example.
```

We have created \foomakedocSubKey → P. 426 as an example.

`\refCom*{`⟨*name*⟩`}`

References a documented LaTeX macro with given ⟨*name*⟩ where ⟨*name*⟩ is written without backslash. There is no page reference.

```
We have created \refCom*{foomakedocSubKey} as an example.
```

We have created \foomakedocSubKey as an example.

`\refEnv{`⟨*name*⟩`}`

References a documented LaTeX environment with given ⟨*name*⟩. The page reference is suppressed if it links to the same page.

```
We have created \refEnv{foocolorbox} as an example.
```

We have created foocolorbox → P. 427 as an example.

`\refEnv*{`⟨*name*⟩`}`

References a documented LaTeX environment with given ⟨*name*⟩. There is no page reference.

```
We have created \refEnv*{foocolorbox} as an example.
```

We have created foocolorbox as an example.

`\refKey{`⟨*name*⟩`}`

References a documented key with given ⟨*name*⟩ where ⟨*name*⟩ is the full path name of the key. The page reference is suppressed if it links to the same page.

```
We have created \refKey{/foo/footitle} as an example.
```

We have created /foo/footitle → P. 428 as an example.

`\refKey*{`⟨*name*⟩`}`

References a documented key with given ⟨*name*⟩ where ⟨*name*⟩ is the full path name of the key. There is no page reference.

```
We have created \refKey*{/foo/footitle} as an example.
```

We have created /foo/footitle as an example.

`\refAux{`⟨*name*⟩`}`

References some auxiliary environment, key, value, or color. The hyperlink color is used, but there is no real link.

```
Some pages back, one can see \refAux{/foo/footitle} as an example.
```

Some pages back, one can see /foo/footitle as an example.

`\refAuxcs{`⟨*name*⟩`}`

References some auxiliary macro ⟨*name*⟩ where ⟨*name*⟩ is written without backslash. The hyperlink color is used, but there is no real link.

```
Some pages back, one can see \refAuxcs{fooaux} as an example.
```

Some pages back, one can see \fooaux as an example.

`\colDef{`⟨*text*⟩`}`

Sets ⟨*text*⟩ with the command color, see /tcb/color command → P. 438.

```
This is my \colDef{text}.
```

This is my text.

`\colOpt{`⟨*text*⟩`}`

Sets ⟨*text*⟩ with the option color, see /tcb/color option → P. 438.

```
This is my \colOpt{text}.
```

This is my text.

N 2014-09-19 `\tcbdocmarginnote[⟨options⟩]{⟨text⟩}`

Creates a `tcolorbox` note with the given ⟨*text*⟩ inside the margin using the `marginnote` package. The style of the `tcolorbox` is predefined and can be altered by `/tcb/docmarginnote` → P. 443 and the given ⟨*options*⟩.

```
Some text\tcbdocmarginnote{Note A}
which is commented by a note inside the margin.
Alternatively to |\tcbdocmarginnote|, you can always use
|\marginnote| with a |tcolorbox| directly.\par
This is further text%
\tcbdocmarginnote[colframe=blue!50!white,colback=blue!5!white]{Note B}
with another note.
```

Note A

Some text which is commented by a note inside the margin. Alternatively to `\tcbdocmarginnote`, you can always use `\marginnote` with a `tcolorbox` directly.

Note B

This is further text with another note.

N 2014-09-19 `\tcbdocnew{⟨date⟩}`

Auxiliary macro which typesets the `/tcb/doclang/new` → P. 439 text with the given ⟨*date*⟩. It may be redefined for customization.

```
\tcbdocnew{1981-10-29}.
% Next one is displayed in the margin:
\tcbdocmarginnote{\tcbdocnew{1978-02-09}}
```

New: 1981-10-29.

New: 1978-02-09

N 2014-09-19 `\tcbdocupdated{⟨date⟩}`

Auxiliary macro which typesets the `/tcb/doclang/updated` → P. 439 text with the given ⟨*date*⟩. It may be redefined for customization.

```
\tcbdocupdated{2014-09-19}.
```

Updated: 2014-09-19.

23.2 Option Keys of the Library

U 2015-03-16 **/tcb/docexample** (style, no value)

Sets the style for dispExample → P. 431 and dispListing → P. 432 with the colors ExampleBack and ExampleFrame. To change the appearance of the examples, this style can be redefined.

```
% Predefined style:
\tcbset{
  docexample/.style={colframe=ExampleFrame,colback=ExampleBack,
    before skip=\medskipamount,after skip=\medskipamount,
    fontlower=\footnotesize}
}
```

/tcb/documentation listing options=⟨*key list*⟩ (no default, initially **style=tcbdocumentation**)

Sets the options from the package **listings** [6]. They are used inside dispExample → P. 431 and dispListing → P. 432 to typeset the listings. Note that this is not identical to the key /tcb/listing options → P. 293 which is used for 'normal' listings.
Used for /tcb/listing engine → P. 298**=listings** only.

/tcb/documentation listing style=⟨*listing style*⟩ (no default, initially **tcbdocumentation**)

Abbreviation for **documentation listing options={style=...}**. This key sets a ⟨*style*⟩ for the **listings** package, see [6]. Note that this is not identical to the key /tcb/listing style → P. 293 which is used for 'normal' listings.
Used for /tcb/listing engine → P. 298**=listings** only.

/tcb/documentation minted style=⟨*key list*⟩ (no default, initially unset)

Sets a ⟨*style*⟩ known to **Pygments** [13] for the package **minted** [11], if used. Note that this is not identical to the key /tcb/minted style → P. 297 which is used for 'normal' listings.
Used for /tcb/listing engine → P. 298**=minted** only.

/tcb/documentation minted options=⟨*minted style*⟩ (no default, initially **tabsize=2,fontsize=\small**)

Sets the options from the package **minted** [11] which are used during typesetting of the listing, if used. Note that this is not identical to the key /tcb/minted options → P. 296 which is used for 'normal' listings.
Used for /tcb/listing engine → P. 298**=minted** only.

! The following two keys are deprecated and without function (v3.50 and above). Use /tcb/before → P. 76 and /tcb/after → P. 76 with appropriate values instead. Also see /tcb/docexample.

U 2015-03-16 **/tcb/before example=**⟨*macros*⟩ (no default, initially empty)

Sets the ⟨*macros*⟩ which are executed before dispExample → P. 431 and dispListing → P. 432 additional to /tcb/before → P. 76.

/tcb/after example=⟨*macros*⟩ (no default, initially empty)

Sets the ⟨*macros*⟩ which are executed after dispExample → P. 431 and dispListing → P. 432 additional to /tcb/after → P. 76.

N 2015-01-09 **/tcb/index command=**⟨*macro*⟩ (no default, initially `\index`)

Replaces the internally used `\index` macro by the given ⟨*macro*⟩. The ⟨*macro*⟩ has to take one mandatory argument like `\index`. This option is mutually exclusive with /tcb/index command name.

```
\tcbset{index command=\myindexcommand}
```

N 2015-01-09 **/tcb/index command name=**⟨*name*⟩ (no default, initially unset)

Replaces the internally used `\index` macro by `\index[`⟨*name*⟩`]`, i.e. `\index{...}` is replaced by `\index[`⟨*name*⟩`]{...}`. This option is intended to be used with `imakeidx` and is mutually exclusive with /tcb/index command.

```
\tcbset{index command name=mydoc}
```

/tcb/index format=⟨*format*⟩ (no default, initially `pgf`)

Determines the basic ⟨*format*⟩ of the generated index. Feasible values are:

- `pgfsection`: The index is formatted like in the `pgf` documentation (as a section).
- `pgfchapter`: The index is formatted like in the `pgf` documentation (as a chapter).
- `pgf`: Alias for `pgfsection`.
- `doc`: The index is assumed to be formatted by `doc` or `ltxdoc`. The usage of `makeindex` with `-s gind.ist` is assumed. The package `hypdoc` has to be loaded *before* `tcolorbox`.
- `off`: The index is not formatted by `tcolorbox`. Use this, if the index is formatted by other package like `imakeidx`.

/tcb/index actual=⟨*character*⟩ (no default, initially `@`)

Sets the character for 'actual' in automatic indexing.

/tcb/index quote=⟨*character*⟩ (no default, initially `"`)

Sets the character for 'quote' in automatic indexing.

/tcb/index level=⟨*character*⟩ (no default, initially `!`)

Sets the character for 'level' in automatic indexing.

/tcb/index default settings (style, no value)

Sets the `makeindex` default values for /tcb/index actual, /tcb/index quote, and /tcb/index level.

/tcb/index german settings (style, no value)

Sets the `makeindex` values recommended for German language texts. This is identical to setting the following:

```
\tcbset{index actual={=},index quote={!},index level={>}}
```

`/tcb/index annotate=true|false` (default `true`, initially `true`)

If set to `true`, the index entries are annotated with short descriptions given by `/tcb/doclang/environment` → P. 439, `/tcb/doclang/key` → P. 439, and others.

`/tcb/index colorize=true|false` (default `true`, initially `false`)

If set to `true`, the index entries colorized according to the color settings given by `/tcb/color environment`, `/tcb/color key`, and others.

`/tcb/color command=`⟨*color*⟩ (no default, initially `Definition`)

Sets the highlight color used by macro definitions.

`/tcb/color environment=`⟨*color*⟩ (no default, initially `Definition`)

Sets the highlight color used by environment definitions.

`/tcb/color key=`⟨*color*⟩ (no default, initially `Definition`)

Sets the highlight color used by key definitions.

`/tcb/color value=`⟨*color*⟩ (no default, initially `Definition`)

Sets the highlight color used by value definitions.

N 2015-01-08 `/tcb/color counter=`⟨*color*⟩ (no default, initially `Definition`)

Sets the highlight color used by counter definitions.

N 2015-01-08 `/tcb/color length=`⟨*color*⟩ (no default, initially `Definition`)

Sets the highlight color used by length definitions.

`/tcb/color color=`⟨*color*⟩ (no default, initially `Definition`)

Sets the highlight color used by color definitions.

`/tcb/color definition=`⟨*color*⟩ (no default, initially `Definition`)

Sets the highlight color for `/tcb/color command`, `/tcb/color environment`, `/tcb/color key`, `/tcb/color value`, `/tcb/color counter`, `/tcb/color length`, and `/tcb/color color`.

`/tcb/color option=`⟨*color*⟩ (no default, initially `Option`)

Sets the color used for optional arguments.

`/tcb/color hyperlink=`⟨*color*⟩ (no default, initially `Hyperlink`)

Sets the color for all hyper-links, i. e. all internal and external links.

The following keys are provided for language specific settings. The English language is predefined.

`/tcb/english language` (style, no value)
Sets all language specific settings to English.

`/tcb/doclang/color=`⟨*text*⟩ (no default, initially `color`)
Text used in the index for colors.

`/tcb/doclang/colors=`⟨*text*⟩ (no default, initially `Colors`)
Heading text in the index for colors.

N 2015-01-08 `/tcb/doclang/counter=`⟨*text*⟩ (no default, initially `counter`)
Text used in the index for counters.

N 2015-01-08 `/tcb/doclang/counters=`⟨*text*⟩ (no default, initially `Counters`)
Heading text in the index for counters.

`/tcb/doclang/environment=`⟨*text*⟩ (no default, initially `environment`)
Text used in the index for environments.

`/tcb/doclang/environments=`⟨*text*⟩ (no default, initially `Environments`)
Heading text in the index for environments.

`/tcb/doclang/environment content=`⟨*text*⟩ (no default, initially `environment content`)
Text used in `docEnvironment` → P. 427.

`/tcb/doclang/index=`⟨*text*⟩ (no default, initially `Index`)
Heading text for the index.

`/tcb/doclang/key=`⟨*text*⟩ (no default, initially `key`)
Text used in the index for keys.

`/tcb/doclang/keys=`⟨*text*⟩ (no default, initially `Keys`)
Heading text used in the index for keys.

N 2015-01-08 `/tcb/doclang/length=`⟨*text*⟩ (no default, initially `length`)
Text used in the index for lengths.

N 2015-01-08 `/tcb/doclang/lengths=`⟨*text*⟩ (no default, initially `Lengths`)
Heading text in the index for lengths.

N 2014-09-19 `/tcb/doclang/new=`⟨*text*⟩ (no default, initially `New`)
Announcement text for new content.

`/tcb/doclang/pageshort=`⟨*text*⟩ (no default, initially `P.`)
Short text for page references.

N 2014-09-19 `/tcb/doclang/updated=`⟨*text*⟩ (no default, initially `Updated`)
Announcement text for updated content.

`/tcb/doclang/value=`⟨*text*⟩ (no default, initially `value`)
Text used in the index for values.

`/tcb/doclang/values=`⟨*text*⟩ (no default, initially `Values`)
Heading text in the index for values.

/tcb/doc left=⟨*length*⟩ (no default, initially 2em)

Sets the left hand offset of the documentation texts from docCommand → P. 426, docEnvironment → P. 427, docKey → P. 428, etc, to ⟨*length*⟩.

```
\begin{docCommand*}[doc left=2cm,doc left indent=-2cm]{myCommandA}{\marg{argment}}
  This is the documentation of \refCom{myCommandA} which takes one \meta{argument}.
  \refCom{myCommandA} does some funny things with its \meta{argument}.
\end{docCommand*}
```

\myCommandA{⟨*argment*⟩}
This is the documentation of \myCommandA which takes one ⟨*argument*⟩. \myCommandA does some funny things with its ⟨*argument*⟩.

/tcb/doc right=⟨*length*⟩ (no default, initially 0em)

Sets the right hand offset of the documentation texts from docCommand → P. 426, docEnvironment → P. 427, docKey → P. 428, etc, to ⟨*length*⟩.

```
\begin{docCommand*}[doc right=2cm]{myCommandB}{\marg{argment}}
  This is the documentation of \refCom{myCommandB} which takes one \meta{argument}.
  \refCom{myCommandB} does some funny things with its \meta{argument}.
\end{docCommand*}
```

\myCommandB{⟨*argment*⟩}
This is the documentation of \myCommandB which takes one ⟨*argument*⟩. \myCommandB does some funny things with its ⟨*argument*⟩.

/tcb/doc left indent=⟨*length*⟩ (no default, initially -2em)

Sets the left hand indent of documentation heads from docCommand → P. 426, docEnvironment → P. 427, docKey → P. 428, etc, to ⟨*length*⟩.

```
\begin{docCommand*}[doc left indent=2cm]{myCommandC}{\marg{argment}}
  This is the documentation of \refCom{myCommandC} which takes one \meta{argument}.
  \refCom{myCommandC} does some funny things with its \meta{argument}.
\end{docCommand*}
```

\myCommandC{⟨*argment*⟩}
This is the documentation of \myCommandC which takes one ⟨*argument*⟩. \myCommandC does some funny things with its ⟨*argument*⟩.

/tcb/doc right indent=⟨*length*⟩ (no default, initially 0pt)

Sets the right hand indent of documentation heads from docCommand → P. 426, docEnvironment → P. 427, docKey → P. 428, etc, to ⟨*length*⟩.

```
\begin{docCommand*}[doc right indent=-10mm,doc right=10mm,
    doc description=test value]{myCommandD}{\marg{argment}}
  This is the documentation of \refCom{myCommandD} which takes one \meta{argument}.
  \refCom{myCommandD} does some funny things with its \meta{argument}.
\end{docCommand*}
```

\myCommandD{⟨*argment*⟩} (test value)
This is the documentation of \myCommandD which takes one ⟨*argument*⟩. \myCommandD does some funny things with its ⟨*argument*⟩.

The head lines of the main documentation environments docCommand → P. 426, docEnvironment → P. 427, docKey → P. 428, etc, are set inside tcolorboxes. Options to these tcolorboxes can be given using the following keys.

/tcb/doc head command=⟨*options*⟩ (no default, initially empty)

Sets ⟨*options*⟩ for the head line of docCommand → P. 426 and docCommand* → P. 427.

```
\tcbset{doc head command={interior style={fill,left color=red!20!white,
  right color=blue!20!white}}}

\begin{docCommand*}{myCommandE}{\marg{argment}}
  This is the documentation of \refCom{myCommandE} which takes one \meta{argument}.
  \refCom{myCommandE} does some funny things with its \meta{argument}.
\end{docCommand*}
```

\myCommandE{⟨*argment*⟩}

This is the documentation of \myCommandE which takes one ⟨*argument*⟩. \myCommandE does some funny things with its ⟨*argument*⟩.

/tcb/doc head environment=⟨*options*⟩ (no default, initially empty)

Sets ⟨*options*⟩ for the head line of docEnvironment → P. 427 and docEnvironment* → P. 427.

```
\tcbset{doc head environment={beamer,boxsep=2pt,arc=2pt,colback=green!20!white,
  after=\par\smallskip}}

\begin{docEnvironment*}{myEnvironment}{\marg{argment}}
  This is the documentation of \refEnv{myEnvironment} which
  takes one \meta{argument}.
\end{docEnvironment*}
```

```
\begin{myEnvironment}{⟨argment⟩}
 ⟨environment content⟩
\end{myEnvironment}
```

This is the documentation of myEnvironment which takes one ⟨*argument*⟩.

/tcb/doc head key=⟨*options*⟩ (no default, initially empty)

Sets ⟨*options*⟩ for the head line of docKey → P. 428 and docKey* → P. 428.

```
\tcbset{doc head key={boxsep=4pt,arc=4pt,boxrule=0.6pt,
  frame style=fill,interior style=fill,colframe=green!50!black}}

\begin{docKey*}{/foo/myKey}{}{no value}
  This is the documentation of \refKey{/foo/myKey}.
\end{docKey*}
```

/foo/myKey (no value)

This is the documentation of /foo/myKey.

/tcb/doc head=⟨*options*⟩ (no default, initially empty)

Shortcut for setting the same ⟨*options*⟩ for /tcb/doc head command, /tcb/doc head environment, and /tcb/doc head key.

The description texts of the main documentation environments docCommand → P. 426, docEnvironment → P. 427, docKey → P. 428, etc, are set in a compact form without indention and parskip=0pt. This settings can overruled by using the following keys to insert code before (or after) the description texts.

N 2015-10-09 **/tcb/before doc body command=⟨*code*⟩** (no default, initially empty)

Executes ⟨*code*⟩ before the description texts of docCommand → P. 426 and docCommand* → P. 427.

```
\tcbset{before doc body command={%
    \setlength{\parindent}{2.5em}%
    \setlength{\parskip}{1ex plus 0.75ex minus 0.25ex}%
}}

\begin{docCommand*}{myCommandG}{\marg{argment}}
  This is the documentation of \refCom{myCommandG} which takes one \meta{argument}.
  \refCom{myCommandG} does some funny things with its \meta{argument}.
\end{docCommand*}
```

\myCommandG{⟨*argment*⟩}

This is the documentation of \myCommandG which takes one ⟨*argument*⟩. \myCommandG does some funny things with its ⟨*argument*⟩.

N 2015-10-09 **/tcb/after doc body command=⟨*code*⟩** (no default, initially empty)

Executes ⟨*code*⟩ after the description texts of docCommand → P. 426 and docCommand* → P. 427.

```
\tcbset{after doc body command={%
    \hfill\nolinebreak[1]\hspace*{\fill}\textcolor{red}{$\diamondsuit$}%
}}

\begin{docCommand*}{myCommandH}{\marg{argment}}
  This is the documentation of \refCom{myCommandH} which takes one \meta{argument}.
  \refCom{myCommandH} does some funny things with its \meta{argument}.
\end{docCommand*}
```

\myCommandH{⟨*argment*⟩}

This is the documentation of \myCommandH which takes one ⟨*argument*⟩. \myCommandH does some funny things with its ⟨*argument*⟩. ♢

N 2015-10-09 **/tcb/before doc body environment=⟨*code*⟩** (no default, initially empty)

Executes ⟨*code*⟩ before the description texts of docEnvironment → P. 427 and docEnvironment* → P. 427.

N 2015-10-09 **/tcb/after doc body environment=⟨*code*⟩** (no default, initially empty)

Executes ⟨*code*⟩ after the description texts of docEnvironment → P. 427 and docEnvironment* → P. 427.

N 2015-10-09 **/tcb/before doc body key=⟨*code*⟩** (no default, initially empty)

Executes ⟨*code*⟩ before the description texts of docKey → P. 428 and docKey* → P. 428.

N 2015-10-09 **/tcb/after doc body key=⟨*code*⟩** (no default, initially empty)

Executes ⟨*code*⟩ after the description texts of docKey → P. 428 and docKey* → P. 428.

N 2015-10-09 **/tcb/before doc body=⟨*options*⟩** (no default, initially empty)

Shortcut for setting the same ⟨*options*⟩ for /tcb/before doc body command, /tcb/before doc body environment, and /tcb/before doc body key.

N 2015-10-09 **/tcb/after doc body=⟨*options*⟩** (no default, initially empty)

Shortcut for setting the same ⟨*options*⟩ for /tcb/after doc body command, /tcb/after doc body environment, and /tcb/after doc body key.

`/tcb/doc description=`⟨*text*⟩ (no default, initially empty)

Sets a (short!) additional description ⟨*text*⟩ for `docCommand` → P. 426 or `docEnvironment` → P. 427. Such a description is mandatory for `docKey` → P. 428.

```
\begin{docCommand*}[doc description=my description]{myCommandF}{\marg{argment}}
  This is the documentation of \refCom{myCommandF} which takes one \meta{argument}.
  \refCom{myCommandF} does some funny things with its \meta{argument}.
\end{docCommand*}
```

`\myCommandF{`⟨*argment*⟩`}` (my description)

This is the documentation of `\myCommandF` which takes one ⟨*argument*⟩. `\myCommandF` does some funny things with its ⟨*argument*⟩.

! Note that the description ⟨*text*⟩ may overlap with the text on the left hand side if too long. Linebreaks can be used inside the ⟨*text*⟩.

`/tcb/doc into index=true|false` (default `true`, initially `true`)

If set to `false`, no index entries are written for the main documentation environments. The same effect is achieved by using e. g. `docCommand*` → P. 427 instead of `docCommand` → P. 426.

N 2014-09-19 `/tcb/doc marginnote=`⟨*options*⟩ (no default, initially empty)

Sets style ⟨*options*⟩ for the displayed box of the `\tcbdocmarginnote` → P. 435 command.

```
\tcbset{doc marginnote={colframe=blue!50!white,colback=blue!5!white}}%
This is some text\tcbdocmarginnote{Note A}
which is commented by a note inside the margin.
```

Note A

This is some text which is commented by a note inside the margin.

N 2014-09-19 `/tcb/doc new=`⟨*date*⟩ (style, no default)

Adds a a marginnote with a 'New: ⟨*data*⟩' message at the beginning of the upper box part. The intended use is inside the option list of `docCommand` → P. 426, `docEnvironment` → P. 427, etc.

```
\begin{docCommand}[doc new=2000-01-01]{foosomething}{\marg{text}}
Some command for something.
\end{docCommand}
```

New: 2000-01-01

`\foosomething{`⟨*text*⟩`}`

Some command for something.

N 2014-09-19 `/tcb/doc updated=`⟨*date*⟩ (style, no default)

Adds a marginnote with a 'Updated: ⟨*data*⟩' message at the beginning of the upper box part. See `/tcb/doc new`.

N 2014-09-19 `/tcb/doc new and updated={`⟨*new date*⟩`}{`⟨*update date*⟩`}` (style, no default)

Adds a marginnote with 'New: ⟨*new date*⟩' and 'Updated: ⟨*update data*⟩' messages at the beginning of the upper box part. See `/tcb/doc new`.

23.3 Predefined Colors of the Library

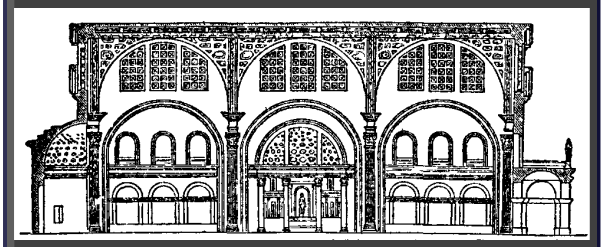
The following colors are predefined. They are used as default colors in some library commands.

`Option` , `Definition` , `ExampleFrame` , `ExampleBack` , `Hyperlink` .

A Picture Credits

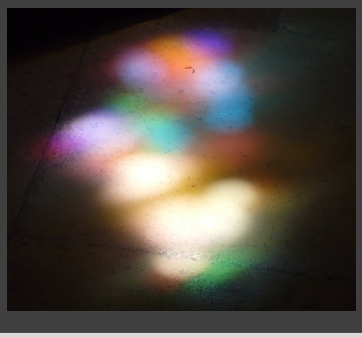
The following pictures were used inside this documentation.

Basilica_5.png



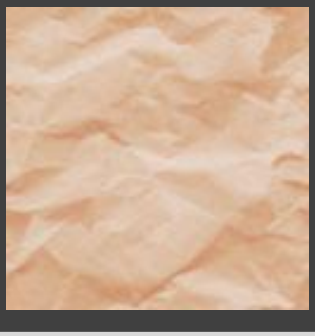
http://commons.wikimedia.org/wiki/File:Basilica_5.png

lichtspiel.jpg



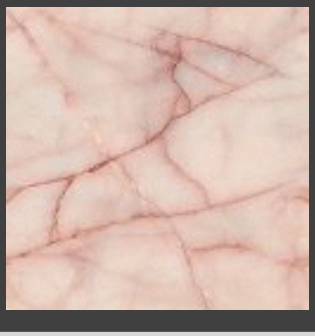
Photograph taken by Thomas F. Sturm.

crinklepaper.png



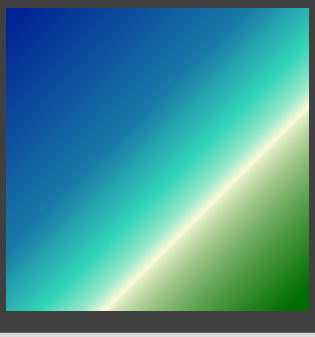
Created with GIMP.
http://www.gimp.org

pink_marble.png



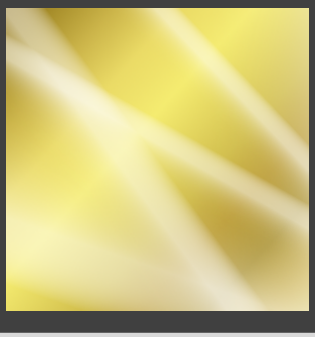
Created with GIMP.
http://www.gimp.org

blueshade.png



Created with GIMP.
http://www.gimp.org

goldshade.png



Created with GIMP.
http://www.gimp.org

References

[1] Donald Arseneau. *The varwidth package*. Sept. 26, 2011.
http://mirrors.ctan.org/macros/latex/contrib/varwidth/varwidth-doc.pdf.

[2] Peter Breitenlohner. *The ε-TEX manual*. Version 2. Feb. 1998.
http://mirrors.ctan.org/systems/doc/etex/etex_man.pdf.

[3] David Carlisle. *The file ltxdoc.dtx for use with LaTeX 2ε*. Nov. 11, 2007.
http://mirror.ctan.org/macros/latex/base/ltxdoc.dtx.

[4] David Carlisle. *The tabularx package*. Oct. 28, 2014.
http://mirror.ctan.org/macros/latex/required/tools/tabularx.pdf.

[5] Toby Cubitt. *The cleveref Package*. Dec. 28, 2013.
http://mirror.ctan.org/macros/latex/contrib/cleveref/cleveref.pdf.

[6] Carsten Heinz, Brooks Moses, and Jobst Hoffmann. *The Listings Package*. Version 1.6. June 4, 2015.
http://mirror.ctan.org/macros/latex/contrib/listings/listings.pdf.

[7] Philipp Lehman and Joseph Wright. *The etoolbox Package*. Aug. 2, 2015.
http://mirror.ctan.org/macros/latex/contrib/etoolbox/etoolbox.pdf.

[8] Frank Mittelbach. *The doc and shortvrb Packages*. June 30, 2011.
http://mirror.ctan.org/macros/latex/base/doc.dtx.

[9] Frank Mittelbach. *An environment for multicolumn output*. Sept. 30, 2015.
http://mirror.ctan.org/macros/latex/required/tools/multicol.pdf.

[10] Heiko Oberdiek. *The listingsutf8 package*. Nov. 10, 2011.
http://mirror.ctan.org/macros/latex/contrib/oberdiek/listingsutf8.pdf.

[11] Geoffrey M. Poore and Konrad Rudolph. *The minted package. Highlighted source code in LaTeX*. Sept. 9, 2015.
http://mirror.ctan.org/macros/latex/contrib/minted/minted.pdf.

[12] The LaTeX3 Project. *The xparse Package*. Oct. 14, 2015.
http://mirror.ctan.org/macros/latex/contrib/l3packages/xparse.pdf.

[13] *Pygments. Python syntax highlighter*. 2013.
http://pygments.org/.

[14] Will Robertson. *A couple of things involving environments*. May 4, 2014.
http://mirror.ctan.org/macros/latex/contrib/environ/environ.pdf.

[15] Rainer Schöpf, Bernd Raichle, and Chris Rowley. *A New Implementation of LaTeX's verbatim and verbatim* Environments*. Mar. 12, 2001.
http://mirror.ctan.org/macros/latex/required/tools/verbatim.pdf.

[16] Thomas F. Sturm. *Mathematik 1 (B.Eng.)* Vorlesungs-Skriptum. Neubiberg: Universität der Bundeswehr München, 2007.
http://www.unibw.de/bw/sturm.

[17] Thomas F. Sturm. *Mathematik 2 (B.Eng.)* Vorlesungs-Skriptum. Neubiberg: Universität der Bundeswehr München, 2008.
http://www.unibw.de/bw/sturm.

[18] Thomas F. Sturm. *Höhere Mathematik: Fortgeschrittene mathematische Methoden (M.Eng.)* Vorlesungs-Skriptum. Neubiberg: Universität der Bundeswehr München, 2010.
http://www.unibw.de/bw/sturm.

[19] Thomas F. Sturm. *LaTeX – Einführung in das Textsatzsystem*. 10th edition. RRZN-Handbücher. Hannover: Regionales Rechenzentrum für Niedersachsen (RRZN), Mar. 2014.
http://www.rrzn.uni-hannover.de/buch.html?&titel=latex.

[20] Till Tantau. *The TikZ and PGF Packages. Manual for version 3.0.1a*. Aug. 29, 2015.
http://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf.

Index

- key, 416
! key, 416

absquote environment, 432
add to height key, 54
add to list key, 95
add to natural height key, 54
add to width key, 34
adjust text key, 18
adjusted title key, 18
adjusted title after break key, 352
after key, 76
after app key, 393
after doc body key, 442
after doc body command key, 442
after doc body environment key, 442
after doc body key key, 442
after example key, 436
after lower key, 64
after lower app key, 393
after lower pre key, 393
after lower* key, 64
after pre key, 393
after skip key, 78
after title key, 62
after title app key, 392
after title pre key, 392
after upper key, 63
after upper app key, 392
after upper pre key, 392
after upper* key, 63
all key, 10
all value, 47, 48, 275, 354, 356
ams align key, 337
ams align lower key, 337
ams align upper key, 337
ams align* key, 337
ams align* lower key, 337
ams align* upper key, 337
ams equation key, 336
ams equation lower key, 336
ams equation upper key, 336
ams equation* key, 336
ams equation* lower key, 336
ams equation* upper key, 336
ams gather key, 338
ams gather lower key, 338
ams gather upper key, 338
ams gather* key, 338
ams gather* lower key, 338
ams gather* upper key, 338
ams nodisplayskip key, 339
ams nodisplayskip lower key, 339
ams nodisplayskip upper key, 339
arc key, 36
arc is angular key, 38
arc is curved key, 38
areasize value, 388
areasize* value, 388
at begin tikz key, 183
at begin tikz reset key, 183
at end tikz key, 183
at end tikz reset key, 183
attach boxed title to bottom key, 147
attach boxed title to bottom center key, 146
attach boxed title to bottom left key, 146
attach boxed title to bottom right key, 146
attach boxed title to bottom* key, 147
attach boxed title to top key, 147
attach boxed title to top center key, 146
attach boxed title to top left key, 146
attach boxed title to top right key, 146
attach boxed title to top* key, 147
attach title key, 20
attach title to upper key, 20
auto value, 93
auto counter key, 100
auto limited value, 93
auto outer arc key, 38
autoparskip key, 76

base value, 77
base color key, 257
baseline key, 77
baselineskip value, 354
beamer key, 217
beamer Skin, 217
beamerfirst Skin, 219
beamerlast Skin, 221
beamermiddle Skin, 220
bean arc key, 37
before key, 76
before app key, 393
before doc body key, 442
before doc body command key, 442
before doc body environment key, 442
before doc body key key, 442
before example key, 436
before lower key, 64
before lower app key, 393
before lower pre key, 393
before nobreak key, 79
before pre key, 393
before skip key, 78
before title key, 62
before title app key, 392
before title pre key, 392
before upper key, 63
before upper app key, 392
before upper pre key, 392

beforeafter skip key, 78
bicolor key, 208
bicolor Skin, 208
bicolorfirst Skin, 210
bicolorlast Skin, 212
bicolormiddle Skin, 211
blank key, 197
blanker key, 227
blankest key, 228
blend before title key, 105
blend before title code key, 106
blend into key, 104
borderline key, 168
borderline east key, 171
borderline horizontal key, 172
borderline north key, 171
borderline south key, 171
borderline vertical key, 172
borderline west key, 171
both value, 115
bottom key, 42
bottom value, 33, 77, 109, 274
bottom seam value, 109
bottomrule key, 35
bottomrule at break key, 355
bottomsep at break key, 355
bottomtitle key, 42
box align key, 77
\boxarrayclear, 373
\boxarraygetbox, 377
\boxarraygetdepth, 378
\boxarraygetheight, 378
\boxarraygetsize, 375
\boxarraygettotalheight, 379
\boxarraygetwidth, 378
\boxarrayreset, 372
boxarraystore environment, 375
boxed title size key, 149
boxed title style key, 150
boxrule key, 36
boxsep key, 38
\brackets, 430
break value, 341
break at key, 353
breakable key, 9, 351
broken value, 156–158

capture key, 90
center key, 82
center value, 30, 33, 77, 109, 274
center lower key, 32
center seam value, 109
center title key, 32
center upper key, 32
change value, 341
change apart value, 341
change break value, 341
change standard value, 340
check odd page key, 95
circular arc key, 37
clear preamble key, 423
clear preclass key, 423
clip lower key, 167
clip title key, 166
clip upper key, 166
clip watermark key, 162
clipped value, 258
code key, 99
colback key, 27
colbacklower key, 209
colbacktitle key, 27
\colDef, 434
colframe key, 27
collower key, 28
colon value, 105
colon hang value, 105
\colOpt, 434
color key, 439
color color key, 438
color command key, 438
color counter key, 438
color definition key, 438
color environment key, 438
color from key, 257
color hyperlink key, 438
color key key, 438
color length key, 438
color option key, 438
color value key, 438
Colors
    Definition, 443
    ExampleBack, 443
    ExampleFrame, 443
    foocolor, 429
    Hyperlink, 443
    Option, 443
colors key, 439
coltext key, 28
coltitle key, 28
colupper key, 28
comment key, 299
comment above listing key, 306
comment above* listing key, 306
comment and listing key, 302
comment only key, 299
comment outside listing key, 304
comment side listing key, 304
comment style key, 302
compilable listing key, 310
compiler key, 422
compress page key, 354
\consumeboxarray, 376
\consumetcboxarray, 376
copy value, 149
counter key, 439
Counters
    foocounter, 429
counters key, 439
Crefname key, 103

crefname key, 103
\cs, 430

dash value, 105
dash hang value, 105
\DeclareTCBInputListing, 412
\DeclareTCBListing, 410
\DeclareTCBox, 407
\DeclareTCBoxFit, 413
\DeclareTColorBox, 404
\DeclareTotalTCBox, 408
\DeclareTotalTCBoxFit, 414
\DeclareTotalTColorBox, 406
Definition color, 443
description color key, 331
description delimiters key, 331
description delimiters none key, 331
description delimiters parenthesis key, 331
description font key, 332
description formatter key, 332
detach title key, 20
direct value, 258
dispExample environment, 431
dispExample* environment, 431
dispListing environment, 432
dispListing* environment, 432
do not store to box array key, 375
doc value, 437
doc description key, 443
doc head key, 441
doc head command key, 441
doc head environment key, 441
doc head key key, 441
doc into index key, 443
doc left key, 440
doc left indent key, 440
doc marginnote key, 443
doc new key, 443
doc new and updated key, 443
doc right key, 440
doc right indent key, 440
doc updated key, 443
\docAuxCommand, 428
\docAuxCommand*, 428
\docAuxEnvironment, 428
\docAuxEnvironment*, 428
\docAuxKey, 429
\docAuxKey*, 429
\docColor, 429
\docColor*, 429
docCommand environment, 426
docCommand* environment, 427
\docCounter, 429
\docCounter*, 429
docEnvironment environment, 427
docEnvironment* environment, 427
docexample key, 436
docKey environment, 428
docKey* environment, 428
\docLength, 429
\docLength*, 429
documentation key, 10
documentation listing options key, 436
documentation listing style key, 436
documentation minted options key, 436
documentation minted style key, 436
\docValue, 428
\docValue*, 428
downhill value, 47, 48
draft key, 237
draft Skin, 237
draftmode key, 192
draw method key, 258
drop fuzzy midday shadow key, 174
drop fuzzy shadow key, 173
drop fuzzy shadow east key, 177
drop fuzzy shadow north key, 177
drop fuzzy shadow northeast key, 177
drop fuzzy shadow northwest key, 177
drop fuzzy shadow south key, 176
drop fuzzy shadow southeast key, 176
drop fuzzy shadow southwest key, 176
drop fuzzy shadow west key, 176
drop large lifted shadow key, 178
drop lifted shadow key, 178
drop midday shadow key, 173
drop shadow key, 173
drop shadow east key, 176
drop shadow north key, 175
drop shadow northeast key, 176
drop shadow northwest key, 175
drop shadow south key, 175
drop shadow southeast key, 175
drop shadow southwest key, 175
drop shadow west key, 175
drop small lifted shadow key, 178

east fading, 258
east value, 47, 48
east size key, 255
east style key, 256
empty key, 226
empty Skin, 226
empty value, 126, 127
emptyfirst Skin, 229
emptylast Skin, 231
emptymiddle Skin, 230
enforce breakable key, 352
english language key, 439
enhanced key, 195
enhanced Skin, 195
enhanced jigsaw key, 202
enhanced jigsaw Skin, 202
enhanced standard key, 197
enhancedfirst Skin, 199
enhancedfirst jigsaw Skin, 203
enhancedlast Skin, 201
enhancedlast jigsaw Skin, 207
enhancedmiddle Skin, 200

enhancedmiddle jigsaw Skin, 204
enlarge bottom at break by key, 81
enlarge bottom by key, 81
enlarge bottom finally by key, 80
enlarge by key, 82
enlarge left by key, 81
enlarge right by key, 81
enlarge top at break by key, 81
enlarge top by key, 81
enlarge top initially by key, 80
enlargepage key, 353
enlargepage flexible key, 354
environment key, 422, 439
environment content key, 439
environment with percent key, 422
Environments
    absquote, 432
    boxarraystore, 375
    dispExample, 431
    dispExample*, 431
    dispListing, 432
    dispListing*, 432
    docCommand, 426
    docCommand*, 427
    docEnvironment, 427
    docEnvironment*, 427
    docKey, 428
    docKey*, 428
    extcolorbox, 419
    extikzpicture, 420
    fooauxenv, 428
    foocolorbox, 427
    foocolorbox*, 427
    tcbclipframe, 163
    tcbclipinterior, 165
    tcbcliptitle, 165
    tcbexternal, 417
    tcbinvclipframe, 164
    tcbitemize, 268
    tcblisting, 287
    tcboutputlisting, 289
    tcboxeditemize, 270
    tcboxedraster, 269
    tcbraster, 267
    tcbverbatimwrite, 118
    tcbwritetemp, 118
    tcolorbox, 12
environments key, 439
equal height group key, 59
evenpage value, 45, 84
every box key, 87
every box on higher layers key, 88
every box on layer n key, 88
every float key, 75
every listing line key, 294
every listing line* key, 294
ExampleBack color, 443
ExampleFrame color, 443
extcolorbox environment, 419
extend freelance key, 239
extend freelancefirst key, 239
extend freelancelast key, 239
extend freelancemiddle key, 239
external key, 10, 98
externalize key, 416
externalize example key, 421
externalize example! key, 421
externalize listing key, 421
externalize listing! key, 421
extikzpicture environment, 420
extras key, 357
extras broken key, 357
extras broken pre key, 401
extras first key, 357
extras first and middle key, 357
extras first and middle pre key, 401
extras first pre key, 401
extras last key, 357
extras last pre key, 401
extras middle key, 357
extras middle and last key, 357
extras middle and last pre key, 401
extras middle pre key, 401
extras pre key, 401
extras unbroken key, 357
extras unbroken and first key, 357
extras unbroken and first pre key, 401
extras unbroken and last key, 357
extras unbroken and last pre key, 401
extras unbroken pre key, 401
extrude bottom by key, 86
extrude by key, 86
extrude left by key, 85
extrude right by key, 85
extrude top by key, 86

fade in key, 259
fade out key, 259
Fadings
    east, 258
    north, 258
    semi east, 258
    semi north, 258
    semi south, 258
    semi west, 258
    south, 258
    west, 258
false value, 76, 79, 276, 351
fbox value, 43
figures value, 104
fill image opacity key, 250
fill image options key, 250
fill image scale key, 250
fill overzoom image key, 246
fill overzoom image* key, 246
fill overzoom picture key, 246
fill plain image key, 244
fill plain image* key, 244
fill plain picture key, 244

fill shrink image key, 248
fill shrink image* key, 248
fill shrink picture key, 248
fill stretch image key, 245
fill stretch image* key, 245
fill stretch picture key, 245
fill tile image key, 249
fill tile image* key, 249
fill tile picture key, 249
fill tile picture* key, 249
fill zoom image key, 247
fill zoom image* key, 247
fill zoom picture key, 247
final value, 390
finish key, 188
finish broken key, 189
finish broken pre key, 399
finish fading vignette key, 263
finish first key, 189
finish first and middle key, 189
finish first and middle pre key, 399
finish first pre key, 399
finish last key, 189
finish last pre key, 399
finish middle key, 189
finish middle and last key, 189
finish middle and last pre key, 399
finish middle pre key, 399
finish pre key, 399
finish raised fading vignette key, 262
finish unbroken key, 189
finish unbroken and first key, 189
finish unbroken and first pre key, 399
finish unbroken and last key, 189
finish unbroken and last pre key, 399
finish unbroken pre key, 399
finish vignette key, 262
first value, 156–158, 356
first and middle value, 156, 356
fit key, 382
fit algorithm key, 388
fit basedim key, 383
fit fontsize macros key, 384
fit height from key, 387
fit height plus key, 385
fit maxfontdiff key, 390
fit maxfontdiffgap key, 390
fit maxstep key, 390
fit maxwidthdiff key, 390
fit maxwidthdiffgap key, 390
fit skip key, 383
fit to key, 383
fit to height key, 383
fit warning key, 390
fit width from key, 386
fit width plus key, 385
fitbox value, 90
fitting key, 9
flip title key, 147
float key, 74
float* key, 74
floatplacement key, 74
flush center value, 30, 32
flush left key, 82
flush left value, 30, 32
flush right key, 82
flush right value, 30, 32
flushleft lower key, 32
flushleft title key, 32
flushleft upper key, 32
flushright lower key, 32
flushright title key, 32
flushright upper key, 32
fontlower key, 29
fontsize value, 388
fontsize* value, 388
fonttitle key, 29
fontupper key, 29
\fooaux, 428
fooaux key, 429
fooauxenv environment, 428
foocolor color, 429
foocolorbox environment, 427
foocolorbox* environment, 427
foocounter counter, 429
foodummy key, 433
\foolength length, 429
\foomakedocSubKey, 426
\foomakedocSubKey*, 427
\foosomething, 443
footitle key, 428
foovalue value, 428
force remake key, 416
forced value, 45, 84
forced center value, 93
forced left value, 93
forced right value, 93
forces nobeforeafter key, 76
frame code key, 129
frame code app key, 399
frame code pre key, 399
frame empty key, 129
frame engine key, 126
frame hidden key, 140
frame style key, 139
frame style image key, 139
frame style tile key, 140
freelance key, 239
freelance Skin, 239
freelance value, 126, 127
freelancefirst Skin, 239
freelancelast Skin, 239
freelancemiddle Skin, 239
fuzzy halo key, 174
fuzzy shadow key, 180

geometry nodes key, 128
graphical environment key, 125
graphics directory key, 243

graphics options key, 243
graphics pages key, 243
grow to left by key, 83
grow to right by key, 83

halign key, 30
halign lower key, 31
halign title key, 32
halign upper key, 30
halo key, 174
hbox key, 90
hbox value, 90
hbox boxed title key, 154
height key, 52
height fill key, 55
height fixed for key, 356
height from key, 53
height plus key, 52
highlight math key, 335
highlight math style key, 335
hooks key, 9
horizontal size key, 256
hybrid value, 388
hybrid* value, 388
Hyperlink color, 443
hyphenationfix key, 92

if odd page key, 95
if odd page* key, 96
IfBooleanTF key, 403
IfNoValueTF key, 402
IfValueTF key, 403
ignore nobreak key, 79
ignored value, 24
image comment key, 299
\imagename, 241
\imagepage, 242
index key, 439
index actual key, 437
index annotate key, 438
index colorize key, 438
index command key, 437
index command name key, 437
index default settings key, 437
index format key, 437
index german settings key, 437
index level key, 437
index quote key, 437
input source on error key, 422
inside node key, 254
interior code key, 130
interior code app key, 400
interior code pre key, 400
interior empty key, 130
interior engine key, 127
interior hidden key, 141
interior style key, 140
interior style image key, 141
interior style tile key, 141
interior titled code key, 129
interior titled code app key, 399
interior titled code pre key, 400
interior titled empty key, 129
interior titled engine key, 126
invisible key, 22
invisible value, 22, 24

justify value, 30

key key, 439
Keys
  /foo/
    fooaux, 429
    foodummy, 433
    footitle, 428
  /tcb/
    add to height, 54
    add to list, 95
    add to natural height, 54
    add to width, 34
    adjust text, 18
    adjusted title, 18
    adjusted title after break, 352
    after, 76
    after app, 393
    after doc body, 442
    after doc body command, 442
    after doc body environment, 442
    after doc body key, 442
    after example, 436
    after lower, 64
    after lower app, 393
    after lower pre, 393
    after lower*, 64
    after pre, 393
    after skip, 78
    after title, 62
    after title app, 392
    after title pre, 392
    after upper, 63
    after upper app, 392
    after upper pre, 392
    after upper*, 63
    ams align, 337
    ams align lower, 337
    ams align upper, 337
    ams align*, 337
    ams align* lower, 337
    ams align* upper, 337
    ams equation, 336
    ams equation lower, 336
    ams equation upper, 336
    ams equation*, 336
    ams equation* lower, 336
    ams equation* upper, 336
    ams gather, 338
    ams gather lower, 338
    ams gather upper, 338
    ams gather*, 338
    ams gather* lower, 338

ams gather* upper, 338
ams nodisplayskip, 339
ams nodisplayskip lower, 339
ams nodisplayskip upper, 339
arc, 36
arc is angular, 38
arc is curved, 38
at begin tikz, 183
at begin tikz reset, 183
at end tikz, 183
at end tikz reset, 183
attach boxed title to bottom, 147
attach boxed title to bottom center, 146
attach boxed title to bottom left, 146
attach boxed title to bottom right, 146
attach boxed title to bottom*, 147
attach boxed title to top, 147
attach boxed title to top center, 146
attach boxed title to top left, 146
attach boxed title to top right, 146
attach boxed title to top*, 147
attach title, 20
attach title to upper, 20
auto outer arc, 38
autoparskip, 76
baseline, 77
beamer, 217
bean arc, 37
before, 76
before app, 393
before doc body, 442
before doc body command, 442
before doc body environment, 442
before doc body key, 442
before example, 436
before lower, 64
before lower app, 393
before lower pre, 393
before nobreak, 79
before pre, 393
before skip, 78
before title, 62
before title app, 392
before title pre, 392
before upper, 63
before upper app, 392
before upper pre, 392
beforeafter skip, 78
bicolor, 208
blank, 197
blanker, 227
blankest, 228
blend before title, 105
blend before title code, 106
borderline, 168
borderline east, 171
borderline horizontal, 172
borderline north, 171
borderline south, 171
borderline vertical, 172
borderline west, 171
bottom, 42
bottomrule, 35
bottomrule at break, 355
bottomsep at break, 355
bottomtitle, 42
box align, 77
boxed title size, 149
boxed title style, 150
boxrule, 36
boxsep, 38
break at, 353
breakable, 351
capture, 90
center, 82
center lower, 32
center title, 32
center upper, 32
check odd page, 95
circular arc, 37
clip lower, 167
clip title, 166
clip upper, 166
clip watermark, 162
code, 99
colback, 27
colbacklower, 209
colbacktitle, 27
colframe, 27
collower, 28
color color, 438
color command, 438
color counter, 438
color definition, 438
color environment, 438
color hyperlink, 438
color key, 438
color length, 438
color option, 438
color value, 438
coltext, 28
coltitle, 28
colupper, 28
comment, 299
comment above listing, 306
comment above* listing, 306
comment and listing, 302
comment only, 299
comment outside listing, 304
comment side listing, 304
comment style, 302
compilable listing, 310
compress page, 354

description color, 331
description delimiters, 331
description delimiters none, 331
description delimiters parenthesis, 331
description font, 332
description formatter, 332
detach title, 20
do not store to box array, 375
doc description, 443
doc head, 441
doc head command, 441
doc head environment, 441
doc head key, 441
doc into index, 443
doc left, 440
doc left indent, 440
doc marginnote, 443
doc new, 443
doc new and updated, 443
doc right, 440
doc right indent, 440
doc updated, 443
docexample, 436
documentation listing options, 436
documentation listing style, 436
documentation minted options, 436
documentation minted style, 436
draft, 237
draftmode, 192
drop fuzzy midday shadow, 174
drop fuzzy shadow, 173
drop fuzzy shadow east, 177
drop fuzzy shadow north, 177
drop fuzzy shadow northeast, 177
drop fuzzy shadow northwest, 177
drop fuzzy shadow south, 176
drop fuzzy shadow southeast, 176
drop fuzzy shadow southwest, 176
drop fuzzy shadow west, 176
drop large lifted shadow, 178
drop lifted shadow, 178
drop midday shadow, 173
drop shadow, 173
drop shadow east, 176
drop shadow north, 175
drop shadow northeast, 176
drop shadow northwest, 175
drop shadow south, 175
drop shadow southeast, 175
drop shadow southwest, 175
drop shadow west, 175
drop small lifted shadow, 178
empty, 226
enforce breakable, 352
english language, 439
enhanced, 195
enhanced jigsaw, 202
enhanced standard, 197
enlarge bottom at break by, 81
enlarge bottom by, 81
enlarge bottom finally by, 80
enlarge by, 82
enlarge left by, 81
enlarge right by, 81
enlarge top at break by, 81
enlarge top by, 81
enlarge top initially by, 80
enlargepage, 353
enlargepage flexible, 354
equal height group, 59
every box, 87
every box on higher layers, 88
every box on layer n, 88
every float, 75
every listing line, 294
every listing line*, 294
extend freelance, 239
extend freelancefirst, 239
extend freelancelast, 239
extend freelancemiddle, 239
external, 98
externalize example, 421
externalize example!, 421
externalize listing, 421
externalize listing!, 421
extras, 357
extras broken, 357
extras broken pre, 401
extras first, 357
extras first and middle, 357
extras first and middle pre, 401
extras first pre, 401
extras last, 357
extras last pre, 401
extras middle, 357
extras middle and last, 357
extras middle and last pre, 401
extras middle pre, 401
extras pre, 401
extras unbroken, 357
extras unbroken and first, 357
extras unbroken and first pre, 401
extras unbroken and last, 357
extras unbroken and last pre, 401
extras unbroken pre, 401
extrude bottom by, 86
extrude by, 86
extrude left by, 85
extrude right by, 85
extrude top by, 86
finish, 188
finish broken, 189
finish broken pre, 399
finish fading vignette, 263
finish first, 189
finish first and middle, 189
finish first and middle pre, 399

finish first pre, 399
finish last, 189
finish last pre, 399
finish middle, 189
finish middle and last, 189
finish middle and last pre, 399
finish middle pre, 399
finish pre, 399
finish raised fading vignette, 262
finish unbroken, 189
finish unbroken and first, 189
finish unbroken and first pre, 399
finish unbroken and last, 189
finish unbroken and last pre, 399
finish unbroken pre, 399
finish vignette, 262
fit, 382
fit algorithm, 388
fit basedim, 383
fit fontsize macros, 384
fit height from, 387
fit height plus, 385
fit maxfontdiff, 390
fit maxfontdiffgap, 390
fit maxstep, 390
fit maxwidthdiff, 390
fit maxwidthdiffgap, 390
fit skip, 383
fit to, 383
fit to height, 383
fit warning, 390
fit width from, 386
fit width plus, 385
flip title, 147
float, 74
float*, 74
floatplacement, 74
flush left, 82
flush right, 82
flushleft lower, 32
flushleft title, 32
flushleft upper, 32
flushright lower, 32
flushright title, 32
flushright upper, 32
fontlower, 29
fonttitle, 29
fontupper, 29
forces nobeforeafter, 76
frame code, 129
frame code app, 399
frame code pre, 399
frame empty, 129
frame engine, 126
frame hidden, 140
frame style, 139
frame style image, 139
frame style tile, 140
freelance, 239
fuzzy halo, 174
fuzzy shadow, 180
geometry nodes, 128
graphical environment, 125
graphics directory, 243
graphics options, 243
graphics pages, 243
grow to left by, 83
grow to right by, 83
halign, 30
halign lower, 31
halign title, 32
halign upper, 30
halo, 174
hbox, 90
hbox boxed title, 154
height, 52
height fill, 55
height fixed for, 356
height from, 53
height plus, 52
highlight math, 335
highlight math style, 335
hyphenationfix, 92
if odd page, 95
if odd page*, 96
IfBooleanTF, 403
IfNoValueTF, 402
IfValueTF, 403
ignore nobreak, 79
image comment, 299
index actual, 437
index annotate, 438
index colorize, 438
index command, 437
index command name, 437
index default settings, 437
index format, 437
index german settings, 437
index level, 437
index quote, 437
interior code, 130
interior code app, 400
interior code pre, 400
interior empty, 130
interior engine, 127
interior hidden, 141
interior style, 140
interior style image, 141
interior style tile, 141
interior titled code, 129
interior titled code app, 399
interior titled code pre, 400
interior titled empty, 129
interior titled engine, 126
invisible, 22
label, 94
label separator, 333
label type, 94

left, 39
left skip, 79
lefthand ratio, 112
lefthand width, 111
leftlower, 39
leftright skip, 79
leftrule, 35
lefttitle, 39
leftupper, 39
lifted shadow, 181
lines before break, 352
list entry, 95
list text, 95
listing above comment, 306
listing above text, 305
listing above* comment, 306
listing above* text, 305
listing and comment, 302
listing and text, 298
listing engine, 298
listing file, 298
listing inputencoding, 294
listing only, 298
listing options, 293
listing outside comment, 304
listing outside text, 303
listing remove caption, 294
listing side comment, 304
listing side text, 303
listing style, 293
listing utf8, 295
lower separated, 25
lowerbox, 24
marker, 204
math, 336
math lower, 336
math upper, 336
middle, 42
minimum for current equal height group, 60
minimum for equal height group, 60
minipage, 90
minipage boxed title, 154
minipage boxed title*, 154
minted language, 296
minted options, 296
minted style, 297
natural height, 52
no borderline, 170
no boxed title style, 153
no extras, 357
no extras first, 357
no extras last, 357
no extras middle, 357
no extras unbroken, 357
no finish, 189
no finish first, 189
no finish last, 189
no finish middle, 189
no finish unbroken, 189
no label type, 94
no listing options, 293
no overlay, 70
no process, 310
no recording, 119
no shadow, 173
no underlay, 186
no underlay boxed title, 187
no underlay first, 187
no underlay last, 187
no underlay middle, 187
no underlay unbroken, 187
no watermark, 158
nobeforeafter, 76
nofloat, 74
noparskip, 76
nophantom, 94
notitle, 18
notitle after break, 352
octogon arc, 37
on line, 92
only, 99
opacityback, 50
opacitybacktitle, 50
opacityfill, 50
opacityframe, 50
opacitylower, 51
opacitytext, 51
opacitytitle, 51
opacityupper, 51
outer arc, 38
overlay, 69
overlay app, 394
overlay broken, 70
overlay broken app, 395
overlay broken pre, 395
overlay first, 70
overlay first and middle, 70
overlay first and middle app, 395
overlay first and middle pre, 395
overlay first app, 394
overlay first pre, 394
overlay last, 70
overlay last app, 395
overlay last pre, 395
overlay middle, 70
overlay middle and last, 70
overlay middle and last app, 395
overlay middle and last pre, 395
overlay middle app, 395
overlay middle pre, 395
overlay pre, 394
overlay unbroken, 70
overlay unbroken and first, 70
overlay unbroken and first app, 395
overlay unbroken and first pre, 395
overlay unbroken and last, 70
overlay unbroken and last app, 395

overlay unbroken and last pre, 395
overlay unbroken app, 394
overlay unbroken pre, 394
oversize, 44
pad after break, 355
pad at break, 355
pad at break*, 355
pad before break, 355
pad before break*, 355
parbox, 91
parskip, 76
pdf comment, 300
pdf extension, 302
phantom, 94
phantomlabel, 94
process code, 310
raster after skip, 272
raster before skip, 272
raster column n, 276
raster column skip, 273
raster columns, 271
raster equal height, 275
raster equal height group, 275
raster equal skip, 272
raster even column, 276
raster even number, 277
raster even row, 277
raster every box, 276
raster force size, 276
raster halign, 274
raster height, 272
raster left skip, 273
raster multicolumn, 278
raster multirow, 279
raster number n, 277
raster odd column, 276
raster odd number, 277
raster odd row, 276
raster reset, 276
raster right skip, 273
raster row m, 277
raster row m column n, 277
raster row skip, 273
raster rows, 271
raster valign, 274
raster width, 271
record, 119
remake, 98
remember, 184
remember as, 185
reset, 98
reset and store to box array, 375
reset box array, 372
right, 40
right skip, 79
righthand ratio, 112
righthand width, 111
rightlower, 41
rightrule, 35
righttitle, 40
rightupper, 40
rotate, 184
rounded corners, 48
run arara, 312
run biber, 312
run bibtex, 312
run dvips, 312
run latex, 312
run lualatex, 312
run makeindex, 312
run pdflatex, 310
run ps2pdf, 312
run system command, 310
run xelatex, 312
savedelimiter, 26
savelowerto, 24
saveto, 23
scale, 184
segmentation code, 130
segmentation code app, 400
segmentation code pre, 400
segmentation empty, 130
segmentation engine, 127
segmentation hidden, 142
segmentation style, 142
separator sign, 330
separator sign colon, 330
separator sign dash, 330
separator sign none, 330
shadow, 179
sharp corners, 47
sharpish corners, 48
shield externalize, 98
show bounding box, 170
shrink break goal, 354
shrink tight, 85
sidebyside, 108
sidebyside adapt, 115
sidebyside align, 109
sidebyside gap, 111
sidebyside switch, 117
size, 43
skin, 125
skin first, 125
skin first is subskin of, 132
skin last, 125
skin last is subskin of, 132
skin middle, 125
skin middle is subskin of, 132
smart shadow arc, 181
space, 56
space to, 57
space to both, 57
space to lower, 56
space to upper, 56
spartan, 236
split, 58
square, 56

squeezed title, 19
squeezed title*, 19
standard, 193
standard jigsaw, 194
step, 94
step and label, 94
store to box array, 373
subtitle style, 21
tabularx, 66
tabularx*, 66
tcbimage comment, 300
tcbox raise, 92
tcbox raise base, 92
tcbox width, 93
tempfile, 92
terminator sign, 332
terminator sign colon, 333
terminator sign dash, 333
terminator sign none, 333
text above listing, 305
text above* listing, 305
text and listing, 298
text fill, 65
text height, 53
text only, 299
text outside listing, 303
text side listing, 303
text width, 34
theorem, 335
theorem name, 334
theorem name and number, 334
theorem number and name, 334
theorem style, 340
tikz, 183
tikz lower, 67
tikz reset, 183
tikz upper, 67
tikznode, 68
tikznode boxed title, 155
tikznode lower, 68
tikznode upper, 68
tile, 213
title, 18
title after break, 352
title code, 131
title code app, 400
title code pre, 400
title empty, 131
title engine, 127
title filled, 27
title hidden, 143
title style, 142
title style image, 143
title style tile, 143
titlerule, 36
titlerule style, 144
toggle enlargement, 84
toggle left and right, 45
top, 41
toprule, 35
toprule at break, 355
topsep at break, 355
toptitle, 41
unbreakable, 352
underlay, 186
underlay boxed title, 187
underlay boxed title pre, 398
underlay broken, 187
underlay broken pre, 398
underlay first, 187
underlay first and middle, 187
underlay first and middle pre, 398
underlay first pre, 398
underlay last, 187
underlay last pre, 398
underlay middle, 187
underlay middle and last, 187
underlay middle and last pre, 398
underlay middle pre, 398
underlay pre, 398
underlay raised fading vignette, 261
underlay raised shading vignette, 261
underlay shade in vignette, 261
underlay unbroken, 187
underlay unbroken and first, 187
underlay unbroken and first pre, 398
underlay unbroken and last, 187
underlay unbroken and last pre, 398
underlay unbroken pre, 398
underlay vignette, 260
upperbox, 22
use height from group, 61
valign, 33
valign lower, 33
valign scale limit, 33
valign upper, 33
varwidth boxed title, 155
varwidth boxed title*, 155
varwidth upper, 68
verbatim, 402
vfill before first, 356
visible, 22
watermark color, 161
watermark graphics, 157
watermark graphics app, 397
watermark graphics app on, 397
watermark graphics on, 157
watermark graphics pre, 397
watermark graphics pre on, 397
watermark opacity, 159
watermark overzoom, 160
watermark shrink, 160
watermark stretch, 161
watermark text, 156
watermark text app, 396

watermark text app on, 396
watermark text on, 156
watermark text pre, 396
watermark text pre on, 396
watermark tikz, 158
watermark tikz app, 397
watermark tikz app on, 397
watermark tikz on, 158
watermark tikz pre, 397
watermark tikz pre on, 397
watermark zoom, 159
widget, 222
width, 34
/tcb/boxtitle/
xshift, 148
yshift, 148
yshift*, 148
yshifttext, 148
/tcb/doclang/
color, 439
colors, 439
counter, 439
counters, 439
environment, 439
environment content, 439
environments, 439
index, 439
key, 439
keys, 439
length, 439
lengths, 439
new, 439
pageshort, 439
updated, 439
value, 439
values, 439
/tcb/external/
-, 416
!, 416
clear preamble, 423
clear preclass, 423
compiler, 422
environment, 422
environment with percent, 422
externalize, 416
force remake, 416
input source on error, 422
minipage, 422
name, 418
PassOptionsToClass, 423
PassOptionsToPackage, 423
plain, 422
preamble, 423
preamble tcbset, 423
preclass, 423
prefix, 416
runner, 416
runs, 422
safety, 422
/tcb/library/
all, 10
breakable, 9
documentation, 10
external, 10
fitting, 9
hooks, 9
listings, 9
listingsutf8, 9
magazine, 9
many, 10
minted, 9
most, 10
raster, 9
skins, 9
theorems, 9
vignette, 9
xparse, 10
/tcb/new/
auto counter, 100
blend into, 104
Crefname, 103
crefname, 103
list inside, 107
list type, 107
no counter, 101
number format, 102
number freestyle, 102
number within, 102
use counter, 101
use counter from, 101
use counter*, 101
/tcb/vig/
base color, 257
color from, 257
draw method, 258
east size, 255
east style, 256
fade in, 259
fade out, 259
horizontal size, 256
inside node, 254
lower left corner, 254
lowered color, 257
north size, 255
north style, 256
outside node, 255
over node, 255
over node offset, 255
raised color, 257
semi fade in, 259
semi fade out, 259
size, 256
south size, 255
south style, 256
upper right corner, 254
vertical size, 256
west size, 255
west style, 257

xmax, 254
xmin, 254
ymax, 254
ymin, 254
/tikz/
fill image opacity, 250
fill image options, 250
fill image scale, 250
fill overzoom image, 246
fill overzoom image*, 246
fill overzoom picture, 246
fill plain image, 244
fill plain image*, 244
fill plain picture, 244
fill shrink image, 248
fill shrink image*, 248
fill shrink picture, 248
fill stretch image, 245
fill stretch image*, 245
fill stretch picture, 245
fill tile image, 249
fill tile image*, 249
fill tile picture, 249
fill tile picture*, 249
fill zoom image, 247
fill zoom image*, 247
fill zoom picture, 247
tcb fill frame, 145
tcb fill interior, 145
tcb fill title, 145
keys key, 439

label key, 94
label separator key, 333
label type key, 94
last value, 156–158, 356
left key, 39
left value, 30, 115, 274
left skip key, 79
lefthand ratio key, 112
lefthand width key, 111
leftlower key, 39
leftright skip key, 79
leftrule key, 35
lefttitle key, 39
leftupper key, 39
length key, 439
Lengths
\foolength, 429
lengths key, 439
lifted shadow key, 181
lines before break key, 352
list entry key, 95
list inside key, 107
list text key, 95
list type key, 107
listing above comment key, 306
listing above text key, 305
listing above* comment key, 306
listing above* text key, 305
listing and comment key, 302
listing and text key, 298
listing engine key, 298
listing file key, 298
listing inputencoding key, 294
listing only key, 298
listing options key, 293
listing outside comment key, 304
listing outside text key, 303
listing remove caption key, 294
listing side comment key, 304
listing side text key, 303
listing style key, 293
listing utf8 key, 295
listings key, 9
listings value, 104, 298
listingsutf8 key, 9
lower left corner key, 254
lower separated key, 25
lowerbox key, 24
lowered color key, 257

magazine key, 9
many key, 10
\marg, 430
margin value, 342
margin apart value, 342
margin break value, 342
marker key, 204
math key, 336
math lower key, 336
math upper key, 336
maximum value, 55
\meta, 430
middle key, 42
middle value, 156–158, 356
middle and last value, 156–158, 356
minimal value, 43
minimum center value, 93
minimum for current equal height group key, 60
minimum for equal height group key, 60
minimum left value, 93
minimum right value, 93
minipage key, 90, 422
minipage value, 90, 268
minipage boxed title key, 154
minipage boxed title* key, 154
minted key, 9
minted value, 298
minted language key, 296
minted options key, 296
minted style key, 297
most key, 10

name key, 418
natural height key, 52
new key, 439
\newboxarray, 372
\newtcbexternalizeenvironment, 424

\newtcbexternalizetcolorbox, 424
\NewTCBInputListing, 412
\newtcbinputlisting, 292
\NewTCBListing, 410
\newtcblisting, 290
\NewTCBox, 407
\newtcbox, 16
\NewTCBoxFit, 413
\newtcboxfit, 381
\newtcbtheorem, 326
\NewTColorBox, 405
\newtcolorbox, 15
\NewTotalTCBox, 409
\NewTotalTCBoxFit, 414
\NewTotalTColorBox, 406
no borderline key, 170
no boxed title style key, 153
no counter key, 101
no extras key, 357
no extras first key, 357
no extras last key, 357
no extras middle key, 357
no extras unbroken key, 357
no finish key, 189
no finish first key, 189
no finish last key, 189
no finish middle key, 189
no finish unbroken key, 189
no label type key, 94
no listing options key, 293
no overlay key, 70
no process key, 310
no recording key, 119
no shadow key, 173
no underlay key, 186
no underlay boxed title key, 187
no underlay first key, 187
no underlay last key, 187
no underlay middle key, 187
no underlay unbroken key, 187
no watermark key, 158
nobeforeafter key, 76
nofloat key, 74
none value, 45, 84, 115, 275, 354, 356
noparskip key, 76
nophantom key, 94
normal value, 43, 149
north fading, 258
north value, 47, 48
north size key, 255
north style key, 256
northeast value, 47, 48
northwest value, 47, 48
notitle key, 18
notitle after break key, 352
number format key, 102
number freestyle key, 102
number within key, 102

\oarg, 430
octogon arc key, 37
off value, 390, 437
on value, 390
on line key, 92
only key, 99
opacityback key, 50
opacitybacktitle key, 50
opacityfill key, 50
opacityframe key, 50
opacitylower key, 51
opacitytext key, 51
opacitytitle key, 51
opacityupper key, 51
Option color, 443
outer arc key, 38
outside node key, 255
over node key, 255
over node offset key, 255
overlay key, 69
overlay app key, 394
overlay broken key, 70
overlay broken app key, 395
overlay broken pre key, 395
overlay first key, 70
overlay first and middle key, 70
overlay first and middle app key, 395
overlay first and middle pre key, 395
overlay first app key, 394
overlay first pre key, 394
overlay last key, 70
overlay last app key, 395
overlay last pre key, 395
overlay middle key, 70
overlay middle and last key, 70
overlay middle and last app key, 395
overlay middle and last pre key, 395
overlay middle app key, 395
overlay middle pre key, 395
overlay pre key, 394
overlay unbroken key, 70
overlay unbroken and first key, 70
overlay unbroken and first app key, 395
overlay unbroken and first pre key, 395
overlay unbroken and last key, 70
overlay unbroken and last app key, 395
overlay unbroken and last pre key, 395
overlay unbroken app key, 394
overlay unbroken pre key, 394
oversize key, 44

pad after break key, 355
pad at break key, 355
pad at break* key, 355
pad before break key, 355
pad before break* key, 355
pageshort key, 439
parbox key, 91
parskip key, 76
PassOptionsToClass key, 423
PassOptionsToPackage key, 423

path value, 126, 127
pathfirst value, 126, 127
pathfirstjigsaw value, 126
pathjigsaw value, 126
pathlast value, 126, 127
pathlastjigsaw value, 126
pathmiddle value, 126, 127
pathmiddlejigsaw value, 126
pdf comment key, 300
pdf extension key, 302
\pdfpages, 243
pgf value, 437
pgfchapter value, 437
pgfsection value, 437
phantom key, 94
phantomlabel key, 94
plain key, 422
plain value, 340
plain apart value, 341
preamble key, 423
preamble tcbset key, 423
preclass key, 423
prefix key, 416
process code key, 310
\ProvideTCBInputListing, 412
\ProvideTCBListing, 410
\ProvideTCBox, 407
\ProvideTCBoxFit, 413
\ProvideTColorBox, 405
\ProvideTotalTCBox, 409
\ProvideTotalTCBoxFit, 414
\ProvideTotalTColorBox, 406

raised color key, 257
raster key, 9
raster after skip key, 272
raster before skip key, 272
raster column n key, 276
raster column skip key, 273
raster columns key, 271
raster equal height key, 275
raster equal height group key, 275
raster equal skip key, 272
raster even column key, 276
raster even number key, 277
raster even row key, 277
raster every box key, 276
raster force size key, 276
raster halign key, 274
raster height key, 272
raster left skip key, 273
raster multicolumn key, 278
raster multirow key, 279
raster number n key, 277
raster odd column key, 276
raster odd number key, 277
raster odd row key, 276
raster reset key, 276
raster right skip key, 273
raster row m key, 277
raster row m column n key, 277
raster row skip key, 273
raster rows key, 271
raster valign key, 274
raster width key, 271
record key, 119
\refAux, 434
\refAuxcs, 434
\refCom, 433
\refCom*, 433
\refEnv, 433
\refEnv*, 433
\refKey, 434
\refKey*, 434
remake key, 98
remember key, 184
remember as key, 185
\renewtcbexternalizeenvironment, 424
\renewtcbexternalizetcolorbox, 425
\RenewTCBInputListing, 412
\renewtcbinputlisting, 292
\RenewTCBListing, 410
\renewtcblisting, 291
\RenewTCBox, 407
\renewtcbox, 16
\RenewTCBoxFit, 413
\renewtcboxfit, 381
\renewtcbtheorem, 327
\RenewTColorBox, 405
\renewtcolorbox, 15
\RenewTotalTCBox, 409
\RenewTotalTCBoxFit, 414
\RenewTotalTColorBox, 406
reset key, 98
reset and store to box array key, 375
reset box array key, 372
right key, 40
right value, 30, 115, 274
right skip key, 79
righthand ratio key, 112
righthand width key, 111
rightlower key, 41
rightrule key, 35
righttitle key, 40
rightupper key, 40
rotate key, 184
rounded corners key, 48
rows value, 275
run arara key, 312
run biber key, 312
run bibtex key, 312
run dvips key, 312
run latex key, 312
run lualatex key, 312
run makeindex key, 312
run pdflatex key, 310
run ps2pdf key, 312
run system command key, 310
run xelatex key, 312

runner key, 416
runs key, 422

safety key, 422
savedelimiter key, 26
savelowerto key, 24
saveto key, 23
scale key, 184
scale value, 33
scale* value, 33
segmentation code key, 130
segmentation code app key, 400
segmentation code pre key, 400
segmentation empty key, 130
segmentation engine key, 127
segmentation hidden key, 142
segmentation style key, 142
semi east fading, 258
semi fade in key, 259
semi fade out key, 259
semi north fading, 258
semi south fading, 258
semi west fading, 258
separator sign key, 330
separator sign colon key, 330
separator sign dash key, 330
separator sign none key, 330
shadow key, 179
sharp corners key, 47
sharpish corners key, 48
shield externalize key, 98
show bounding box key, 170
shrink break goal key, 354
shrink tight key, 85
sidebyside key, 108
sidebyside adapt key, 115
sidebyside align key, 109
sidebyside gap key, 111
sidebyside switch key, 117
size key, 43, 256
skin key, 125
skin first key, 125
skin first is subskin of key, 132
skin last key, 125
skin last is subskin of key, 132
skin middle key, 125
skin middle is subskin of key, 132
Skins
  beamer, 217
  beamerfirst, 219
  beamerlast, 221
  beamermiddle, 220
  bicolor, 208
  bicolorfirst, 210
  bicolorlast, 212
  bicolormiddle, 211
  draft, 237
  empty, 226
  emptyfirst, 229
  emptylast, 231
  emptymiddle, 230
  enhanced, 195
  enhanced jigsaw, 202
  enhancedfirst, 199
  enhancedfirst jigsaw, 203
  enhancedlast, 201
  enhancedlast jigsaw, 207
  enhancedmiddle, 200
  enhancedmiddle jigsaw, 204
  freelance, 239
  freelancefirst, 239
  freelancelast, 239
  freelancemiddle, 239
  spartan, 236
  standard, 193
  standard jigsaw, 194
  tile, 213
  tilefirst, 214
  tilelast, 216
  tilemiddle, 215
  widget, 222
  widgetfirst, 223
  widgetlast, 225
  widgetmiddle, 224
skins key, 9
small value, 43
smart shadow arc key, 181
south fading, 258
south value, 47, 48
south size key, 255
south style key, 256
southeast value, 47, 48
southwest value, 47, 48
space key, 56
space to key, 57
space to both key, 57
space to lower key, 56
space to upper key, 56
spartan key, 236
spartan Skin, 236
spartan value, 126, 127
split key, 58
square key, 56
squeeze value, 388
squeezed title key, 19
squeezed title* key, 19
standard key, 193
standard Skin, 193
standard value, 126, 127, 149, 340
standard jigsaw key, 194
standard jigsaw Skin, 194
step key, 94
step and label key, 94
store to box array key, 373
subtitle style key, 21

tables value, 104
tabularx key, 66
tabularx* key, 66
tcb fill frame key, 145

tcb fill interior key, 145
tcb fill title key, 145
tcbclipframe environment, 163
tcbclipinterior environment, 165
tcbcliptitle environment, 165
\tcbcontinuedraftmode, 192
\tcbcounter, 100
\tcbdocmarginnote, 435
\tcbdocnew, 435
\tcbdocupdated, 435
tcbexternal environment, 417
\tcbEXTERNALIZE, 416
\tcbfitdim, 381–384
\tcbfitsteps, 382
\tcbfontsize, 381
\tcbheightfromgroup, 61
\tcbheightspace, 137
\tcbhighmath, 328
\tcbifexternal, 423
\tcbifoddpage, 96
tcbimage comment key, 300
\tcbincludegraphics, 240
\tcbincludepdf, 242
\tcbinputlisting, 289
\tcbinputrecords, 119
\tcbinterruptdraftmode, 192
tcbinvclipframe environment, 164
\tcbitem, 268
tcbitemize environment, 268
\tcbline, 198
\tcbline*, 198
tcblisting environment, 287
\tcblistof, 107
\tcblower, 12
\tcbmakedocSubKey, 433
\tcbmaketheorem, 327
tcboutputlisting environment, 289
\tcbox, 14
tcbox raise key, 92
tcbox raise base key, 92
tcbox width key, 93
tcboxeditemize environment, 270
tcboxedraster environment, 269
\tcboxedtitleheight, 148
\tcboxedtitlewidth, 148
\tcboxfit, 380
\tcboxmath, 328
\tcboxverb, 409
\tcbpatcharcangular, 251
\tcbpatcharcround, 251
tcbraster environment, 267
\tcbrecord, 119
\tcbset, 13
\tcbsetforeverylayer, 13
\tcbsetmacrotoheightofnode, 252
\tcbsetmacrotowidthofnode, 252
\tcbsetmanagedlayer, 88
\tcbsettoheightofnode, 252
\tcbsettowidthofnode, 252
\tcbsidebyside, 114
\tcbstartdraftmode, 192
\tcbstartrecording, 119
\tcbstopdraftmode, 192
\tcbstoprecording, 119
\tcbsubskin, 132
\tcbsubtitle, 21
\tcbtextheight, 138
\tcbtextwidth, 137
\tcbtitle, 20
\tcbtitletext, 20
\tcbuselibrary, 9
\tcbuselistinglisting, 289
\tcbuselistingtext, 289
\tcbusetemp, 118
\tcbusetemplisting, 289
tcbverbatimwrite environment, 118
\tcbvignette, 253
tcbwritetemp environment, 118
tcolorbox environment, 12
\tcolorboxenvironment, 17
tempfile key, 92
terminator sign key, 332
terminator sign colon key, 333
terminator sign dash key, 333
terminator sign none key, 333
text above listing key, 305
text above* listing key, 305
text and listing key, 298
text fill key, 65
text height key, 53
text only key, 299
text outside listing key, 303
text side listing key, 303
text width key, 34
theorem key, 335
theorem name key, 334
theorem name and number key, 334
theorem number and name key, 334
theorem style key, 340
theorems key, 9
\thetcbcounter, 100, 101
\thetcbrasternum, 267, 278
\thetcolorboxnumber, 97
\thetcolorboxpage, 97
tight value, 43
tikz key, 183
tikz lower key, 67
tikz reset key, 183
tikz upper key, 67
tikznode key, 68
tikznode boxed title key, 155
tikznode lower key, 68
tikznode upper key, 68
tile key, 213
tile Skin, 213
tilefirst Skin, 214
tilelast Skin, 216
tilemiddle Skin, 215

title key, 18
title value, 43, 149
title after break key, 352
title code key, 131
title code app key, 400
title code pre key, 400
title empty key, 131
title engine key, 127
title filled key, 27
title hidden key, 143
title style key, 142
title style image key, 143
title style tile key, 143
titlerule key, 36
titlerule style key, 144
toggle enlargement key, 84
toggle left and right key, 45
top key, 41
top value, 33, 77, 109, 274
top seam value, 109
toprule key, 35
toprule at break key, 355
topsep at break key, 355
toptitle key, 41
true value, 79, 351

unbreakable key, 352
unbroken value, 156–158
unbroken and first value, 156–158
underlay key, 186
underlay boxed title key, 187
underlay boxed title pre key, 398
underlay broken key, 187
underlay broken pre key, 398
underlay first key, 187
underlay first and middle key, 187
underlay first and middle pre key, 398
underlay first pre key, 398
underlay last key, 187
underlay last pre key, 398
underlay middle key, 187
underlay middle and last key, 187
underlay middle and last pre key, 398
underlay middle pre key, 398
underlay pre key, 398
underlay raised fading vignette key, 261
underlay raised shading vignette key, 261
underlay shade in vignette key, 261
underlay unbroken key, 187
underlay unbroken and first key, 187
underlay unbroken and first pre key, 398
underlay unbroken and last key, 187
underlay unbroken and last pre key, 398
underlay unbroken pre key, 398
underlay vignette key, 260
unlimited value, 350, 351
updated key, 439
uphill value, 47, 48
upper right corner key, 254
upperbox key, 22

use counter key, 101
use counter from key, 101
use counter* key, 101
use height from group key, 61
\useboxarray, 375
\usetcboxarray, 376

valign key, 33
valign lower key, 33
valign scale limit key, 33
valign upper key, 33
value key, 439
Values
    all, 47, 48, 275, 354, 356
    areasize, 388
    areasize*, 388
    auto, 93
    auto limited, 93
    base, 77
    baselineskip, 354
    both, 115
    bottom, 33, 77, 109, 274
    bottom seam, 109
    break, 341
    broken, 156–158
    center, 30, 33, 77, 109, 274
    center seam, 109
    change, 341
    change apart, 341
    change break, 341
    change standard, 340
    clipped, 258
    colon, 105
    colon hang, 105
    copy, 149
    dash, 105
    dash hang, 105
    direct, 258
    doc, 437
    downhill, 47, 48
    east, 47, 48
    empty, 126, 127
    evenpage, 45, 84
    false, 76, 79, 276, 351
    fbox, 43
    figures, 104
    final, 390
    first, 156–158, 356
    first and middle, 156, 356
    fitbox, 90
    flush center, 30, 32
    flush left, 30, 32
    flush right, 30, 32
    fontsize, 388
    fontsize*, 388
    foovalue, 428
    forced, 45, 84
    forced center, 93
    forced left, 93
    forced right, 93

freelance, 126, 127
hbox, 90
hybrid, 388
hybrid*, 388
ignored, 24
invisible, 22, 24
justify, 30
last, 156–158, 356
left, 30, 115, 274
listings, 104, 298
margin, 342
margin apart, 342
margin break, 342
maximum, 55
middle, 156–158, 356
middle and last, 156–158, 356
minimal, 43
minimum center, 93
minimum left, 93
minimum right, 93
minipage, 90, 268
minted, 298
none, 45, 84, 115, 275, 354, 356
normal, 43, 149
north, 47, 48
northeast, 47, 48
northwest, 47, 48
off, 390, 437
on, 390
path, 126, 127
pathfirst, 126, 127
pathfirstjigsaw, 126
pathjigsaw, 126
pathlast, 126, 127
pathlastjigsaw, 126
pathmiddle, 126, 127
pathmiddlejigsaw, 126
pgf, 437
pgfchapter, 437
pgfsection, 437
plain, 340
plain apart, 341
right, 30, 115, 274
rows, 275
scale, 33
scale*, 33
small, 43
south, 47, 48
southeast, 47, 48
southwest, 47, 48
spartan, 126, 127
squeeze, 388
standard, 126, 127, 149, 340
tables, 104
tight, 43
title, 43, 149
top, 33, 77, 109, 274
top seam, 109
true, 79, 351
unbroken, 156–158
unbroken and first, 156–158
unlimited, 350, 351
uphill, 47, 48
visible, 22, 24
west, 47, 48
values key, 439
varwidth boxed title key, 155
varwidth boxed title* key, 155
varwidth upper key, 68
verbatim key, 402
vertical size key, 256
vfill before first key, 356
vignette key, 9
visible key, 22
visible value, 22, 24

watermark color key, 161
watermark graphics key, 157
watermark graphics app key, 397
watermark graphics app on key, 397
watermark graphics on key, 157
watermark graphics pre key, 397
watermark graphics pre on key, 397
watermark opacity key, 159
watermark overzoom key, 160
watermark shrink key, 160
watermark stretch key, 161
watermark text key, 156
watermark text app key, 396
watermark text app on key, 396
watermark text on key, 156
watermark text pre key, 396
watermark text pre on key, 396
watermark tikz key, 158
watermark tikz app key, 397
watermark tikz app on key, 397
watermark tikz on key, 158
watermark tikz pre key, 397
watermark tikz pre on key, 397
watermark zoom key, 159
west fading, 258
west value, 47, 48
west size key, 255
west style key, 257
widget key, 222
widget Skin, 222
widgetfirst Skin, 223
widgetlast Skin, 225
widgetmiddle Skin, 224
width key, 34

xmax key, 254
xmin key, 254
xparse key, 10
xshift key, 148

ymax key, 254
ymin key, 254
yshift key, 148

yshift* key, 148
yshifttext key, 148