



# tubstyle – Hintergrundelemente für *tubslatex*\*

Enrico Jörns  
e dot joerns at tu minus bs dot de

30. November 2020

## Zusammenfassung

Diese Datei stellt Befehle zur Verfügung, um die Basiselemente des Corporate-Design im Hintergrund der Seite darzustellen. Dies geschieht unter Verwendung des Paketes `eso-pic`.

## Inhaltsverzeichnis

<b>1</b>	<b>Implementierung</b>	<b>1</b>
1.1	Optionen	1
1.2	Hilfsbefehle	4
1.3	Elemente – Interne Befehle	4
1.3.1	Trennline	4
1.3.2	Siegelbandlogo	5
1.3.3	Logo	8
1.3.4	Hintergrund	9
1.4	Elemente – Benutzer-Befehle	11
1.5	Optionen für <code>bgelement</code>	12
1.5.1	<code>bglayout</code> -Optionen	14
1.6	Kombination / Gaußraaster-Seiten	18

---

\*This document corresponds to `tubstyle ?`, dated ?.

# 1 Implementierung

Allgemein: Alle zu setzenden Elemente werden in dem Makro @next@shipout gespeichert und am Ende der Layoutdefinition zum Shipout-Picture hinzugefügt (eso-pic). Abhängig von den Optionen wird dieses dann für das aktuelle shipout oder alle shipouts gesetzt. Alte Shipout-Definitionen werden dann jeweils komplett gelöscht.

## 1.1 Optionen

```
1 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
2 \ProvidesPackage{tubsstyle}[\tubslatexVersion, file v0.1 Hintergrund-Darstellung für tubslatex]
3 \RequirePackage{xkeyval}
4 \RequirePackage{ifthen}
```

‘twoside’ Darstellung für zweiseitiges Layout umschalten.

```
5 \newboolean{tubsstyle@twoside}\setboolean{tubsstyle@twoside}{false}
6 \define@boolkey{tubsstyle.sty}[ts@]{twoside}[true]{%
7   \ifts@twoside
8     \setboolean{tubsstyle@twoside}{true}
9   \else
10    \setboolean{tubsstyle@twoside}{false}
11  \fi
12 }
```

‘sender’ Option bestimmt Position des Kommunikationsbereichs und Orientierung des Rasters. Mögliche Werte: top, bottom

Diese Information wird im shipout gespeichert!

```
13 \newboolean{tubsstyle@bottomsender}\setboolean{tubsstyle@bottomsender}{false}
14 \define@choicekey*{tubsstyle.sty}{sender}[\val\nr]{top,bottom}{%
15   \ifcase\nr\relax
16     \setboolean{tubsstyle@bottomsender}{false}
17   \or
18     \setboolean{tubsstyle@bottomsender}{true}
19   \fi
20 }
```

‘fullline’ Top-Linie über gesamte Seitenbreite darstellen (für Aushänge, etc.)

```
21 \newboolean{tubsstyle@fullline}\setboolean{tubsstyle@fullline}{false}
22 \define@boolkey{tubsstyle.sty}[ts@]{fullline}[true]{%
23   \ifts@fullline
24     \setboolean{tubsstyle@fullline}{true}
25   \else
26     \setboolean{tubsstyle@fullline}{false}
27   \fi
28 }
```

‘scifiposter’ Anpassungen für wissenschaftliches Layout

```
29 \def\ts@post@process{}
30 \define@boolkey{tubsstyle.sty}[ts@]{scifiposter}[true]{%
31   \ifts@scifiposter
32     \def\ts@post@process{
33       \disable@keys{bglayout}{designhelper}%
34       \let\bgelement\undefined
35     }
36   \else
37     \def\ts@post@process{
38       \disable@keys{bglayout}{bgcolor}%
39     }
40   \fi
41 }
```

Fehler bei unbekannter Option

```
42 \DeclareOption*{%
43   \PackageWarning{tubsstyle}{Unknown option ‘\CurrentOption’}{}%
44 }

45 ⟨*package⟩
46 \ExecuteOptionsX{sender=top, twoside=false, scifiposter=false, fullline=false}
47 \ProcessOptionsX*\relax
```

Lade benötigte tubslatex-Pakete.

```
48 \RequirePackage{tubslogo}
49 \RequirePackage{tubscolors}
50 \RequirePackage{tubstypearea}
```

Prüft, ob alte oder neue Version von eso-pic vorhanden und Passt die Reihenfolge in der tubsbox und eso-pic geladen werden entsprechend an. Sonst würde die Reihenfolge der Darstellungsebenen vertauscht.

```
51 \begingroup
52 \let\OldProvidesPackage\ProvidesPackage
53 \def\ProvidesPackage#1[#2]{\OldProvidesPackage#1[#2]\endinput}
54 \RequirePackage{eso-pic}[2010/10/06]
55 \let\ProvidesPackage\OldProvidesPackage
56 \newif\if@OldEsoPicVersion
57 \@ifpackagelater{eso-pic}{2010/10/06}{%
58   \global\@OldEsoPicVersionfalse
59 }{%
60   \global\@OldEsoPicVersiontrue
61 }
62 \global\expandafter\let\csname ver@eso-pic.sty\endcsname\relax
63 \global\expandafter\let\csname opt@eso-pic.sty\endcsname\relax
64 \endgroup
```

```

65 \if@OldEsoPicVersion%
66   \RequirePackage{tubsbox}
67   \RequirePackage{eso-pic}
68 \else%
69   \RequirePackage{eso-pic}
70   \RequirePackage{tubsbox}
71 \fi%

```

Temporäre Längen für Berechnungen

```

72 \newlength{\tmp@posx}
73 \newlength{\tmp@posy}

```

Flags für Optionen.

```

74 \newboolean{tubsstyle@logoright}\setboolean{tubsstyle@logoright}{false}
75 \newboolean{tubsstyle@singlepage}\setboolean{tubsstyle@singlepage}{false}
76 \newboolean{tubsstyle@designhelper}\setboolean{tubsstyle@designhelper}{false}

```

Merker für horizontale Positionierung, Plain-Darstellung und Bindekorrekturbehandlung des Siegelbandlogos

```

77 \newif\iftubslogo@inside\tubslogo@insidefalse
78 \newif\iftubslogo@plain\tubslogo@plainfalse
79 \newif\iftubslogo@expandinbcorr\tubslogo@expandinbcorrfalse

```

Execute commands if todonotes package is loaded. This fixes some layout problems with the todonotes boxes

```

80 \AtBeginDocument{%
81   \@ifpackageloaded{todonotes}{%
82     \reversemarginpar
83     \setlength{\marginparwidth}{2cm}
84   }{}
85 }

```

## 1.2 Hilfsbefehle

`\if@tubs@oddpag`  $\{\langle odd\ cmd\rangle\}\{\langle even\ cmd\rangle\}$

Switch um Kommandos zu definieren, die für gerade und ungerade Seiten unterschiedlich arbeiten. Ist nur aktiv, wenn Option `twosided` benutzt wird.

```

86 \providecommand{\if@tubs@oddpag}[2]{%
87   \ifthenelse{\boolean{tubsstyle@twoside}}{%
88     \Ifthispageodd{#1}{#2}%
89   }{#1}%
90 }

```

## 1.3 Elemente – Interne Befehle

### 1.3.1 Trennline

`\tubs@topline` Kommando, um eine horizontale Linie am oberen Rand des Kommunikationsbereichs zu platzieren.

```
91 \newcommand\tubs@topline{%
92   % Setze Bindungs-Offset abhängig von oddness
93   \if@tubs@oddpage{%
94     \setlength{\tmp@posx}{\tubspage@bcor}%
95   }{%
96     \setlength{\tmp@posx}{0mm}%
97   }%
98   % Randbreite dazuaddieren, wenn nicht Option 'fullline' gewählt wurde
99   \ifthenelse{\boolean{tubsstyle@fullline}}{%
100     \addtolength{\tmp@posx}{\tubsborderwidth}%
101   }
102   \ifthenelse{\boolean{tubsstyle@bottomsender}}{%
103     \setlength{\tmp@posy}{\paperheight-(\tubsborderwidth+\thetubspage@gauussum\tubsgaussbasehei
104   }{%
105     \setlength{\tmp@posy}{\tubsborderwidth+\thetubspage@gauussum\tubsgaussbaseheight}%
106   }%
107   \ifthenelse{\boolean{tubsstyle@fullline}}{%
108     \put(\LenToUnit{\tmp@posx},\LenToUnit{\tmp@posy})%
109     {{\color{tuRed}\rule{\paperwidth-\tubspage@bcor}{0.24ex}}}%
110   }{%
111     \put(\LenToUnit{\tmp@posx},\LenToUnit{\tmp@posy})%
112     {{\color{tuRed}\rule{\paperwidth-2\tubsborderwidth-\tubspage@bcor}{0.24ex}}}%
113   }
114 }
```

### 1.3.2 Siegelbandlogo

`\tubs@tubslogo@calcp`  $\{\langle side \rangle\}\{\langle xlengthvar \rangle\}\{\langle ylengthvar \rangle\}$

Berechnet Position des zu platzierenden Siegelband-Logos. *side* legt die Positionierungsseite fest und sollte einer der Werte 'links' oder 'rechts' sein. Mit *xlengthvar* und *ylengthvar* werden Längen angegeben in die das berechnete Ergebnis gespeichert wird.

```
115 \newcommand\tubs@tubslogo@calcp[3]{%
116   % Berechene y-Position, abhängig von Raster-Orientierung (topdown/bottomsender)
117   \ifthenelse{\boolean{tubsstyle@bottomsender}}{%
118     \setlength{\tmp@posy}{\paperheight%
119       -(\tubsborderwidth+\thetubspage@gauussum\tubsgaussbaseheight-0.25\tubslogoHeight)%
120     -\tubslogoHeight}%
121   }{%
122     \setlength{\tmp@posy}{%
```

```

123      \tubsborderwidth+\thetubspage@gausssum\tubsgaussbaseheight-0.25\tubslogoHeight}%
124    }%
125    % Berechne x-Position, abhängig von oddness
126    \if@tubs@oddpagel{%
127      \ifthenelse{\equal{#1}{left}}{%
128        \setlength{\tmp@posx}{\tubspage@bcor}%
129      }{%
130        \setlength{\tmp@posx}{\paperwidth-\tubslogoWidth}%
131        \setboolean{tubsstyle@logoright}{true}%
132      }%
133    }{%
134      \ifthenelse{\equal{#1}{left}}{%
135        \setlength{\tmp@posx}{\paperwidth-\tubslogoWidth-\tubspage@bcor}%
136        \setboolean{tubsstyle@logoright}{true}%
137      }{%
138        \setlength{\tmp@posx}{0cm}%
139      }%
140    }%
141    \setlength{#2}{\tmp@posx}%
142    \setlength{#3}{\tmp@posy}%
143  }

```

‘pos’ Wahl der vertikale Position des Siegelband-Logos.

```

144 \define@choicekey{tubslogo}{pos}[\val\nr]{left,right,inside,outside}{%
145   \ifcase\nr\relax%
146     \global\tubslogo@insidefalse%
147   \or%
148     \global\tubslogo@insidetrue%
149   \or%
150     \global\tubslogo@insidefalse%
151   \or%
152     \global\tubslogo@insidetrue%
153   \fi%
154 }

```

‘left’ Logo-Platzierung links bzw. innen bei zweiseitigem Layout

```

155 \define@key{tubslogo}{left}[true]{%
156   \setkeys{tubslogo}{pos=left}%
157 }

```

‘right’ Logo-Platzierung rechts bzw. außen bei zweiseitigem Layout

```

158 \define@key{tubslogo}{right}[true]{%
159   \setkeys{tubslogo}{pos=right}%
160 }

```

‘inside’ Logo-Platzierung innen bzw. links bei einseitigem Layout

```

161 \define@key{tubslogo}{inside}[true]{%
162   \setkeys{tubslogo}{pos=inside}%
163 }

```

‘outside’ Logo-Platzierung außen bzw. rechts bei einseitigem Layout

```

164 \define@key{tubslogo}{outside}[true]{%
165   \setkeys{tubslogo}{pos=outside}%
166 }

```

‘plain’ Darstellung des Siegelbandlogos mit einfarbigem Hintergrund, ohne Grafik

```

167 \define@boolkey{tubslogo}{plain}[true]{%
168   \ifKV@tubslogo@plain%
169     \tubslogo@plaintrue%
170   \else%
171     \tubslogo@plainfalse%
172   \fi%
173 }

```

‘inbcorr’ Führt den Hintergrund des Siegelbandlogos in den Bereich der Bindekorrektur fort, um optisch störende Abstände Zwischen Bindung und Siegelbandanfang zu vermeiden. Die Option wirkt sich nur auf linksseitig platzierte Logos aus.

```

174 \define@boolkey{tubslogo}{inbcorr}[true]{%
175   \ifKV@tubslogo@inbcorr%
176     \tubslogo@expandinbcorrtrue
177   \else%
178     \tubslogo@expandinbcorrfalse
179   \fi%
180 }

```

Standardpositionierung setzen.

```

181 \setkeys{tubslogo}{pos=left}%

```

`\tubs@tubslogo` [*⟨pos⟩*]

Dieses Kommando fügt das TU-Siegelband-Logo in den Hintergrund ein. Syntax: `\tubs@tubslogo[⟨pos⟩]`, wobei `pos` die horizontale Position des Logos bestimmt und entweder die Werte `left` oder `right` akzeptiert. Die Boolean-Variable `tubsstyle@logoright` wird gesetzt, um die automatische Positionierung des individuellen Logos zu ermöglichen. Dieses wird auf der entgegengesetzten Papierseite platziert. Mit `tubslogo@expandinbcorr` wird bei linksseitig platziertem Siegelband das Band bis in die Bindekorrektur hinein gezeichnet.

```

182 \newcommand\tubs@tubslogo[1][ ]{%
183   \setkeys{tubslogo}{#1}%
184   \iftubslogo@inside%

```

```

185 \tubs@tubslogo@calcpow{right}{\tmp@posx}{\tmp@posy}%
186 \else%
187 \tubs@tubslogo@calcpow{left}{\tmp@posx}{\tmp@posy}%
188 \fi%
189 \iftubslogo@expandinbcorr
190 \if@tubs@oddpagel{%
191 \iftubslogo@inside\else
192 \addtolength\tmp@posx{-\tubspage@bcor}%
193 \put(\LenToUnit{\tmp@posx},\LenToUnit{\tmp@posy}){%
194 \fboxsep0mm\colorbox{tuRed}{\parbox[b][\tubslogoHeight]{\tubslogoWidth+\tubspage@bcor}
195 \addtolength\tmp@posx{\tubspage@bcor}%
196 \fi
197 }{%
198 \iftubslogo@inside\else
199 \put(\LenToUnit{\tmp@posx},\LenToUnit{\tmp@posy}){%
200 \fboxsep0mm\colorbox{tuRed}{\parbox[b][\tubslogoHeight]{\tubslogoWidth+\tubspage@bcor}
201 \fi
202 }%
203 \fi
204 \iftubslogo@plain%
205 \put(\LenToUnit{\tmp@posx},\LenToUnit{\tmp@posy}){%
206 \fboxsep0mm\colorbox{tuRed}{\parbox[b][\tubslogoHeight]{\tubslogoWidth}{~}}}%
207 \else%
208 \put(\LenToUnit{\tmp@posx},\LenToUnit{\tmp@posy}){%
209 \tubslogo}%
210 \fi%
211 }

```

### 1.3.3 Logo

`\tubs@logo`  $\{\langle logo \rangle\}$

Makro, um das Institutslogo zum Seitenlayout hinzuzufügen. Dabei werden auch zweiseitige Layouts beachtet. Syntax: `\tubs@logo{\langle logos \rangle}`

```

212 \newcommand\tubs@logo[1]{%
213 \def\@ragged{\raggedleft}%
214 % Berechene y-Position, abhängig von Raster-Orientierung (topdown/bottomsender)
215 \ifthenelse{\boolean{tubsstyle@bottomsender}}{%
216 \setlength{\tmp@posy}{\tubsborderwidth}%1
217 }{%
218 \ifts@scifiposter
219 \setlength{\tmp@posy}{\tubscommunicationheight+1.5\tubsborderwidth}%1
220 \else
221 \setlength{\tmp@posy}{\tubscommunicationheight+2\tubsborderwidth}%1
222 \fi
223 }%
224 % calculate x pos of logo and line depending on oddness
225 \if@tubs@oddpagel{%

```



```

226 \ifthenelse{\boolean{tubsstyle@logoright}}{%
227   \setlength{\tmp@posx}{\tubsborderwidth+\tubspage@bcor}%
228   \def\@ragged{\raggedright}%
229 }{%
230   \setlength{\tmp@posx}{0.5\paperwidth+0.5\tubspage@bcor}%
231 }%
232 }{%
233   \ifthenelse{\boolean{tubsstyle@logoright}}{%
234     \setlength{\tmp@posx}{\tubsborderwidth}%
235     \def\@ragged{\raggedright}%
236   }{%
237     \setlength{\tmp@posx}{0.5\paperwidth-0.5\tubspage@bcor}%
238   }%
239 }%
240 \put(\LenToUnit{\tmp@posx},\LenToUnit{\tmp@posy}){%
241   \newdimen\tmp@logoheight
242   \ifts@scifiposter
243     \setlength\tmp@logoheight{\tubssenderheight-1.5\tubsborderwidth}%
244   \else
245     \setlength\tmp@logoheight{\tubssenderheight-2\tubsborderwidth}%
246   \fi
247   \parbox[b][\tmp@logoheight][c]{
248     {0.5\paperwidth-0.5\tubspage@bcor-\tubsborderwidth}{%
249       \setkeys{Gin}{height=\tmp@logoheight,width=0.5\textwidth,keepaspectratio}%
250       \usekomafont{institute}\@ragged%
251       \leavevmode\color{tuRed}\#1%
252     }%
253   }%
254 }

```

### 1.3.4 Hintergrund

`\tubs@background`  $\{\langle color \rangle\} \{\langle first \rangle\} \{\langle height \rangle\}$

Berechnet Längen für die Positionierung eines Hintergrundelementes vor. first:  
Erstes Element (von oben nach unten), height: Anzahl Elemente

```

255 \newcounter{tmp@calc@bgpos}
256 \newcounter{tmpb@calc@bgpos}
257 \newcommand\tubs@background[2]{%
258   % Makro |\@inv@arg| wird benutzt, um Argument 'inverted' an
259   % |calc@gauss@elementpos| zu übergeben, falls bottomsender-Layout gewählt wurde.
260   \def\@inv@arg{}%
261   \ifthenelse{\boolean{tubsstyle@bottomsender}}{%
262     \def\@inv@arg{inverted}%
263   }{}%
264   % calculate and set y pos (origin: lower left corner)
265   \calc@gauss@elementpos[\@inv@arg]{tmp@calc@bgpos}\#1%
266   \ifthenelse{\boolean{tubsstyle@bottomsender}}{%

```

```

267 \setlength{\tmp@posy}{%
268 \tubssenderheight-\thetmp@calc@bgpos\tubsgaussbaseheight}% pos hack
269 }{%
270 \setlength{\tmp@posy}{%
271 \tubsborderwidth-\thetmp@calc@bgpos\tubsgaussbaseheight}% pos hack
272 }%
273 % calculate x pos of logo and line depending on oddness
274 \if@tubs@oddpagel{%
275 \setlength{\tmp@posx}{\tubsborderwidth+\tubspage@bcor}%
276 }{%
277 \setlength{\tmp@posx}{\tubsborderwidth}%
278 }%
279 % calculate and set height
280 \calc@gauss@elementpos[\@inv@arg]{\tmpb@calc@bgpos}{\#1+\#2}%
281 % additional fixment hack because of inverted y orientation
282 \setcounter{tmp@calc@bgpos}{\thetmpb@calc@bgpos-\thetmp@calc@bgpos}%
283 \addtolength{\tmp@posy}{%
284 -\thetmp@calc@bgpos\tubsgaussbaseheight%
285 +\thetubspage@gausssum\tubsgaussbaseheight}%
286 }

```

`\tubs@background@colored` Fügt ein farbiges Hintergrund-Element im Gaußraster ein.

```

287 \newcommand\tubs@background@colored[3]{%
288 \tubs@background{\#2}{\#3}%
289 \put(\LenToUnit{\tmp@posx},\LenToUnit{\tmp@posy}){%
290 {{\setlength\fbboxsep{0mm}\colorbox{\#1}{%
291 \parbox[b]{\thetmp@calc@bgpos\tubsgaussbaseheight}%
292 {\paperwidth-2\tubsborderwidth-\tubspage@bcor}}}%
293 }}%
294 }

```

`\tubs@background@image` Fügt ein Bild-Hintergrund-Element im Gaußraster ein.

```

295 \newlength{\@image@xorig}
296 \newlength{\@image@xcalc}
297 \newlength{\@image@yorig}
298 \newlength{\@image@ycalc}
299 %
300 \newcommand\tubs@background@image[3]{%
301 \def\img@to@insert{%
302 \tubs@background{\#2}{\#3}%
303 \ifthenelse{\equal{\tubstyle@imagefit}{scaled}}{%
304 \def\img@to@insert{%
305 \includegraphics*[%
306 height=\thetmp@calc@bgpos\tubsgaussbaseheight,
307 width=\paperwidth-2\tubsborderwidth-\tubspage@bcor]{\#1}
308 }
309 }{\ifthenelse{\equal{\tubstyle@imagefit}{cropped}}{%
310 % Ermittelt, ob das Bild an den Seiten oder oben und unten beschnitten

```

```

311 % werden muss, um in den Darstellungsbereich zu passen
312 % Dazu wird die Höhe des auf korrekte Breite skalierten Bildes
313 % mit der Höhe des Darstellungsbereichs verglichen und entsprechend
314 % eine crop-Option gesetzt.
315 \settoheight{\@image@ycalc}{%
316   \includegraphics*[width=\paperwidth-2\tubsborderwidth-\tubspage@bcor]{#1}}
317 \ifthenelse{\lengthtest{\@image@ycalc>\thetmp@calc@bgpos\tubsgaussbaseheight}}{%
318   \renewcommand{\tubstyle@imagefit}{cropy}
319 }{%
320   \renewcommand{\tubstyle@imagefit}{cropx}
321 }
322 {}
323 \ifthenelse{\equal{\tubstyle@imagefit}{croxy}}{%
324   % Berechne abzuschneidende Ränder (oben+unten)
325   % Dazu wird die Differenz zwischen Darstellungsbereich und Höhe des
326   % korrekt auf die Breite skalierten Bildes berechnet und mit dem
327   % ermittelten Skalierungsfaktor multipliziert, sowie durch 2 geteilt.
328   % Das Ergebnis wird dann einmal am oberen und einmal am unteren Teil
329   % des (Original-)Bildes mit Hilfe der 'trim'-Option abgeschnitten.
330   \settoheight{\@image@yorig}{%
331     \includegraphics*{#1}}
332   \settoheight{\@image@ycalc}{%
333     \includegraphics*[width=\paperwidth-2\tubsborderwidth-\tubspage@bcor]{#1}}
334   \setlength{\@image@ycalc}{(\@image@ycalc-(\thetmp@calc@bgpos\tubsgaussbaseheight))*\ratio{\@
335     \setlength{\@image@ycalc}{0.5\@image@ycalc}
336   \def\img@to@insert{%
337     \includegraphics*[%
338       width=\paperwidth-2\tubsborderwidth-\tubspage@bcor,
339       trim=0pt {\@image@ycalc} 0pt {\@image@ycalc}]{#1}
340   }
341   }\ifthenelse{\equal{\tubstyle@imagefit}{cropx}}{%
342     \settowidth{\@image@xorig}{%
343       \includegraphics*{#1}}
344     \settowidth{\@image@xcalc}{%
345       \includegraphics*[height=\thetmp@calc@bgpos\tubsgaussbaseheight]{#1}}
346     \setlength{\@image@xcalc}{(\@image@xcalc-(\paperwidth-2\tubsborderwidth-\tubspage@bcor))*
347     \setlength{\@image@xcalc}{0.5\@image@xcalc}
348   \def\img@to@insert{%
349     \includegraphics*[%
350       height=\thetmp@calc@bgpos\tubsgaussbaseheight,
351       trim={\@image@xcalc} 0pt {\@image@xcalc} 0pt]{#1}
352   }
353   }{}
354 }
355 \put(\LenToUnit{\tmp@posx},\LenToUnit{\tmp@posy}){\img@to@insert}%
356 }

```

\substack{\draw@topline} Die Befehle \substack{\draw@topline} und \substack{\draw@tubslgo} dienen zum Schemen der Darstellung von Logo und Toplinie, um die Hintergrundreihenfolge kor-  
\substack{\draw@logo}

rekt zu gewährleisten.

```
357 \newcommand{\tubs@draw@topline}{}
358 \newcommand{\tubs@draw@tubslogo}{}
359 \newcommand{\tubs@draw@logo}{}

```

## 1.4 Elemente – Benutzer-Befehle

`\showtopline` Benutzer-Kommando: Ermöglicht das Darstellen einer Linie zwischen Kommunikations- und Absenderbereich. Setzt das Makro `\tubs@draw@topline`.

```
360 \newcommand{\showtopline}{\renewcommand{\tubs@draw@topline}{%
361   \g@addto@macro{\@next@shipout}{\tubs@topline}}}

```

`\if@tubs@showlogo` If-Abfrage, um testen zu können, ob Logo auf aktuellem Layout gesetzt wurde

```
362 \newif\if@tubs@showlogo\@tubs@showlogofalse

```

`\showtubslogo` [*<pos>*]

Benutzer-Kommando: Ermöglicht das Darstellen des TU-Logos. Setzt das Makro `\tubs@draw@tubslogo`.

```
363 \newcommand{\showtubslogo}[1][{}]{%
364   \@tubs@showlogotrue%
365   \renewcommand{\tubs@draw@tubslogo}{%
366     \thispagestyle{empty}%
367     \g@addto@macro{\@next@shipout}{\tubs@tubslogo[#1]}}%
368 }

```

`\showlogo` {*<logo>*}

Benutzer-Kommando: Ermöglicht das Darstellen des Instituts-Logos. Setzt das Makro `\tubs@draw@tubslogo`.

```
369 \newcommand{\showlogo}[1]{\renewcommand{\tubs@draw@logo}{%
370   \g@addto@macro{\@next@shipout}{\tubs@logo[#1]}}}

```

## 1.5 Optionen für bgelement

`\tubs@bg@cmd` Speicher für den Befehl zu Erstellung des Hintergrundes

```
371 \def\tubs@bg@cmd{}

```

`\tubstyle@imagefit` Speicher für die imagefit-Option

```
372 \newcommand{\tubstyle@imagefit}{cropped}

```

Zähler für das aktuelle y-Segment im Gaußraster, das bearbeitet werden kann, bzw. mit `tubsrow` erzeugt.

```
373 \newcounter{tubs@yseg@cnt}
```

‘`bgcolor`’ Option zum Setzen der Hintergrundfarbe eines Elements.

```
374 \define@key{bgelement}{bgcolor}{%
375   \edef\tubs@bg@cmd@args{{#1}}{\thetubs@yseg@cnt}{\@current@height}}
376   \expandafter\def\expandafter\tubs@bg@cmd\expandafter{%
377     \expandafter\tubs@background@colored\tubs@bg@cmd@args}
378 }
```

‘`bgimage`’ Option zum Einfügen eines Hintergrundbildes in ein Element.

```
379 \define@key{bgelement}{bgimage}{%
380   \edef\tubs@bg@cmd@args{{#1}}{\thetubs@yseg@cnt}{\@current@height}}
381   \expandafter\def\expandafter\tubs@bg@cmd\expandafter{%
382     \expandafter\tubs@background@image\tubs@bg@cmd@args}
383 }
```

‘`imagefit`’ Option gibt an, wie dargestelltes Bild skaliert bzw. beschnitten werden soll.

- `scaled` Das Bild wird sowohl horizontal als auch vertikal so skaliert, dass es in den vorgegebenen Bereich passt. Da dabei das Seitenverhältnis nicht berücksichtigt wird, kann es zu starken Verzerrungen führen und sollte daher nur für Bilder eingesetzt werden, deren Seitenverhältnis in etwa dem des Darstellungsbereichs entspricht.
- `clipped` Das Bild wird so in den Darstellungsbereich eingepasst, dass es entweder horizontal oder vertikal passend skaliert wird und in anderer Richtung (vertikal/horizontal) auf den Bereich zugeschnitten wird.
- `vclip` Ähnlich wie `clipped`, nur dass das Bild auf jeden Fall horizontal skaliert und vertikal beschnitten wird.
- `hclip` Wie `vclip`, nur dass das Bild vertikal skaliert und horizontal beschnitten wird.

```
384 %
385 \define@choicekey{bgelement}{imagefit}[\val\nr]{%
386   default,scaled,clipped,cropped,clipx,hclip,clipy,vclip,fitheight,fitwidth}{%
387   \ifcase\nr\relax
388     % default
389     \PackageInfo{tubsstyle}{%
390       Option 'imagefit' not set. Using standard value 'clipped'.
391     }
392     \renewcommand{\tubstyle@imagefit}{cropped}
393   \or
394     % scaled
395     \renewcommand{\tubstyle@imagefit}{scaled}
396   \or
```

```

397 % clipped
398 \renewcommand{\tubstyle@imagefit}{cropped}
399 \or
400 % cropped
401 \renewcommand{\tubstyle@imagefit}{cropped}
402 \or
403 % clipx
404 \renewcommand{\tubstyle@imagefit}{cropx}
405 \or
406 % cropx
407 \renewcommand{\tubstyle@imagefit}{cropx}
408 \or
409 % clipy
410 \renewcommand{\tubstyle@imagefit}{cropy}
411 \or
412 % cropy
413 \renewcommand{\tubstyle@imagefit}{cropy}
414 \or
415 % fitheight
416 \renewcommand{\tubstyle@imagefit}{cropx}
417 \or
418 % fitwidth
419 \renewcommand{\tubstyle@imagefit}{cropy}
420 \fi
421 }

```

`\bgelement` [*<options>*]{*<height>*}  
`\bgsegment` Hintergrund-Element,

```

422 \newcommand{\bgelement}[2][] {
423   \def\@current@height{#2}
424   \setkeys*{bgelement}{#1}
425   % Führt das gespeicherte Hintergrund-Kommando mit den gespeicherten Argumenten aus.
426   \expandafter\g@addto@macro\expandafter\@next@shipout\expandafter{%
427     \expandafter\renewcommand\expandafter\tubstyle@imagefit\expandafter{%
428       \tubstyle@imagefit}
429   \expandafter\g@addto@macro\expandafter\@next@shipout\expandafter{%
430     \tubs@bg@cmd}
431   \addtocounter{tubs@yseg@cnt}{#2}
432 }
433 \let\bgsegment\bgelement

```

### 1.5.1 bglayout-Optionen

‘pages’ Option legt fest auf welchen Seiten das definierte Layout zu sehen sein soll. Mögliche Werte sind `all` für alle folgenden Seiten oder `single` für Darstellung nur auf der aktuellen Seite.

```

434 \define@choicekey{bglayout}{pages}[\val\nr]{all,single}{%
435   \ifcase\nr\relax
436     \setboolean{tubsstyle@singlepage}{false}
437   \or
438     \setboolean{tubsstyle@singlepage}{true}
439   \fi
440 }

```

‘designhelper’ Defines if \iftubs@bglayout@designhelper.

```

441 \define@boolkey{bglayout}{designhelper}[true]{%
442   \setboolean{tubsstyle@designhelper}{true}
443 }

```

‘bgcolor’ Mit dieser Option kann der komplette Kommunikationsbereich mit einer gewünschten Hintergrundfarbe gefüllt werden. Wird kein Wert mit übergeben, so wird die Farbe tuGray20 benutzt.

Diese Option ist nur für wissenschaftliche Poster zulässig!

```

444 \define@key{bglayout}{bgcolor}[tuGray20]{%
445   \g@addto@macro{\@@next@shipout}{%
446     \ifthenelse{\boolean{tubsstyle@bottomsender}}{%
447       \put(0,\LenToUnit{\tubssenderheight}){%
448         \setlength\fbboxsep{0mm}\colorbox{#1}{%
449           \parbox[b][\paperheight-\tubssenderheight]{%
450             {\paperwidth}{~}}}%
451       }%
452     }{%
453       \put(0,0){%
454         \setlength\fbboxsep{0mm}\colorbox{#1}{%
455           \parbox[b][\paperheight-\tubssenderheight]{%
456             {\paperwidth}{~}}}%
457       }%
458     }%
459   }
460 }

```

‘border’ Erzeugt einen farbigen Rahmen. TODO: NICHT CD-konform!

```

461 \define@key{bglayout}{border}[tuBlack]{%
462   \g@addto@macro{\@@next@shipout}{%
463     \ifthenelse{\boolean{tubsstyle@bottomsender}}{%
464       \put(0,\LenToUnit{\tubssenderheight}){%
465         \setlength\fbboxsep{0mm}\colorbox{#1}{%
466           \parbox[b][\paperheight-\tubssenderheight]{%
467             {\paperwidth}{~}}}%
468       }%
469     }{%
470       \put(0,0){%

```

```

471         \setlength\fbboxsep{0mm}\colorbox{#1}{%
472         \parbox[b][\paperheight-\tubssenderheight]%
473         {\paperwidth}{~}}%
474     }}%
475 }%
476 }
477 }

```

`\@layout@pre` {*<options>*}

Layout-Voreinstellungen, Optionen, etc.

```

478% Makro für alle Kommandos, die im shipout zwingend als erstes
479% ausgeführt werden müssen
480 \newcommand*{\@next@shipout}{}
481 %
482 \newcommand{\@layout@pre}[1]{%
483   \@tubs@showlogofalse
484   \def\tubs@draw@topline{}
485   \def\tubs@draw@tubslogo{}
486   \def\tubs@draw@logo{}
487   \def\tubs@draw@designhelper{}
488   \def\@next@shipout{}
489   \def\@next@shipout{} % Makro to set content after keys
490   \def\@layout@preset{}
491   \setcounter{tubs@yseg@cnt}{1}
492   \setkeys*{tubsstyle.sty}{#1}
493   \edef\tmp@KV{{\bglayout}{pages=all,\XKV@rm}}
494   \expandafter\setkeys\tmp@KV
495   \g@addto@macro{\@next@shipout}{\@next@shipout}
496   \ClearShipoutPicture
497 }

```

`\only@ifshipout@allowed` Sicherheits-Check, der shipouts nur setzt, wenn die aktuelle Umgebung dies erlaubt.

Aktuell wird nur getestet, ob die tikz-library external geladen wurde, da diese inkompatibel zur tubslatex shipout-Routine ist und sonst Teile des bglayout im externalisierten Bild zu finden sind.

```

498 \newcommand\only@ifshipout@allowed[1]{%
499   \ifdefined\tikzifexternalizing
500     \tikzifexternalizing{}{%
501       #1%
502     }%
503   \else
504     #1%
505   \fi
506 }

```



`\@layout@post` Layout-Postprocessing, Setzen des Bildes, Zusicherung der Reihenfolge. Die unter dem Makro `\@next@shipout` gespeicherten Kommandos für den Aufbau des Hintergrundbildes werden in ein neues ShipoutPicture geschrieben.

```

507 \newcommand{\@layout@post}{%
508   \tubs@draw@topline
509   \tubs@draw@tubslogo
510   \tubs@draw@logo
511   \ifthenelse{\boolean{tubsstyle@designhelper}}{%
512     \setboolean{tubsstyle@designhelper}{false}%
513     \showdesignhelper%
514   }{}%
515   \tubs@draw@designhelper% TODO: keep?
516   \only@ifshipout@allowed{%
517     \ifthenelse{\boolean{tubsstyle@singlepage}}{%
518       \AddToShipoutPicture*{\@next@shipout}%
519     }{%
520       \AddToShipoutPicture{\@next@shipout}%
521     }%
522   }%
523 }

```

`\bglayout` [*options*]{*bglayout*}

Legt das darzustellende Layout mit den übergebenen Optionen (*options*) und dem übergebenem Inhalt (*layout*) fest.

```

524 \newcommand{\bglayout}[2][]{%
525   \@layout@pre{#1}
526   #2
527   \@layout@post
528 }

```

`\tubs@background@designhelper` {*ypos*}

```

529 \newcommand\tubs@background@designhelper[1]{%
530   \tubs@background{#1}{1}%
531   \put(\LenToUnit{\tmp@posx},\LenToUnit{\tmp@posy})%
532     {\setlength\fbboxsep{0mm}\fbbox%
533       \parbox[b][\thetmp@calc@bgpos\tubsgaussbaseheight]%
534         {\paperwidth-2\tubsborderwidth-\tubspage@bcor}{~}}%
535   }{}%
536 }
537 %

```

`\showdesignhelper`

```

538 \newcommand{\showdesignhelper}{%
539   \def\tubs@draw@designhelper{%

```

```

540 \g@addto@macro{\@next@shipout}{%
541 % Rahmen für links und rechts positioniertes TU-Logo
542 \tubs@tubslogo@calcpos{left}{\tmp@posx}{\tmp@posy}%
543 \addtolength{\tmp@posy}{\tubslogoHeight}%
544 \put(\LenToUnit{\tmp@posx},\LenToUnit{\tmp@posy}){%
545 \fboxsep0mm\fbox{\parbox[t][\tubslogoHeight]{\tubslogoWidth}{~}}}%
546 \tubs@tubslogo@calcpos{right}{\tmp@posx}{\tmp@posy}%
547 \addtolength{\tmp@posy}{\tubslogoHeight}%
548 \put(\LenToUnit{\tmp@posx},\LenToUnit{\tmp@posy}){%
549 \fboxsep0mm\fbox{\parbox[t][\tubslogoHeight]{\tubslogoWidth}{~}}}%
550 %
551 \tubs@background@designhelper{1}%
552 \tubs@background@designhelper{2}%
553 \tubs@background@designhelper{3}%
554 \tubs@background@designhelper{4}%
555 \tubs@background@designhelper{5}%
556 \tubs@background@designhelper{6}%
557 \tubs@background@designhelper{7}%
558 \tubs@background@designhelper{8}%
559 %
560 \AtTextUpperLeft{%
561 \fbox{\parbox[t][\textheight]{\textwidth}{~}}}%
562 }%
563 }%
564 }%
565 }

```

`\defbglayout` [*options*]{*name*}{*bglayout*}

Erstellt ein neues Layouts.

Layout und Optionen werden unter den Makros `\@bg@layout@options@name` und `\@bg@layout@name` gespeichert.

```

566 \newcommand{\defbglayout}[3][{}]{%
567 \expandafter\newcommand\csname @bg@layout@options@#2\endcsname{#1}
568 \expandafter\newcommand\csname @bg@layout@#2\endcsname{#3}
569 }

```

`\usebglayout` {*name*}

Benutzung von vordefinierten Layouts.

Führt `\layout` mit den unter dem Layoutnamen und dem Präfix `\@bg@layout@options@` bzw. `\@bg@layout@` gespeicherten Optionen und Layout-Definitionen aus.

```

570 \newcommand{\usebglayout}[2][{}]{%
571 \expandafter\edef\expandafter\@curr@opts\expandafter{%
572 \expandafter\csname @bg@layout@options@#2\endcsname}
573 \expandafter\bglayout\expandafter[\@curr@opts]{\csname @bg@layout@#2\endcsname}
574 }
575 </package>

```

## 1.6 Kombination / Gaußraster-Seiten

**gausspage** Erzeugt eine neue Gaußlayout-Seite. Steuert die Aufbaureihenfolge des Hintergrundes, indem die Vordergrundelemente des Hintergrundes (Logos, Linie), die in den Makros `\tubs@draw@topline`, `\tubs@draw@tubslogo` und `tubs@draw@logo` gespeichert sind, zuletzt in den Hintergrund gezeichnet werden.

```

576 \newenvironment{gausspage}[1][\]{%
577   \clearpage
578   \@layout@pre{pages=single,#1}
579   \gausspagesetup[nosep,#1]
580   \sffamily
581 }{%
582   \@layout@post
583   ~\newpage
584 }
```

**segment** [*options*]{*ysegments*}

Erzeugt einen neuen Inhalts-Segment.

**options:** Mögliche Optionen, **ysegments:** Anzahl Gauß-Elemente für Höhe. Mögliche Optionen:

**bgcolor** Hintergrundfarbe

**bgimage** Hintergrundbild

Erzeugt eine `gaussbox` mit voller Seitenbreite und angegebener Höhe. Die vertikale Position im Gaußraster wird automatisch aus den zuvor gesetzten Boxen bzw. ihren Höhen berechnet.

```

585 \newenvironment{segment}[2][\]{%
586   \def\@current@height{#2}% TODO: unschön
587   \setkeys*{bgelement}{#1}% Muss vorher gesetzt werden, um |\XKV@rm| nutzen zu können
588   % Sorgt dafür, dass alle Argumente erst expandiert und dann eingefügt werden!
589   % TODO...
590   \def\@extra@opt{}
591   \ifthenelse{\boolean{tubsstyle@bottomsender}}{\}{%
592     \if@tubs@showlogo%
593       \def\@extra@opt{logosep}
594     \else%
595       \def\@extra@opt{}
596     \fi%
597   }
598   \edef\@tubs@arg{[\@extra@opt,\XKV@rm]{1}{\thetubs@yseg@cnt}{6}{#2}}
599   \bgelement{#2}% TODO: how to skip gaussbox related options?
600   % Erzeuge gaussbox
601   \expandafter\gaussbox\@tubs@arg
602 }{%
603   \endgaussbox
604 }
```

`modulepage` Erzeugt eine neue Modul-Layout-Seite. Steuert die Aufbaureihenfolge des Hintergrundes, indem die Vordergrundelemente des Hintergrundes (Logos, Linie), die in den Makros `\tubs@draw@topline`, `\tubs@draw@tubslogo` und `tubs@draw@logo` gespeichert sind, zuletzt in den Hintergrund gezeichnet werden.

```
605 \newenvironment{modulepage}[1][\{%  
606   \clearpage  
607   \@layout@pre{pages=single,bgcolor=tuSecondaryLight,#1}%  
608   \showtubslogo  
609   \sffamily  
610   \modcontainer[#1]%  
611 }{\%  
612   \endmodcontainer  
613   \@layout@post  
614   ~\newpage  
615 }
```

Behandlungen für Paketende. Hier werden Optionen deaktiviert, Befehle gelöscht, etc.

```
616 \ts@post@process
```