Das TUD-Script-Bundle setzt das Corporate Design der Technischen Universität Dresden für \LaTeX\ um. Die enthaltenen Klassen und Pakete basieren auf dem KOMA-Script-Bundle und sind sehr eng mit diesen verwoben. Momentan ergänzen sie das Vorlagenpaket von Klaus Bergmann, das auf den Standard-\LaTeX\-Klassen basiert und als veraltet betrachtet werden kann. Die dazugehörigen Klassen sollen mittel- bis langfristig ersetzt werden.\footnote{Aktuell ist dies \texttt{tudbook}, geplant \texttt{tudfax}, \texttt{tudletter}, \texttt{tudform}, \texttt{tudhaus} und evtl. auch \texttt{tudbeamer}} Es handelt sich bei diesem Dokument \emph{nicht} um das Anwenderhandbuch sondern um den dokumentierten Quelltext der Implementierung von TUD-Script. Das Anwenderhandbuch kann über die Kommandozeile respektive das Terminal mit dem Aufruf \texttt{texdoc tudscr} geöffnet werden.

Inhaltsverzeichnis

\begin{itemize}
\item[a] \texttt{tudscr-version.dtx} \hspace{1cm} 6 \\
\item[b] \texttt{tudscr-base.dtx} \hspace{1cm} 9 \\
\item[4] Das Paket \texttt{tudscrbas}e – Basisdefinitionen für TUD-Script \hspace{1cm} 9 \\
\item[4.1] Notwendige Pakete für TUD-Script \hspace{1cm} 9 \\
\item[4.2] Grundlegende Befehle für TUD-Script \hspace{1cm} 10 \\
\item[4.2.1] Temporäre Makros zur internen Verwendung durch TUD-Script \hspace{1cm} 10 \\
\item[4.2.2] Robuster Test auf leeres Argument \hspace{1cm} 10 \\
\item[4.2.3] Robuster Test auf numerischen Ausdruck \hspace{1cm} 11 \\
\item[4.2.4] Test auf booleschen Ausdruck in Form eines Strings \hspace{1cm} 11 \\
\item[4.2.5] Test auf Angabe einer Länge \hspace{1cm} 11 \\
\end{itemize}
9  Der Satzspiegel für TUD-Script  74
9.1 Kompatibilität der Satzspiegeleinstellungen mit weiteren Paketen  74
9.1.1 Unterstützung von Schnittmarken mit dem Paket crop  74
9.1.2 Unterstützung der Klasse standalone  74
9.2 Definition der Größen und Maße in abhängig vom Papierformat  75
9.3 Optionen für den Satzspiegel  80
9.4 Realisierung der Satzspiegeleinstellungen  82

10  Der Seitenstil des Corporate Designs  104
10.1 Definition des Seitenstils mit dem Paket scrlayer-scrpage  104
10.1.1 Erweiterung der Seitenstilauswahl  111
10.1.2 Gestaltungsvarianten für Kopf- und Fußzeile  113
10.1.3 Gestaltung des Querbalkens in der Kopfzeile  115
10.1.4 Boxen für Layerinhalte  119
10.2 Der Kopfbereich der Seitenstile  119
10.2.1 Das Hauptlogo der Technischen Universität Dresden  121
10.2.2 Optionales Zweit- oder DRESDEN-concept-Logo  122
10.3 Der Fußbereich der Seitenstile  124
10.3.1 Optionales DRESDEN-concept-Logo oder Drittlogos  126
10.3.2 Optionaler Inhalt im Fußbereich  130
10.4 Optionen für das DRESDEN-concept-Logo  134
10.5 Umgebungsparameter für die neuen Seitenstile  137

11  Das Layout des Corporate Designs  142
11.1 Gestalt von Umschlagseite, Titel, Teile und Kapitel  142
11.2 Schrifieinstellung und Positionierung der Überschriften  150
11.3 Umsetzung des Layouts  153
11.3.1 Präambeln für Teile und Kapitel  160
11.3.2 Layoutumsetzung speziell für Teile  161
11.3.3 Layoutumsetzung speziell für Kapitel  164
11.3.4 Erzwungene Majuskeln für Überschriften  167

12  Titelei für die TUD-Script-Klassen  170
12.1 Optionen und Schriftelemente für die Titelei  170
12.2 Titelseite und Titelkopf  171
12.2.1 Parameter für Titelseite und Titelkopf  171
12.2.2 Die Definition der Titelseite  173
12.2.3 Die Definition des Titelkofpfes  179
12.2.4 Hilfsmakros für die Ausgabe  182
12.3 Einspaltige Titelei in zweielspaltigen Dokumenten  184
12.4 Umschlagseite (Cover)  186
o  tudscr-twocolfix.dtx  257
20  Das Paket twocolfix – Bugfix für den zweispaltigen Satz  257

p  tudscr-mathswap.dtx  260
21  Das Paket mathswap – Aktive Trennzeichen im Mathematikmodus  260

Anhang  263
Index  263
Änderungsliste  277
Teil a
tudscr-version.dtx

1 Die Version des TUD-Script-Bundles

Für alle Klassen und Pakete, die zum TUD-Script-Bundle auf KOMA-Script-Basis gehören
wird als erstes die aktuelle Version festgelg.

\TUDVersion
\TUDVersion@Check
\TUDVersion@KOMA
\TUDScriptVersion
\TUDScriptVersionNumber

Das Makro \TUDVersion gibt an, zu welcher TUD-Script-Version die Datei gehört. Die Klas-
sen und Pakete des Bundles verwenden dieses Makro außerdem zur eigenen Versionsangabe.
Je nachdem, ob \TUDVersion bereits definiert ist oder nicht, wird mit \TUDVersion@Check
die Definition überprüft oder eine globale Definition vorgenommen. Da das Ganze auch bei
der Erstellung der Dokumentation geschieht, wird \makeatletter innerhalb einer Gruppe
verwendet. Das Makro \TUDVersion@KOMA definiert die mindestens notwendige Version
von KOMA-Script. In \TUDScriptVersion wird die Versioninformation für den Anwender
definiert.

1 \begingroup
2 \catcode`\@11\relax
3 (*.\package\ |\ \class)\)
4 \ifx\newcommand\@undefined
5 \gdef\TUDVersion@Check#1{%
6 \gdef\TUDVersion{\space\space#1}%
7 \gdef\TUDScriptVersion{\space\space#1}%
8 \aftergroup\endinput%
9 \}
10 \else
11 (//\package\ |\ \class)\)
12 \ifx\TUDVersion\@undefined
13 \newcommand*\TUDVersion@Check[1]{{%
14 \gdef\TUDVersion{#1}%
15 \gdef\TUDScriptVersion{#1}%
16 \def\@tempa#1 v##2 ##3 \relax\#4 \relax{\gdef\TUDScriptVersionNumber{##2}}%
17 \edef\@tempb\@tempa\relax? ? \relax\relax%
18 \gdef\TUDVersion@KOMA{3.17}%
19 \}
20 \}
21 \else
22 \newcommand*\TUDVersion@Check[1]{{%
23 \def\@tempa{#1}%
24 \ifx\TUDVersion\@tempa\else%
25 \\latex@warning@no@line{%
26 \noexpand\TUDVersion\space is \\TUDVersion\MessageBreak%
27 but \#1 was expected\MessageBreak%
28 You should not use classes, packages or files from\MessageBreak%
29 different TUD-Script-Bundle versions}\%
30 \}
31 \fi%
32 \}
33 \}
34 (*.\package\ |\ \class)\)
35 \fi
36 (//\package\ |\ \class)
Als erstes das benötigte \LaTeX-Format.

\begin{verbatim}
\NeedsTeXFormat{LaTeX2e}
\tudscrmail Die E-Mail-Adresse für Supportanfragen.
\end{verbatim}

Mit diesen Befehlen erfolgt die Versionsangabe der TUD-Script-Klassen beim Laden. Dabei werden hier einmalig die Namen der jeweiligen Klassen definiert. Hierfür werden die Befehle \TUD@ProvidesClass für die Hauptklassen und \TUD@ProvidesWrapperClass für auf diesen aufbauende Wrapper-Klassen genutzt. Das erste Argument ist die zu definierende Klasse, das zweite Argument entspricht der Basis-Klasse. Das dritte Argument enthält eine kurze Beschreibung.

\begin{verbatim}
\providecommand*{\TUD@ProvidesClass}[3]{%
  \ProvidesClass{#1}[%
    ^^^J#3 (#2)%
  ]%
\newcommand*{\TUD@Class@KOMA}{#2}%
\ifx\TUD@Class@Parent\@undefined%
  \newcommand*{\TUD@Class@Name}{#1}%
  \TUD@Class@Info%
\fi%
}
\end{verbatim}

Beim Verwenden der Klassen wird in der log-Datei ein Vermerk mit \typeout erstellt.

\begin{verbatim}
\providecommand*{\TUD@Class@Info}{%
  \typeout{+---------------------------------------------------------------}%
  \typeout{| Corporate Design of Technische Universitaet Dresden}%
  \typeout{| Class: \TUD@Class@Name}%
  \typeout{| Version: \TUD@Version}%
  \typeout{| Author: Falk Hanisch (\tudscrmail)}%
  \typeout{| Forum: http://latex.wcms-file3.tu-dresden.de/phpBB3/}%
  \typeout{+---------------------------------------------------------------}%
}\end{verbatim}
Hier erfolgt nun die Identifizierung und die Zuweisung der Klassennamen.

74 ⟨book⟩ \TUD@ProvidesClass{tudscrbook}{scrbook}{document class} ⟨report⟩ \TUD@ProvidesClass{tudscrreprt}{scrreprt}{document class} ⟨article⟩ \TUD@ProvidesClass{tudscrartcl}{scratcl}{document class} ⟨poster⟩ \TUD@ProvidesClass{tudscrpstert}{srartcl}{poster class} ⟨doc⟩ \TUD@ProvidesClass{tudscrdoct}{srartcl}{source code documentation class} ⟨manual⟩ \TUD@ProvidesWrapperClass{tudscrmnll}{tudscrreprt}{manual class}

Die Benutzermakros für den Namen der verwendeten TUD-Script-Klasse.

80 \edef\TUDScriptClassName{\TUD@Class@Name}
81 \providecommand*\TUDClassName{\TUDScriptClassName}

2 Verwendbarkeit von TUD-Script-Paketen

Einige Pakete sind nur mit den TUD-Script-Klassen verwendbar. Diese erzeugen einen Fehler, wenn sie nicht mit diesen verwendet werden.

82 \ifx\TUD@Class@Name\@undefined
83 \PackageError{% (supervisor) tudscrsupervisor%
84 (comp & base) tudscrcmp%
85 (comp & book) tudscrcmp-book%
86 (comp & poster) tudscrcmp-poster%
87 (tutorial) tudscrtutorial%
88 ){Unsupported class found}{% This package can only be used with a class out of the\MessageBreak%
89 tudscr bundle (tudscrbook, tudscrreprt, tudscrartcl, tudscrpstert).}%
90 %}
91 \endinput
92 \fi

3 Das TUD-Script-Logo

Der Schriftzug von TUD-Script.

96 \ifdefined{TUDScript}%
97 \DeclareRobustCommand{TUDScript}{%
98 \ifdim{TUD-SCRIPT}{%
99 \textsf{T\kern.05em U\kern.05em D\kern.1em\text{-}kern.1em Script}%
100 }\csname xspace\endcsname%
101 %
102 %}
103 }%
104 /package | class}
Teil b

4 Das Paket tudscrbase – Basisdefinitionen für TUD-Script

Für die Erstellung der TUD-Script-Klassen werden die dafür benötigten Pakete eingebunden und Steuerungsbefehle definiert. Die Klassen sowie einige Pakete benötigen das Paket und laden dieses auch.

\begin{verbatim}
(*load*)
\RequirePackage{tudscrbase}[\TUD@Version]
(*/load*)
\end{verbatim}

4.1 Notwendige Pakete für TUD-Script

Das Paket \texttt{scrbase} wird zur Optionsdefinition benötigt, \texttt{scrlfile} für verschiedene Hooks vor sowie nach Klassen und Paketen.

\begin{verbatim}
(*base*)
\RequirePackage{scrbase}[2013/12/19]
\@ifpackagelater{scrbase}{2015/10/04}{}{%
\providecommand*{\kernel@ifstar}[1]{\kernel@ifnextchar*{\@firstoftwo{#1}}}%
\@ifpackagelater{scrbase}{2016/05/11}{}{%
\renewcommand*{\test@@ifdimunt}[1]{%
  \@tempswafalse
  \ifstr{#1}{pt}{\@tempswatrue}{%
    \ifstr{#1}{pc}{\@tempswatrue}{%
      \ifstr{#1}{in}{\@tempswatrue}{%
        \ifstr{#1}{bp}{\@tempswatrue}{%
          \ifstr{#1}{cm}{\@tempswatrue}{%
            \ifstr{#1}{mm}{\@tempswatrue}{%
              \ifstr{#1}{dd}{\@tempswatrue}{%
                \ifstr{#1}{cc}{\@tempswatrue}{%
                  \ifstr{#1}{sp}{\@tempswatrue}{%
                    \ifstr{#1}{ex}{\@tempswatrue}{%
                      \ifstr{#1}{px}{\@tempswatrue}{%}
                    }%}
                  }%
                }%
              }%
            }%
          }%
        }%
      }%
    }%
  }%
}\scr@ifpdfoutput{%
  \ifstr{#1}{px}{\@tempswatrue}{}%
}\scr@ifpdfoutput{%}
}\test@@ifdimunt{#1}%
\}
\}
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
Das Paket `fix-tudscrfonts` muss bereits vor der Dokumentklasse geladen werden und führt anschließend spezifische Anpassungen an diese aus. Dafür ist das Paket `scrlfile` notwendig.

Das Paket `etoolbox` wird für die Manipulation bereits definierter Makros sowie zur erweiterten Auswertung boolescher Ausdrücke benötigt.

Das Paket `kvsetkeys` erweitert `keyval` um die Möglichkeit, das Verhalten bei der Angabe eines nicht definierten Schlüssels festzulegen.

Das Paket `etexcmds` wird für die Manipulation bereits definierter Makros sowie zur erweiterten Auswertung boolescher Ausdrücke benötigt.

Das Paket `tudscrfonts` muss bereits vor der Dokumentklasse geladen werden und führt anschließend spezifische Anpassungen an diese aus. Dafür ist das Paket `scrlfile` notwendig.

Das Paket `etoolbox` wird für die Manipulation bereits deﬁnierter Makros sowie zur erweiterten Auswertung boolescher Ausdrücke benötigt.

Das Paket `kvsetkeys` erweitert `keyval` um die Möglichkeit, das Verhalten bei der Angabe eines nicht deﬁnierten Schlüssels festzulegen.

Das Paket `etexcmds` wird für die Manipulation bereits deﬁnierter Makros sowie zur erweiterten Auswertung boolescher Ausdrücke benötigt.

Das Paket `kvsetkeys` erweitert `keyval` um die Möglichkeit, das Verhalten bei der Angabe eines nicht deﬁnierten Schlüssels festzulegen.

4.2 Grundlegende Befehle für TUD-Script

4.2.1 Temporäre Makros zur internen Verwendung durch TUD-Script

Hier werden Makros deﬁniert, die allein für die interne temporäre Nutzung gedacht sind.

```
\tud@reserved
```

Ein Makro zur temporären Verwendung.

```
\newcommand*{\tud@reserved}{}
```

```
\tud@toks@
\eaddto@hook
```

Es wird ein token-Registrier reserviert, welches im weiteren Verlauf an allen möglichen Punkten verwendet wird. Mit \eaddto@hook wird – in Ergänzung zu \addto@hook – ein Makro zum Anfügen des expandierten Arguments an ein token-Register bereitgestellt.

```
\BeforePackage{tudscrbase}{%
\edef\tud@reserved{}% 
\edef\eaddto@hook{}% 
}%
```

Die Befehle für die Klasse `tudscrdoc` bereits deﬁnierten Befehle werden vor dem Laden von `tudscrbase` undefiniert gesetzt.

```
\ifxblank
```

Hiermit kann ein Argument geprüft werden, ob dieses blank ist (leer oder Leerzeichen). In seiner Syntax ist er identisch zu `\ifblank`, allerdings expandiert er im Gegensatz zu diesem das gegebene Argument.

```
\newcommand*{\ifxblank}{\ifstr{}}
```

File b: tudscr-base.dtx
4.2.3 Robuster Test auf numerischen Ausdruck

\texttt{\textbackslash ifxnumber} Dieser Befehl dient zum Testen, ob ein gegebenes Argument eine Zahl ist. Die Syntax lautet:
\texttt{\ifxnumber\langle Argument\rangle\{\text{Wahr}\}\{\text{Falsch}\}}

\begin{verbatim}
58 \newcommand*\ifxnumber[1]{%
59 \if\relax\detokenize\expandafter{\romannumeral-0#1}\relax%
60 \expandafter\@firstoftwo%
61 \else%
62 \expandafter\@secondoftwo%
63 \fi%
64 }
\end{verbatim}

4.2.4 Test auf booleschen Ausdruck in Form eines Strings

\texttt{\textbackslash ifstrbool} Dieser Befehl dient zum Testen, ob ein gegebener String als boolescher Ausdruck interpretierbar ist. Ist der String als „wahr“ interpretierbar, wird das zweite Argument ausgeführt. Kann der String als „falsch“ angesehen werden, entsprechen das dritte. Ist der String ein logischer Wert, kommt das letzte Argument zum Tragen. Die Syntax lautet:
\texttt{\ifstrbool\langle Argument\rangle\{\text{Wahr}\}\{\text{Falsch}\}\{\text{Andernfalls}\}}

\begin{verbatim}
65 \newcommand*\ifstrbool[4]{%
66 \ifstr{#1}{true}{#2}{%
67 \ifstr{#1}{on}{#2}{%
68 \ifstr{#1}{yes}{#2}{%
69 \ifstr{#1}{false}{#3}{%
70 \ifstr{#1}{no}{#3}{%
71 \ifstr{#1}{off}{#3}{%
72 \#4%
73 }%
74 }%
75 }%
76 }%
77 }%
78 }%
79 }
\end{verbatim}

4.2.5 Test auf Angabe einer Länge

\texttt{\textbackslash ifxlen} Hiermit kann getestet werden, ob das gegebene Argument einer Länge entspricht.

\begin{verbatim}
80 \newcommand*\ifxlen[1]{%
81 \begingroup%
82 \protected@edef\@tempa{#1}%
83 \@tempswafalse%
84 \expandafter\ifisdimen\expandafter{\@tempa}{\@tempswatrue}{}%
85 \expandafter\ifisskip\expandafter{\@tempa}{\@tempswatrue}{}%
86 \expandafter\ifisdimexpr\expandafter{\@tempa}{\@tempswatrue}{}%
87 \if@tempswa\edef\@tempa{\the\@tempa}\fi%
88 \expandafter\ifisdimen\expandafter{\@tempa}{%
89 \aftergroup\@firstoftwo%
90 }%
91 \aftergroup\@secondoftwo%
92 }%
93 \endgroup
94 }
\end{verbatim}
4.2.6 Test auf eine verwendbare Sprache

Hiermit kann getestet werden, ob das Argument als Sprache nutzbar ist.

\iflanguageloaded
Hiermit kann getestet werden, ob das Argument als Sprache nutzbar ist.

95 \newcommand*{\iflanguageloaded[1]}{%
96 \PackageWarning{tudscrbase}{%
97 Whether package 'babel' nor package 'polyglossia' has been loaded%
98 }%
99 \expandafter\@secondoftwo%
100 }
101 \AfterPackage*[babel] {%
102 \renewcommand*{\iflanguageloaded[1]}{%
103 \expandafter\in@\{#1,\bbel\loaded,\}%
104 \ifi% 105 \expandafter\@firstoftwo%
106 \else% 107 \expandafter\@secondoftwo%
108 \fi%
109 }%
110 }
111 \AfterPackage*[polyglossia] {%
112 \renewcommand*{\iflanguageloaded[1]}{\ifcsdef{#1@loaded}%
113 }
114 \protect\expandtwoargs
4.2.7 Expansion geschützter Makros

Im \LaTeX-Kern wird der Befehl \expandtwoargs definiert, welcher zwei Argumente in ein angegebenes Makro vollständig expandiert. Dabei erfolgt die Expansion der beiden Argumente aufgrund der standardmäßigen Verwendung von \edef allerdings vollständig und ohne die Beachtung von \protect.

\protected\expandtwoargs
Der Befehl \protected\expandtwoargs kann äquivalent genutzt werden, lässt dabei aber mit \protect geschützte Makros unberührt.

114 \providecommand*{\protected\expandtwoargs[3]}{%
115 \protected\edef\reserved@a{\noexpand#1{#2}{#3}}\reserved@a%
116 }

4.2.8 Division für rationale Zahlen

Mit \LaTeX 2e kann – ohne die Verwendung zusätzlicher Pakete – nicht ohne Weiteres eine Division durchgeführt werden, die als Ergenis eine rationale Zahl liefert. Dem wird hier Abhilfe geleistet.

\tud@divide
Der Befehl erwartet im ersten Argument das Makro, in welches das Ergebnis der Division expandiert werden soll, danach folgen Dividend und Divisor. Die Berechnung erfolgt in einer Gruppe, damit die benötigten Längenregister nach dieser erhalten bleiben.

117 \newcommand*{\tud@divide[3]}{%
118 \begingroup%
Die Division wird über temporäre Längenregister durchgeführt. Die zu teilenden Zahlen werden normiert, wodurch auch Längen mit unterschiedlichen Einheiten geteilt werden können.

\begin{verbatim}
\@defaultunits\@tempdima #2pt\relax\@nnil%
\@defaultunits\@tempdimb #3pt\relax\@nnil%
\end{verbatim}

Danach wird mit der größeren der beiden Zahlen der größtmögliche, durch 2 teilbare Faktor zur Berechnung ermittelt.

\begin{verbatim}
\@defaultunits\@tempdima #2pt\relax\@nnil%
\@defaultunits\@tempdimb #3pt\relax\@nnil%
\ifdim\@tempdima>\@tempdimb\relax%
\@whiledim\@tempdimb>1pt\relax\do{\setlength\@tempdima{.1\@tempdima} \setlength\@tempdimb{.1\@tempdima}}\% \setlength\@tempdimc{\@tempdima} \else% \@whiledim\@tempdima>1pt\relax\do{\setlength\@tempdima{.1\@tempdima} \setlength\@tempdimb{.1\@tempdima}}\% \setlength\@tempdimc{\@tempdimb} \fi%
\@tempcnta=1\relax%
\@whiledim\dimexpr\@tempcnta\@tempdimc\relax<.01\maxdimen\do{\multiply\@tempcnta by 10\relax}%
\setlength\@tempdimc{\@tempdimc/\@tempcnta}\
\end{verbatim}

Mit dem im temporären Zählerregister \@tempcnta gespeicherten Faktor wird der Dividend erweitert und der berechnete Quotient wieder gekürzt.

\begin{verbatim}
\setlength{\@tempdimc}{\dimexpr(\@tempcnta\@tempdima / \@tempdimb * \p@)\relax}% \setlength{\@tempdimc}{\dimexpr \@tempdimc/\@tempcnta\relax}%
\end{verbatim}

Das Ergebnis wird in das angegebene Makro expandiert.

\begin{verbatim}
\edef\@tempa{\endgroup\def\noexpand#1{\strip@pt\@tempdimc}}%
\edef\@tempa{\strip@pt\@tempdimc}
\end{verbatim}

4.2.9 Erzwungene Kleinschreibung von Strings

Um angegebene Werte bei Schlüssel-Wert-Paaren oder Schlüsselwörtern in bestimmten Feldern mit Sicherheit erkennen zu können, werden diese zwingend in Kleinbuchstaben geschieben.

\tud@lowerstring Das Makro wird mit \tud@lowerstring\{(Zielmakro)\}{(String)} benutzt.

\begin{verbatim}
\newcommand*\tud@lowerstring[2]{%
\protectededef#1[#2]%
\lowercase\expandafter{\edef#1{#1}}%
\end{verbatim}

File b: tudscr-base.dtx
4.2.10 Erweiterung von KOMA-Script-Schriftelementen

Es wird die Möglichkeit geschaffen, bestimmten Schriftelementen weitere Eigenschaften mitzugeben. Damit dies optionsabhängig geschehen kann und \addtokomafont nur einmalig verwendete werden muss, wird im Zweifelsfall einem Schriftelement ein Hilfsmakro \tud@font@koma@set zugewiesen, welches intern angepasst werden kann. Mit \tud@font@koma@set werden dem angegebenen Element die gewünschten Schriftattribute zugeteilt.

\newcommand*{\tud@font@koma@set}[2]{% 
  \ifcsdef{tud@font@koma@#1}{}{\addtokomafont{#1}{\csuse{tud@font@koma@#1}}}% 
  \csdef{tud@font@koma@#1}{#2\nobreak}% 
}\end{verbatim}

Mit \tud@font@koma@unset kann das angegebene Element wieder zurückgesetzt werden. Dabei wird das hinzugefügte Makro auf \relax gesetzt.

\newcommand*{\tud@font@koma@unset}[1]{% 
  \ifcsdef{tud@font@koma@#1}{}{\addtokomafont{#1}{\csuse{tud@font@koma@#1}}}% 
  \csdef{tud@font@koma@#1}{\relax}% 
}\end{verbatim}

Werden KOMA-Script-Schriftelemente auf ihre Ausgangsdefinition zurückgesetzt, so kann diesen mit \tud@font@koma@reset kann das passende Hilfsmakro abermals zugewiesen werden.

\newcommand*{\tud@font@koma@reset}[1]{% 
  \ifcsdef{tud@font@koma@#1}{\addtokomafont{#1}{\csuse{tud@font@koma@#1}}}{}% 
}\end{verbatim}

4.2.11 Optionsdefinition für TUD-Script

In Anlehnung an KOMA-Script werden hier Befehle zur Definition und Ausführung unterschiedlicher Klassenoptionen mithilfe der Funktionen aus dem scrbase-Paket erstellt.

\newcommand*{\TUDProcessOptions}{\FamilyProcessOptions{TUD}}
\newcommand*{\TUDExecuteOptions}{\FamilyExecuteOptions{TUD}}
\newcommand*{\TUDoptions}{\FamilyOptions{TUD}}
\newcommand*{\TUDoption}{\FamilyOption{TUD}}

Zuerst ein paar Makros zur einfachen Verwendung.

\newcommand*{\TUD@key}[1]{% 
  \DefineFamilyMember[{#1}]{TUD}% 
  \DefineFamilyKey[{#1}]{TUD}%;% 
}\end{verbatim}

Dies sind die Befehle zur Definition der Optionen. Klassenoptionen können entweder als Schalter (\TUD@ifkey) oder aber mit mehreren möglichen Werten (\TUD@numkey) definiert werden. Prinzipiell ist auch eine freie Definition mit anschließender Abarbeitung ohne die zuvor genannten Befehle möglich.

\newcommand*{\TUD@key}[1]{% 
  \TUD@set@keyval{#1}{\@currname.\@currext}{% 
  \TUD@numkey{#1}{(TUD)}% 
  \TUD@set@numkey{#1}{}% 
  \TUD@set@lengthkey{#1}{}% 
  \TUD@set@dimenkey{#1}{}% 
  \TUD@unknown@keyval% 
}\end{verbatim}

File b: tudscr-base.dtx
Dies sind die Befehle zur Definition einer booleschen Option.

\newcommand*{\TUD@ifkey}[1]{\DefineFamilyMember{TUD}\FamilyBoolKey[#1]{TUD}}

Dies sind die Befehle zur Definition einer Option mit definierten Werten.

\newcommand*{\TUD@numkey}[1]{\DefineFamilyMember{TUD}\FamilyNumericalKey[#1]{TUD}}

Um Dopplungen im Code zu vermeiden, werden für die numerische Schlüssel die booleschen Standardwertzuweisungen in einem Makro gespeichert.

\newcommand*{\TUD@bool@numkey}{\{false\}{0},\{off\}{0},\{no\}{0},\{true\}{1},\{on\}{1},\{yes\}{1}}

Dies sind die Befehle zur Definition einer Option zur Festlegung einer Länge.

\newcommand*{\TUD@lengthkey}[1]{\DefineFamilyMember{TUD}\FamilyLengthKey[#1]{TUD}}

Mit \TUD@set@dimenkey wird der übergebene Längenwert im spezifizierten Makro gespeichert.

\newcommand*{\TUD@set@dimenkey}[3]{\ifxlen{#3}{\def#2{#3}\FamilyKeyStateProcessed}{\TUD@unknown@keyval{#1}{#3}{dimens}}}
Basierend auf dem Namen des Schalters wird die notwendige, boolesche Variable erzeugt \( \texttt{\textbackslash if@tud@} \texttt{(\textbackslash Schaltername\texttt{)@lock}} \), welche im Falle des direkten Aufrufs des Schlüssels durch den Anwender, ein internes Überschreiben verhindert.

\begin{verbatim}
\newbool{\@tud@#2@locked} %
\TUD@key[{#1}]{#2}%
\end{verbatim}

Es kann intern über \( \texttt{\textbackslash if(\textbackslash Schaltername\texttt{)@lock}} \) geprüft werden, ob einem mit Schlüssel, der mit \( \texttt{\TUD@key@lock} \) definiert wurde, durch den Anwender ein explizites Verhalten zugewiesen wurde. Sollte dies nicht der Fall sein, kann dieser hiermit intern beliebig angepasst werden. Da es durch das Setzen der Option mit \( \texttt{\TUD@option} \) zu einer Sperrung kommt, muss diese folgend wieder rückgängig gemacht werden.

\begin{verbatim}
\newcommand*{\TUD@std@ifkey@lock}[3]{%
\ifbool{\@tud@#1@locked}{}{%
\TUD@set@ifkey{#1}{#2}{#3}%
\boolfalse{\@tud@#1@locked}%
}%
}%
\newcommand*{\TUD@set@ifkey@lock}[1]{%
\booltrue{\@tud@#1@locked}%
\TUD@set@ifkey{#1}%
}%
\let\TUD@std@numkey@lock\TUD@std@ifkey@lock
\newcommand*{\TUD@set@numkey@lock}[1]{%
\booltrue{\@tud@#1@locked}%
\TUD@set@numkey{#1}%
}%
\end{verbatim}

Mit \texttt{\cs@lock\{\texttt{(Name)}\}\{\texttt{(Definition)}\}} kann ein Befehl definiert werden, welcher intern nur mit \texttt{\cs@std@lock\{\texttt{(Name)}\}\{\texttt{(Definition)}\}} geschrieben wird, wenn nicht über eine entsprechende Option ein explizites Verhalten mit \texttt{\cs@set@lock\{\texttt{(Name)}\}\{\texttt{(Definition)}\}} zugewiesen wurde. Dies ist äquivalent zu \texttt{\TUD@std@ifkey@lock} und \texttt{\TUD@set@ifkey@lock} bzw. \texttt{\TUD@std@numkey@lock} und \texttt{\TUD@set@numkey@lock}.

\begin{verbatim}
\newcommand*{\cs@lock}[2]{%
\expandafter\newcommand\expandafter*\csname#1\endcsname{#2}%
\newbool{#1@locked}%
}%
\newcommand*{\cs@std@lock}[2]{%
\ifbool{#1@locked}{}{%
\csdef{#1}{#2}%
\boolfalse{#1@locked}%
}%
}%
\newcommand*{\cs@set@lock}[2]{%
\csdef{#1}{#2}%
\booltrue{#1@locked}%
}%
\end{verbatim}

Diese Makros dienen in Anlehnung an die vorherigen zum Definieren und Setzen von sperrbaren booleschen Schaltern. Dabei wird \texttt{\tud@setbool} definiert, um nicht nur \texttt{true} und \texttt{false} sondern auch alle anderen bekannten booleschen Zuweisungen (\texttt{yes} und \texttt{on} sowie \texttt{no} und \texttt{off}) mit diesen Befehlen verwenden zu können.
Mit diesen Befehlen wird es möglich, Originalbefehle sichern, nutzen und wiederherstellen zu können. Dies wird benötigt, um zwischen den einzelnen Layouts über Optionseinstellungen zu wechseln und dabei von einem definierten Anfangszustand auszugehen.

Mit dem Aufruf \texttt{\tud@cmd@store\{\textit{Befehlsname}\}} wird der im Argument angegebene Befehl in einem neuen Makro \texttt{@@tud@\textit{Befehlsname}}} gespeichert. Dieser kann danach beliebig angepasst werden. Soll der Befehl zu einem späteren Zeitpunkt auf den Orginalzustand zurückgesetzt, kann hierfür jederzeit der Befehl \texttt{\tud@cmd@reset\{\textit{Befehlsname}\}} verwendet werden. Durch das Makro \texttt{\tud@cmd@restore\{\textit{Befehlsname}\}} wird das mit \texttt{\tud@cmd@store} erstellte Hilfsmakro zusätzlich noch gelöscht. Der ursprüngliche Befehl kann als solcher mit \texttt{\tud@cmd@use\{\textit{Befehlsname}\}} weiterhin genutzt werden. Ein zuvor bereits gesicherter Befehl kann mit dem Makro \texttt{\tud@cmd@update\{\textit{Befehlsname}\}} auf die momentane Definition aktualisiert werden.
Dieses Makro dient zum Überprüfen, ob der zu sichernde beziehungsweise wiederherzustellende Befehl überhaupt definiert ist. Sollte das nicht der Fall sein, wird ein Fehler ausgegeben.

Die Befehle \tud@cmd@store und \tud@cmd@check werden für das Paket fix-tudscrfonts bereits außerhalb von tudscrbase definiert. Damit deshalb keine Probleme entstehen, werden diese kurzerhand vor dem Laden von tudscrbase undefiniert gesetzt.

Dieses Makro wird verwendet, wenn mit den Mitteln von etoolbox bereits vorhandene Befehle angepasst werden sollen (\apptocmd, \pretocmd, \patchcmd) und dies nicht gelingt.

\PackageWarning{tudscrbase}{It wasn't possible to patch '\@backslashchar#1'\MessageBreak% Please contact the TUD-Script maintainer\MessageBreak% via \tudscrmail. Without a bugfix an\MessageBreak% erroneous output may occur%}

Mit \TUD@parameter@family{\{Familienname\}}{\{Definitionen\}} können Schlüssel-Wert-Parameter für die optionalen Argumente von Befehle definiert werden. Das erste Argument definiert den Familiennamen für den jeweiligen Befehl, welcher eindeutig gewählt werden sollte. Dieser wird im Hilfsmakro \TUD@parameter@checkfamily gesichert. Dies soll im Zusammenspiel mit dem Makro \TUD@parameter@checkfamily dafür sorgen, dass die im Folgenden bereitgestellten Befehle \TUD@parameter@def, \TUD@parameter@let und \TUD@parameter@handler@macro – welche die eigentliche Definition der Parameter für den Benutzer bewerkstelligen – ohne die Angabe der Familie nur innerhalb des zweiten Argumentes von \TUD@parameter@family verwendet werden können.

Dieser Befehl prüft, ob eine Familie für den Paramter definiert wurde.

\PackageError{tudscrbase}{%
360 No family for \backslashchar#2 defined%
361 }
362 You have to use \backslashchar#2\space within the\MessageBreak%
363 second argument of \string\TUD@parameter@family. The first\MessageBreak%
364 argument of \string\TUD@parameter@family\space has to be\MessageBreak%
365 a unique family name. Alternatively, you can specify\MessageBreak%
366 the family name within the optional argument of\MessageBreak%
367 \backslashchar#2.
368 }
369 }
370 }

\TUD@parameter@def
\TUD@parameter@def\{\langle Name\rangle\}\{\langle Säumniswert\rangle\}\{\langle Verarbeitung\rangle\} nutzt \define@key aus dem keyval-Paket, um einen Schlüssel und dessen Verarbeitung zu definieren, wobei auf den zugewiesenen Wert innerhalb des zweiten obligatorischen Argumentes mit #1 zugegriffen werden kann.

\TUD@parameter@set
Mit \TUD@parameter@set\{\langle Familienname\rangle\}\{\langle Parameterliste\rangle\} wird die Verarbeitung aller gegebenen Parameter veranlasst. Normalerweise wird dieser Befehl nicht innerhalb des Argumentes von \TUD@parameter@family verwendet. In jedem Fall muss die zu verwendende Familie angegeben werden.

\TUD@parameter@nokey
Hierfür wird die angegebene Parameterliste sukzessive abgearbeitet. Elemente, welche in Schlüssel-Wert-Syntax angegeben wurden, bleiben unverändert…
Ursprünglich entwickelte sich dieser Ansatz aus der Situation, dass das Makro \cs{maketitle} in seiner Standarddefinition ein optionales Argument für eine Seitenzahl bereithalt, bei den \TUDScript-Klassen jedoch auch die Möglichkeit besteht, verschiedene zusätzliche Parameter für diesen Befehl zu verwenden. Um allerdings für den Anwender das gewohnte Vorgehen beibehalten zu können, wurde diese Sonderbehandlung implementiert.

\TUD@parameter@handler@macro

Durch \TUD@parameter@handler@macro kann definiert werden, wie mit einem zuvor nicht mit \TUD@parameter@def definiertem Parameter beziehungsweise gewöhnlichem optionalen Argument umzugehen ist. Dabei wird unterschieden, ob es sich bei dem unbekannten Parameter um ein normales optionales Argument oder um eine Angabe in Schlüssel-Wert-Syntax handelt. Dabei wurden einfache optionale Argumente zuvor durch \TUD@parameter@set dem Schlüssel \TUD@parameter@nokey als Wert zugewiesen, um Umlaute etc. ohne Bedenken verwenden zu können.

Das erste obligatorische Argument von \TUD@parameter@handler@macro wird für die Verarbeitung unbekannter Schlüssel-Wert-Paare genutzt, das zweite für einfache optionale Argumente.

\TUD@parameter@handler@value

Mit diesem Makro wird einem optionalen Argument, welches ohne Schlüssel angegeben wurde, ein expliziter Parameter zugewiesen, welcher damit gesetzt wird. Unbekannte Schlüssel-Wert-Argumente werden durch \kv@handled@false immer als Fehler zurückgemeldet.
Hiermit kann sowohl die zu verwendende Sprache als auch die Anzahl der gewünschten Spalten für bestimmte Umgebungen ohne die explizite Angabe eines Schlüssels festgelegt werden. Momentan betrifft das die beiden Umgebungen abstract und tudpage sowie Befehle und Umgebungen, welche auf letzterer basieren. Unbekannte Argumente in Schlüssel-Wert-Syntax werden nicht unterstützt.

Das Makro \TUD@parameter@error\{Parameter\}\{Werteliste\} gibt für den Fall einer ungültigen Wertzuweisung an einen bestimmten (Parameter) eine Warnung mit einem entsprechenden Hinweis auf gültige Werte innerhalb von (Werteliste) aus.

Ab und an ist es notwendig, bestimmten Quelltext gezielt in Abhängigkeit vom Ladezustand eines Paketes auszuführen.
Dieser Befehl dient zur Ausführung von Quelltext, falls ein Paket bis zum Ende der Dokumentpräambel nicht geladen wurde. Im ersten obligatorischen Argument wird das Paket angegeben, im zweiten der Quellcode.

```
\newcommand*{\TUD@UnwindPackage}[2]{\AtEndPreamble{%\ifpackageloaded{#1}{#2}}}
```

Mit diesen beiden Befehlen wird die Ausführung von Quellcode erst nach dem Laden des gewünschten Paketes oder – falls das Paket geladen wurde – direkt im Dokument ausgeführt. Im ersten Argument wird das Paket angegeben, im zweiten der Quellcode.

```
\newcommand*{\TUD@AfterPackage@set}[1]{\newbool{@tud@#1@loaded}}
\newcommand*{\TUD@AfterPackage@do}[2]{\ifcsdef{if@tud@#1@loaded}{\if@atdocument{\ifbool{@tud@#1@loaded}{#2}{}}}{\PackageError{tudscrbase}{\string\TUD@AfterPackage@set{#1} missing}{You have to set \string\TUD@AfterPackage@set{#1} before the usage of \string\TUD@AfterPackage@do{#1}{<code>} is possible.}}}{\PackageWarning{tudscrbase}{You should load package 'scrextend' right after the\MessageBreak\documentclass. Option \string\TUD@KOMAoptions{#1} is gobbled}}}
```

4.2.13 Bedingt verzögerte Ausführung von KOMA-Script-Optionen

Dieses Makro wird verwendet, um innerhalb der TUD-Script-Klassen zu unterscheiden, wie eine KOMA-Script-Option auszuführen ist. Wurde bereits das Paket `scrextend` oder eine KOMA-Script-Klasse geladen, erfolgt die Ausführung des Arguments direkt über `\KOMAoptions`. Andernfalls wird das angegebene Argument an die zuladende Klasse durchgereicht.

```
\newcommand*{\TUD@KOMAoptions}[1]{\ifdef{\KOMAClassName}{\let\TUD@KOMAoptions\KOMAoptions}{\renewcommand*{\TUD@KOMAoptions}[1]{\PackageWarning{tudscrbase}{You have to set \string\TUD@AfterPackage@set{#1} before the usage of \string\TUD@AfterPackage@do{#1}{<code>} is possible.}}}}
\AtEndPreamble{\ifpackageloaded{scrextend}{\let\TUD@KOMAoptions\KOMAoptions}}
```

File b: tudscr-base.dtx

499 ⟨ load & class ⟩
500 ⟨ *inherit ⟩
501 \renewcommand*{\TUD@KOMAoptions}[1]{\PassOptionsToClass[#1]{\TUD@Class@KOMA}}
502 \AfterClass{\TUD@Class@KOMA}{\let\TUD@KOMAoptions\KOMAoptions}
503 ⟨ /inherit ⟩
504 ⟨ *inherit ⟩
505 \renewcommand*{\TUD@KOMAoptions}[1]{\PassOptionsToClass[#1]{\TUD@Class@Parent}}
506 \AfterClass{\TUD@Class@Parent}{\let\TUD@KOMAoptions\KOMAoptions}
507 ⟨ /inherit ⟩
508 ⟨ /load & class ⟩

5 Externe Pakete für die TUD-Script-Klassen

Für die Verwendung der hier erstellten KOMA-Script-Wrapper-Klassen werden einige wenige Pakete eingebunden. Dabei wurde versucht, die Anzahl der Pakete möglichst gering zu halten und nur die wirklich notwendigen zu verwenden.

5.1 Erweiterte Definition von Umgebungen mit dem Paket environ

Die abstract-Umgebung wird im Vergleich zu den KOMA-Script-Klassen stark erweitert. Für diese sowie für die Umgebungen declarations und tudpage wird das Paket environ für die Umgebungsdefinition benötigt.

509 \RequirePackage{environ}[2013/04/01]


\tud@x@multicol@num \tud@x@multicol@check

Im Makro \tud@x@multicol@num wird die Anzahl der gewünschten Spalten in einer Umgebung für die Verwendung des multicol-Paketes gespeichert.

510 \cs@lock{tud@x@multicol@num}{1}

Der Befehl \tud@x@multicol@check prüft, ob das Paket multicol geladen wurde. Falls dies nicht der Fall ist, wird eine Warnung ausgegeben und die Änderung des Wertes über einen Parameter der Umgebungen tudpage respektive abstract sowie declarations über \set@set@lock verhindert.

511 \newcommand*{\tud@x@multicol@check}{%
512 \ifnum\tud@x@multicol@num>\@ne\relax%
513 \ClassWarning{\TUD@Class@Name}{%
514 The option ‘columns=\tud@x@multicol@num’ is only supported, when package ‘multicol’ is loaded%
515 }%
516 \cs@set@lock{tud@x@multicol@num}{1}%
517 \fi%
518 %}
519 }
520 \AfterPackage{multicol}{\renewcommand*{\tud@x@multicol@check}{\relax}}
5.2 Pakete für Grafiken und Farben

Es folgen die Pakete, welche bei Bedarf am Ende der Präambel geladen werden.

521 \AtEndPreamble{

Mit \texttt{graphicx} werden die Logos der TU Dresden sowie von Dresden Concept mit dem Befehl \texttt{\usepackagegraphics} u. a. auf der Titelseite eingebunden. Sollte lediglich das \texttt{graphics}-Paket geladen worden sein, so wird der Nutzer mit einer Warnung informiert, dass zusätzlich das \texttt{graphicx}-Paket geladen wird.

522 \ifpackageloaded{graphicx}{
523 \ifpackageloaded{graphics}{
524 \ClassWarningNoLine{\TUD@Class@Name}{
525 Package ‘graphics’ was superseded by ‘graphicx’,\MessageBreak
526 which now will be loaded automatically\MessageBreak
527 }
528 }
529 \RequirePackage{graphicx}[1999/02/16]
530 }

Mit dem Paket \texttt{tudscrcolor} werden die Befehle für die Auswahl der Farben des Corporate Designs definiert, welches wiederum \texttt{xcolor} lädt.

531 \RequirePackage{tudscrcolor}[\TUD@Version]
6 Die Hausschriften der Technischen Universität Dresden

Für das Corporate Design der Technischen Universität Dresden werden die Schriften Univers und DIN BOLD benötigt. Zur fehlerfreien Verwendung dieser Klassen sollten diese bitte installiert werden. Hier werden die \LaTeX-Befehle zur Verwendung der Schriften im Dokument definiert. Sollen nur kurze Passagen in einer anderen Schrift als der restliche Text gesetzt werden, so sollten anstatt der einfachen Schalter wie beispielsweise \univ... besser die Befehle \text...{Auszeichnungstext} verwendet werden.

Die Schriften des Corporate Designs können nicht nur mit den TUD-Script-Klassen sondern auch mit anderen \LaTeX-Dokumentklassen verwendet werden. Hierfür wird das Paket tudsocrfonts bereitgestellt. Damit kann auf die Installation der Schriftfamilien für die alten Klassen verzichtet werden. Im Vergleich zu dieser Installation werden dabei fehlende Glyphen ergänzt und Probleme beim Kerning behoben.

6.1 Verwendung durch die TUD-Script-Klassen und das Paket tudsocrfonts

Die Schriften des Corporate Designs der Technischen Universität Dresden werden standardmäßig durch die TUD-Script-Klassen verwendet. Für Dokumentklassen, welche nicht zum TUD-Script-Bundle gehören, können diese über das Paket tudsocrfonts geladen werden.

\begin{verbatim}
\PreventPackageFromLoading[%
\ClassWarning{\TUD@Class@Name}{%\The package 'tudsocrfonts' must not be used with\MessageBreak%
a TUD-Script class%}%
]%{tudsocrfonts}
\end{verbatim}

Sollte keine KOMA-Script-Klasse genutzt werden, wird zusätzlich das Paket scrextend benötigt.

\begin{verbatim}
\ifcsname KOMAClass\endcsname\else
\RequirePackage{scrextend}[2013/12/19]\fi
\end{verbatim}

Wird das Paket tudsocrfonts nach einer der Klassen verwendet, welche die veralteten Schriftfamilien für Univers und DIN BOLD definieren, so wird eine Warnung mit dem Hinweis auf das Paket fix-tudsocrfonts ausgegeben.

\begin{verbatim}
\tempswafalse%
\for{\tempa=}{{tudbook,tudbeamer,tudfax,tudform,tudhaus,tudletter,tudposter,tudmathposter}}{do{%
\ifclassloaded{\tempa}{%\if@tempswa\else\tempswa\let\tempb@\tempa\fi%\tempswawtrue\let\tempb\tempa\fi%}{}%}\fi%\if@tempswa%
\ifpackageloaded{fix-tudsocrfonts}{%\if@tempswa%
\tempswawfalse%\end{verbatim}

File c: tudscr-fonts.dtx
6.2 Auswahl des Fontformates

Das TUD-Script-Bundle unterstützt sowohl die Schriftauswahl im klassischen New Font Selection Scheme (NFSS) als auch das Einbinden der Schriften im OpenType-Format aus dem System über das Paket fontspec für LuaLaTeX bzw. XeLaTeX, wobei die letztere Variante nicht ideal ist. Zum einen funktioniert das Kerning nicht sonderlich gut und zum anderen sind keine Schriften für den mathematischen Satz vorhanden. Da die skriptbasierte Installation der PostScript-Schriften gut funktioniert, ist diese Variante zu bevorzugen. Trotzdem werden beide Varianten, die Schriftfamilie des Corporate Designs für den Fließtext und die Überschriften zu verwenden, angeboten.

\if@tud@x@fontspec@enabled

\newif\if@tud@univers
\newif\if@tud@universmath
\newif\if@tud@dinbold
\newif\if@tud@dinboldmath

\tud@univers\iffalse\fi
\tud@universmath\iffalse\fi
\tud@dinbold\iffalse\fi
\tud@dinboldmath\iffalse\fi

6.3 Prüfen der Schriftverfügbarkeit

Nachfolgend werden Makros definiert, mit denen geprüft werden kann, ob die notwendigen Schriften im passenden Fontformat auch installiert sind.

Zu Beginn werden Schalter und Hilfsmakros definiert.
Mit \texttt{\@tud@font@check} wird überprüft, ob die benötigten Schriften \texttt{Univers} und \texttt{DIN BOLD} im PostScript- bzw. OTF-Format installiert sind und die passenden Schalter gesetzt. Beim Nichtvorhandensein der Schriften wird eine Warnung ausgegeben und eine Rückfallebene definiert. Dies geschieht für OTF-Schriften mit dem Befehl \texttt{\IfFontExistsTF}. Für die Mathematikschriften ist es unerheblich, ob das Paket \texttt{fontspec} genutzt wird, da hier immer die PostScript-Schriften zum Einsatz kommen.

Existieren die Schriften, werden die Schalter auf \texttt{true} gesetzt, sonst wird eine Warnung ausgegeben, falls die Schriften eigentlich verwendet werden sollen.

Für jede der OTF-Schriften wird eine Warnung ausgegeben, falls diese nicht installiert ist. Um sicher zu gehen, dass das Layout komplett genutzt werden kann, ist es hinreichend, wenn eine der benötigten Schriften nicht existiert, um die Rückfallebene zu aktivieren.
Ist \texttt{fontspec} nicht aktiv, wird auf die Schriften für den Fließtext im gleichen Maße geprüft.

\begin{verbatim}
88 \def@tempc##1{% 
89 \ifnum\tud@cdfont@num>\z@\relax \texttt{true/heavy}
90 \tud@font@missing@wrn{##1}%
91 }%
92 \ifexpandafter\@tempa\expandafter{\encodingdefault}%
93 {\tud@univers}{\@tud@universtrue}{\@tempc{Univers}}%
94 \expandafter\@tempa\expandafter{\encodingdefault}%
95 {\tud@dinbold}{\@tud@dinboldtrue}{\@tempc{DIN-Bold}}%
96 \fi%
97 \ifboolexpr{bool {@tud@univers} and bool {@tud@dinbold}}{}{% 
98 \renewcommand*\tud@cdfont@num{0}%
99 \bool@std@lock{@tud@cdfmath}{false}%
100 }%
101 \undefined\tud@font@check%
102 }

104 \newcommand*{\tud@font@missing@wrn}[1]{%
105 \begingroup%
106 \def@tempa{%
107 ′#1′ font not found for ′\encodingdefault′ encoding.\MessageBreak%
108 }%
109 \if@tud@fonts@enabled%
110 \appto@tempa{%
111 It seems, you haven’t installed the required\MessageBreak%
112 OTF-font on your system%
113 }%
114 \else%
115 \ifstr{TU}{\encodingdefault}{%
116 \appto@tempa{%
117 You should load package ′fontspec′\MessageBreak%
118 or %
119 }% 
120 \appto@tempa{Please }%
121 }%
122 \@tempswafalse%
123 \ifstr{OT1}{\encodingdefault}{\@tempswatrue}{% 
124 \ifstr{T1}{\encodingdefault}{\@tempswatrue}{% 
125 \if@tempswa%
126 \appto@tempa{%
127 use one of the installation scripts listed\MessageBreak%
128 in the TUD-Script manual%
129 }% 
130 \else%
131 \appto@tempa{%
132 choose a possible encoding with\MessageBreak%
133 package ′fontenc′ (′OT1′ or ′T1′)%
134 }% 
135 \fi%
136 \fi%
137 \appto@tempa{.\MessageBreak A fallback layout is used}%
138 (*class)
139 \ClassWarning{\TUD@Class@Name}{\@tempa}%
140 (/class)
141 (*package)
142 \PackageWarning{tudscrfonts}{\@tempa}%
143 (/package)
\endgroup%
\end{verbatim}

File c: tudscr-fonts.dtx

6.4 Kompatibilität der Schriften

6.4.1 Majuskel-ß für Unicode-Prozessoren

Es wird für die Majuskelvariante der Glyphe „ß“ eine Rückfallebene (Substitution mit „SS“) definiert, da diese für Unicode-Engines standardmäßig nicht bereitgestellt wird. Um diese zu erkennen, wird auf die gleiche Methodik wie im Paket newunicodechar zurückgegriffen.

\begingroup
\catcode`\^=7
\catcode30=12
\catcode`\! =12
\edef\tud@reserved{\@gobble^^^^0021}
\expandafter\endgroup

Wurde eine Unicode-Engine erkannt (^^^^0021 wird als ein Token gelesen), dann werden die Kodierungen für kleines und großes „ß“ dahingehend angepasst, dass diese „ss“ bzw. „SS“ verwenden, wenn die entsprechende Glyphe nicht vorhanden ist.

\ifx\tud@reserved\@empty\relax
\RequirePackage{newunicodechar}[2012/11/12]\%
\uccode"00DF=1E9E%
\lccode"1E9E=00DF%
\newunicodechar{^^^^00df}{\iffontchar\font"00DF \symbol{"00DF}\else ss\fi}%
\newunicodechar{^^^^1e9e}{\iffontchar\font"1E9E \symbol{"1E9E}\else SS\fi}%
\fi

6.4.2 Ausrichtung von Überschriften und das Paket ragged2e

Die Überschriften in DIN BOLD sollen laut Corporate Design linksbündig und nie mit aktiver Silbentrennung gesetzt werden. Normalerweise ist das dies beim linksbündigen Satz bei \LaTeX der Fall. Allerdings ist es mit dem Paket ragged2e möglich, das Verhalten für den Flattersatz zu ändern und die Silbentrennung zu aktivieren. Mit der Option newcommands werden dabei die originalen Befehle überschrieben, vorher jedoch in \LaTeX... gesichert. Diese Makros dienen dazu, die Überschriften in jedem Fall – auch bei der Verwendung von ragged2e mit der Option newcommands – ohne Trennungen zu setzen.

\newcommand*{\tud@raggedright}{\raggedright}
\newcommand*{\tud@RaggedRight}{\raggedright}
6.4.3 Anpassungen für das Paket siunitx

Das Paket siunitx nutzt den Befehl \lseries für den Fall, dass für den Fließtext eine Schrift die Serie l verwendet. Dem wird hier Rechnung getragen. Außerdem muss bei der Auswahl der Schriften evtl. auf das Paket reagiert werden, weil es sich bei der Definition der Schriften für den Mathematikmodus auf \familydefault bzw. \rmfamily verlässt.

6.4.4 Anpassungen für die Klasse beamer

Die beamer-Klasse lädt standardmäßig serifenlose Mathematikschriften. Dies soll verhindert werden, damit die Auswahl über die Option cdmath erfolgen kann.

6.4.5 Mathematikschriften in Verbindung mit dem Paket bm

Das Paket bm stellt den Befehl \bm für fette und kurische Symbole im Mathematiktext bereit. Damit dies funktioniert, muss das Laden des Paketes auf das Ende der Präambel verzögert werden, um zuvor alle Einstellungen für die mathematischen Symbole vornehmen zu können.
6.5 Schriftauswahl und -optionen

6.5.1 Schriften für den Fließtext

Hier werden die Optionen und Befehle zur Schriftauswahl definiert.

Mit dieser Option wird die zentrale Benutzerschnittstelle für alle möglichen Schrifteinstellungen innerhalb Dokumentes geschaffen. Durch diese Option können sowohl die verwendete Schrift für den Fließtext, als auch die Stärke der Schrift geändert werden. Zusätzlich lässt sich die Verwendung von **DIN BOLD** für die Überschriften deaktivieren. Außerdem kann eingestellt werden, ob für den Mathematiksatz seriflose Schriften zum Einsatz kommen sollen.

```latex
\if@tud@cdfont@num
  \if@tud@cdfont@din
    \if@tud@cdfont@ultrabold
      Mit dieser Option wird die zentrale Benutzerschnittstelle für alle möglichen Schrifteinstellungen innerhalb Dokumentes geschaffen. Durch diese Option können sowohl die verwendete Schrift für den Fließtext, als auch die Stärke der Schrift geändert werden. Zusätzlich lässt sich die Verwendung von **DIN BOLD** für die Überschriften deaktivieren. Außerdem kann eingestellt werden, ob für den Mathematiksatz seriflose Schriften zum Einsatz kommen sollen.
    \newif\if@tud@cdfont@din
    \newif\if@tud@cdfont@ultrabold
    \@tud@cdfont@dintrue
    \newcommand*\tud@cdfont@num{1}
    \TUD@key{cdfont}[true]{%
      \TUD@set@numkey{cdfont}{@tempa}{% 
        \TUD@bool@numkey,%
        \{light\}{1},\{lightfont\}{1},\{lite\}{1},\{litefont\}{1},\{noheavyfont\}{1},%
        \{heavy\}{2},\{heavyfont\}{2},\{bold\}{2},\{boldfont\}{2},%
        \{normalbold\}{3},\{boldnormal\}{3},%
        \{ultrabold\}{4},\{boldultra\}{4},\{heavybold\}{4},\{boldheavy\}{4},%
        \{extendedbold\}{4},\{boldextended\}{4},%
        \{nodin\}{5},\{nodinbold\}{5},\{noDIN\}{5},\{noDINBold\}{5},%
        \{din\}{6},\{dinbold\}{6},\{DIN\}{6},\{DINBold\}{6},%
      }
      \{+class\}
      \{nohead\}{7},\{nocdhead\}{7},\{noheadcdfont\}{7},%
      \{nccdheadfont\}{7},\{nobar\}{7},\{nobarfont\}{7},%
      \{head\}{8},\{headfont\}{8},\{cdhead\}{8},\{headcdfont\}{8},\{cdheadfont\}{8},%
      \{bar\}{8},\{barfont\}{8},%
      \{lighthead\}{8},\{lightheadfont\}{8},\{lightfonthead\}{8},%
      \{lightcdhead\}{8},\{lightcdheadfont\}{8},\{lightcdeckhead\}{8},%
      \{lightbar\}{8},\{lightbarfont\}{8},\{lightfbar\}{8},%
      \{headlight\}{8},\{headlightfont\}{8},\{headfbar\}{8},%
      \{cdheadlight\}{8},\{cdheadlightfont\}{8},\{cdheadfbar\}{8},%
      \{barlight\}{8},\{barlightfont\}{8},\{barmfbar\}{8},%
      \{litehead\}{8},\{liteheadfont\}{8},\{litechfonthead\}{8},%
      \{litechcdhead\}{8},\{litechcdheadfont\}{8},\{litechcdeckhead\}{8},%
      \{litechbar\}{8},\{litechbarfont\}{8},\{litechfbar\}{8},%
      \{headlite\}{8},\{headlitefont\}{8},\{headlitfbar\}{8},%
      \{cdheadlite\}{8},\{cdheadlitefont\}{8},\{cdheadlifbar\}{8},%
      \{barlite\}{8},\{barlitefont\}{8},\{barlitfbar\}{8},%
      \{noheavyhead\}{8},\{noheavyheadfont\}{8},\{noheavyfonthead\}{8},%
      \{noheavycdeckhead\}{8},\{noheavycdfont\}{8},%
      \{noheavybar\}{8},\{noheavybarfont\}{8},\{noheavyfbar\}{8},%
      \{headnoheavy\}{8},\{headfonthefn\}{8},\{headnoheavyf\}{8},\{headnoheavycf\}{8},%
      \{cdecnofn\}{8},\{cdcncd\}{8},\{cdcncdf\}{8},\{cdcnofc\}{8},%
      \{barncf\}{8},\{barfnt\}{8},\{barof\}{8},%
      \{heavyhead\}{9},\{heavyheadfont\}{9},\{heavyfonthead\}{9},%
      \{cdeh\}{9},\{cdeh\}{9},\{cdeh\}{9},\{cdeh\}{9},%
      \{head\}{9},\{headf\}{9},\{head\}{9},%
      \{cdh\}{9},\{cdh\}{9},\{cdh\}{9},\{cdh\}{9},%
      \{bar\}{9},\{bar\}{9},\{bara\}{9},%
      \{nomath\}{10},\{nocdmath\}{10},%
```

File c: tudscr-fonts.dtx
Die ersten drei Werte dienen zum Umschalten auf die Schriften des Corporate Designs für den Fließtext.

Für die Definition der fetten Schriftstärke kommt je nach Einstellung ein anderer Schriftschnitt zum Einsatz.

Hier wird die Verwendung von **DIN BOLD** für die Überschriften festgelegt.

Die Schrift für den Querbalken.
Die Mathematikschriften.

Die Neigung der großen griechischen Buchstaben.

Die Verwendung von \texttt{fontspec}.

Wird kein gültiger Wert für die Option angegeben, so wird davon ausgegangen, dass es sich bei diesem um die gewünschte Schriftgröße handelt.

Die Option \texttt{relspacing} setzt einige Abstände und Längen in Abhängigkeit der aktuellen Grundschriftgröße, wenn sie aktiviert wurde. Die verwendete Grundschriftgröße kann mit
der KOMA-Script-Option fontsize eingestellt werden. Die Definiton der Längen wird durch das Makro \tud@font@skip@set vorgenommen.

\newif\if@tud@relspacing
\TUD@key{relspacing}[true]{%
\TUD@set@numkey{relspacing}{@tempa}{%
{absolute}{0},{classic}{0},{standard}{0},%
{relative}{1},{fontsize}{1},{font}{1},{size}{1}%
}\if[#1]%
\family\FamilyKeyState\FamilyKeyStateProcessed%
\ifcase@tempa\relax% false
\@tud@relspacingfalse%
\or% true
\@tud@relspacingtrue%
\fi%
\TUD@SpecialOptionAtDocument{tud@font@skip@set}%
\fi%
%
\DefineFamilyMember{KOMA}
\DefineFamilyKey{KOMA}{fontsize}{%
\TUD@SpecialOptionAtDocument{tud@font@skip@set}%
\FamilyKeyStateProcessed%}
%
Mit dem Paket trimspaces werden für die alternative Schriftauswahl mit \cdfont bzw. \textcdfont eventuell vorhandene Leerzeichen aus dem Argument entfernt. Auch beim Einlesen von Feldern wird dies genutzt.

\RequirePackage{trimspaces}[2009/09/17]

\tud@font@set
\if@tud@x@mweights@enabled
\bfseries@rm
\bfseries@sf
\bfseries@tt
\mdseries@rm
\mdseries@sf
\mdseries@tt
\rmdef@ult
\sfdef@ult
\ttdef@ult
\bfdef@ult
\mddef@ult
\ifatdocument%
\ifnum\tud@cdfont@num>\z@\relax% true/heavy
\if@tud@univers\else%
\tud@font@missing@wrn{Univers}%
\fi%
\if@tud@dinbold\else%
\tud@font@missing@wrn{DIN-Bold}%
\fi%
\renewcommand*\tud@font@set{%
\ifboolexpr{bool {\tud@univers} and bool {\tud@dinbold}}{}{%
\bool@std@lock{\tud@cdmath}{false}%
}%
\fi%

Sind die Schriften des Corporate Designs nicht vorhanden, wird die Option zurückgesetzt.

\ifboolexpr{bool {\tud@univers} and bool {\tud@dinbold}}{}{%
\renewcommand*\tud@cdfont@num{0}%
\bool@std@lock{\tud@cdmath}{false}%
}%
\fi%
Werden die Schriften des Corporate Designs deaktiviert, wird auf die zu Beginn gesicherten Schriften zurückgeschaltet. Innerhalb der Klassen wird auch die Schrift im Querbalken zurückgesetzt, wenn kein Layout im Corporate Design genutzt wird. 

Ist die Verwendung der Schriften des Corporate Designs gewünscht, werden die Schnitte der Standardschriften vorher gesichert, um diese später vollständig wiederherstellen zu können.

\[\text{mit und ohne Serifen}\]
Die Schrift im Querbalken wird bei den Klassen angepasst.

\cs@std@lock{\tud@head@font@num}{\tud@cdfont@num}%

Danach erfolgt die Definition der Schriften des Corporate Designs entweder mit dem Paket \texttt{fontspec} oder...

\if@tud@univers%
  \if@tud@x@fontspec@enabled%
    \setsansfont{Univers}{\tud@x@fontspec@set@univ}%
    \setmonofont{LMMono10-Regular}{\tud@x@fontspec@set@univ}%
    BoldFont={},%
    ItalicFont={},%
    BoldItalicFont={},%
    SmallCapsFont={},%
    FontFace={l}{\sldefault}{LMMonoLt10-Regular},%
    FontFace={m}{\sldefault}{LMMono10-Regular},%
    FontFace={b}{\sldefault}{LMMonoLt10-Bold},%
    FontFace={bx}{\sldefault}{LMMonoLt10-Bold},%
    FontFace={eb}{\sldefault}{LMMonoLt10-Bold},%
    FontFace={eb}{\sldefault}{LMMonoLt10-BoldOblique},%
    FontFace={m}{\sldefault}{LMMonoSlant10-Regular},%
    FontFace={b}{\sldefault}{LMMonoLt10-BoldOblique},%
    NFSSFamily=:\tud@univers %
  \else%
    \renewcommand*{\sfdefault}{\tud@univers}%
    \renewcommand*{\ttdefault}{\tud@univers}%
  \fi%
  \ifnum\tud@cdfont@num=\@ne\relax% true
    \if@tud@cdfont@ultrabold% ultrabold
      \renewcommand*{\bfdefault}{b}%
    \else% normalbold
      \renewcommand*{\bfdefault}{m}%
    \fi%
    \renewcommand*{\mddefault}{l}%
    \if@tud@cdfont@ultrabold% ultrabold
      \renewcommand*{\bfseries@rm}{m}%
      \renewcommand*{\bfseries@sf}{m}%
      \renewcommand*{\bfseries@tt}{m}%
    \else% normalbold
      \renewcommand*{\bfseries@rm}{b}%
    \fi%
    \renewcommand*{\mdseries@rm}{l}%
    \if@tud@cdfont@ultrabold% ultrabold
      \renewcommand*{\bfseries@sf}{l}%
    \else% normalbold
      \renewcommand*{\bfseries@sf}{m}%
    \fi%
  \else% heavy
    \if@tud@cdfont@ultrabold% ultrabold
      \renewcommand*{\bfdefault}{b}%
    \else% normalbold
      \renewcommand*{\bfdefault}{m}%
    \fi%
    \renewcommand*{\mddefault}{l}%
    \if@tud@cdfont@ultrabold% ultrabold
      \renewcommand*{\bfseries@rm}{m}%
      \renewcommand*{\bfseries@sf}{m}%
      \renewcommand*{\bfseries@tt}{m}%
    \else% normalbold
      \renewcommand*{\bfseries@rm}{b}%
    \fi%
    \renewcommand*{\mdseries@rm}{l}%
    \if@tud@cdfont@ultrabold% ultrabold
      \renewcommand*{\bfseries@sf}{l}%
    \else% normalbold
      \renewcommand*{\bfseries@sf}{m}%
    \fi%
  \fi%
\else% heavy
  \if@tud@cdfont@ultrabold% ultrabold
    \renewcommand*{\bfdefault}{b}%
  \else% normalbold
    \renewcommand*{\bfdefault}{m}%
  \fi%
  \renewcommand*{\mddefault}{l}%
  \if@tud@cdfont@ultrabold% ultrabold
    \renewcommand*{\bfseries@rm}{m}%
    \renewcommand*{\bfseries@sf}{m}%
    \renewcommand*{\bfseries@tt}{m}%
  \else% normalbold
    \renewcommand*{\bfseries@rm}{b}%
  \fi%
\fi%

Mit diesem Befehl wird nach der Auswahl der Schriftgröße die Länge einiger Abstände angepasst, die bei den KOMA-Script-Klassen nicht abhängig von dieser sind.

Es werden die Standardeinstellungen abhängig von cdfont entweder wiederhergestellt oder gesichert und neu definiert.
Hiermit erfolgt das eigentliche Setzen der Abstände.

\newcommand*{\tud@font@skip@set}{
  \smallskipamount=\partopsep \%
  \medskipamount=\partopsep \%
  \multiply\medskipamount by 2 \%
  \bigskipamount=\partopsep \%
  \multiply\bigskipamount by 4 \%
  \tud@skip@set{abovecaptionskip}{\textwidth[\bigskipamount*5/6]\textwidth[\relax]}\%
  \tud@skip@set{belowcaptionskip}{\z@}\%
  \tud@skip@set{columnsep}{\baselineskip}\%
}

Das Makro \tud@font@init wird für die Defintion aller Befehle zur Auswahl der Schriften des Corporate Designs verwendet.

\newcommand*{\tud@font@init}{
  \if@tud@x@fontspec@enabled\%
    \if@tud@univers\%
      \newcommand*{\tud@x@fontspec@set@univ}{
        \Ligatures=TeX,\%
        \BoldFont={},\%
        \ItalicFont={},\%
        \BoldItalicFont={},\%
        \SmallCapsFont={},\%
        \tud@x@fontspec@fam@univ,\%
        \tud@x@fontspec@set@univ,\%
        \tud@font@fallback\%
      }\%
      \newfontfamily{\tud@x@fontspec@fam@univ}{Univers}\[
        \tud@x@fontspec@set@univ,NFSSFamily=Univers\%
      ]\%
    \fi\%
    \if@tud@dinbold\%
      \newcommand*{\tud@x@fontspec@set@din}{
        \Ligatures=TeX,\%
        \BoldFont={},\%
        \ItalicFont={},\%
        \BoldItalicFont={},\%
        \SmallCapsFont={},\%
        \tud@x@fontspec@fam@din,\%
        \tud@x@fontspec@set@din,\%
        \tud@font@fallback\%
      }\%
      \newfontfamily{\tud@x@fontspec@fam@din}{DIN-Bold}\[
        \tud@x@fontspec@set@din,NFSSFamily=DIN-Bold\%
      ]\%
    \fi\%
  \fi\%
}\%

\newcommand*{\tud@x@fontspec@fam@univ}{Univers\%
  \tud@x@fontspec@set@univ,\%
  \tud@font@fallback\%
}\%
\newcommand*{\tud@x@fontspec@fam@din}{DIN-Bold\%
  \tud@x@fontspec@set@din,\%
  \tud@font@fallback\%
}\%
\newcommand*{\tud@x@fontspec@fam@univ}{Univers\%
  \tud@x@fontspec@set@univ,\%
  \tud@font@fallback\%
}\%
\newcommand*{\tud@x@fontspec@fam@din}{DIN-Bold\%
  \tud@x@fontspec@set@din,\%
  \tud@font@fallback\%
}\%

File c: tudcr-fonts.dtx
Es werden mehrere Schalter zur expliziten Auswahl für die Schriftschnitte Univers 45 Light, Univers 55 Regular, Univers 65 Bold, Univers 75 Black, Univers 45 Light Oblique, Univers 55 Regular Oblique, Univers 65 Bold Oblique sowie Univers 75 Black Oblique und DIN BOLD definiert. Diese können so im Dokument unabhängig von der Option cdfont genutzt werden. Bei der Schriftauswahl der DIN BOLD wird neben dieser noch die entsprechende Mathematikschrift aktiviert.

Sind die notwendigen Schriften nicht installiert, werden die Schalter für die Rückfallebene so definiert, dass diese eine Warnung ausgeben.

Nach der Univers-Schriftfamilie wird für DIN BOLD äquivalent verfahren.

Da der Schalter \dinbn innerhalb der Überschriften verwendet wird, sollte dieser zumindest auch auf den Überschriftenstil schalten, falls DIN BOLD selbst nicht installiert ist.
Außerdem werden für die Schriften noch die entsprechenden Textauswahlbefehle bereitgestellt.

\newcommand*\textuln{}\newcommand*\texturn{}\newcommand*\textubn{}\newcommand*\textuxn{}\newcommand*\textuls{}\newcommand*\texturs{}\newcommand*\textubs{}\newcommand*\textuxs{}\newcommand*\textdbn{}

\DeclareTextFontCommand\textuln{\univln}\DeclareTextFontCommand\texturn{\univrn}\DeclareTextFontCommand\textubn{\univbn}\DeclareTextFontCommand\textuxn{\univxn}\DeclareTextFontCommand\textuls{\univls}\DeclareTextFontCommand\texturs{\univrs}\DeclareTextFontCommand\textubs{\univbs}\DeclareTextFontCommand\textuxs{\univxs}\DeclareTextFontCommand\textdbn{\dinbn}

Zuletzt werden Makros bereitgestellt, welche normalerweise durch das Paket babel vorgehalten werden.

\def\tud@reserved##1##2{%\ifundef{##2}{}{\ProvideTextCommand{##1}{\encodingdefault}{\TextOrMath{##2}{\mbox{##2}}}\}}\def\tud@reserved{\SS}{SS}\def\tud@reserved{\glq}{\quotesinglbase}\def\tud@reserved{\grq}{\textquoteleft}\def\tud@reserved{\glqq}{\quotedblbase}\def\tud@reserved{\grqq}{\textquotedblleft}\def\tud@reserved{\flq}{\guilsinglleft}\def\tud@reserved{\frq}{\guilsinglright}\def\tud@reserved{\flqq}{\guillemotleft}\def\tud@reserved{\frqq}{\guillemotright}

Nach der optionsabhängigen Definition der Schriftauswahlbefehle wird der Initialisierungsbefehl undefiniert gesetzt.

\def\tud@font@init{}

Hier wird die Rückfallebene für die Schriftschalter sowie die dazugehörige Warnung definiert.

\newcommand*\tud@font@fallback[3]{\tud@font@fallback@wrn{\DIN-Bold}\tud@font@init\}\newcommand*\tud@font@fallback@wrn[1]{⟨\text*{class}⟩\ClassWarning{\TUD@Class@Name}⟨/class⟩}\newcommand*\tud@font@fallback@wrn[1]{⟨\text*{package}⟩\PackageWarning{\TUD@Package@Name}⟨/package⟩}
Die Klassen \texttt{tudposter} und \texttt{tudmathposter} definieren den Befehl \texttt{tudfont}, mit welchem die Schriften des Corporate Designs der Technischen Universität Dresden auch mit Klarnamen ausgewählt werden können. Zur Kompatibilität wird der Befehl \texttt{cdfont} definiert, welcher die gleiche Funktionalität bereitstellt. Zusätzlich wird der Befehl \texttt{textcdfont} bereitgestellt, der die Auswahl der Schrift als Textbefehl ermöglicht.
6.5.2 Schriften für den Mathematiksatz


Mit dieser Option kann die genutzte Standardschrift für den Mathematiksatz für das gesamte Dokument umgestellt werden.
Mit dieser Option kann die Neigung der griechischen Majuskeln geändert werden.

Um ggf. Anpassungen an den griechischen Majuskeln vornehmen zu können, wird das Makro `\tud@font@greek@loop` definiert, mit dem eine Namensliste der Buchstaben durchlaufen wird.

In Anlehnung an verschiedene Pakete für Mathematikschriften werden Symbole für aufrechte und kursive Majuskeln der griechischen Lettern definiert. Damit auch bei diesen Buchstaben ein Umschalten für die unterschiedlichen Schriften möglich ist, wird eine interne Version definiert und im Bedarfsfall der dazugehörige \LaTeX-Befehl mit dieser überschrieben.

Für die Mathematikschriften im Fließtext werden für die lateinischen Lettern die Univers-Schriften des Corporate Designs verwendet. Darin enthalten sind die auch die griechischen Buchstaben aus dem Paket `cmbright`. Zusätzlich werden aus dem `iwona`-Paket die mathematische Symbole verwendet. Es werden zwei Mathematikversion mit normalen sowie mit fetten Glyphen bereitgestellt.
It's recommended to load package `fontspec` with option `no-math`, if you want to use the corporate design fonts in math mode.

Das Paket `cmbright` bringt zusätzlich noch Schriftschnitte für das Paket `amsfonts` mit. Diese werden gegebenenfalls geladen.

Das Paket ` cmbright` bringt zusätzlich noch Schriftschnitte für das Paket `amsfonts` mit. Diese werden gegebenenfalls geladen.
Auch für die Schrift **DIN BOLD** werden mathematische Glyphen bereitgestellt. Diese wurden aus dem iwona-Paket entnommen. Auch die verwendeten Symbole stammen aus diesem Paket.

```latex
\if@tud@dinboldmath
\DeclareMathVersion{dinbold}\
\SetSymbolFont{operators}{dinbold}{OT1}{\tud@dinbold}{b}{n}\
\SetSymbolFont{letters}{dinbold}{OML}{\tud@dinbold}{b}{sl}\
\SetSymbolFont{symbols}{dinbold}{OMS}{\tud@dinbold}{b}{n}\
\SetSymbolFont{largesymbols}{dinbold}{OMX}{iwona}{ebc}{n}\
\SetMathAlphabet{\mathrm}{dinbold}{OT1}{\tud@dinbold}{b}{n}\
\SetMathAlphabet{\mathnormal}{dinbold}{OML}{\tud@dinbold}{b}{sl}\
\SetMathAlphabet{\mathbf}{dinbold}{OT1}{\tud@dinbold}{b}{n}\
\SetMathAlphabet{\mathsf}{dinbold}{OT1}{\tud@dinbold}{b}{n}\
\SetMathAlphabet{\mathit}{dinbold}{OT1}{\tud@dinbold}{b}{sl}\
\SetMathAlphabet{\mathtt}{dinbold}{OT1}{\tud@dinbold}{b}{n}\
\SetMathAlphabet{\mathcal}{dinbold}{OMS}{lmsy}{b}{n}\
\fi
```

Wird das Paket `unicode-math` verwendet, werden keine Einstellungen für die Mathematik-schriften vorgenommen...

```
\@ifpackageloaded{unicode-math}{\
\let\tud@font@math@set\relax\
\bool@set@lock{@tud@cdmath}{false}}\
```

Für die griechischen Buchstaben wird mit `\tud@font@greek@check@slanted` geprüft, ob durch ein Paket diese auf die kursive Variante gesetzt wurden. Ist dies der Fall, wird der Standardwert für die Option `slantedgreek` auf `true` gesetzt. Anschließend erfolgt die Initialisierung der griechischen Großbuchstaben mit dem Befehl `\tud@font@greek@init`.

```
\newcommand*{\tud@font@greek@check@slanted}{\
\let\@tempb\relax\
\let\@tempc\relax\
\DeclareMathSymbol{\@tempb}{\mathalpha}{letters}{0}\
\DeclareMathSymbol{\@tempc}{\mathord}{letters}{0}\
\@tempswafalse\
\ifnum\Gamma=\@tempb\relax\@tempswatrue\fi
```

Zum Schluss werden abhängig von der gewählten Einstellung für die Schriftart der Option `cdfont` die benötigten Mathematikversionen geladen.

```
\tud@font@math@set
```

Daran anschließend werden die **DIN BOLD**-Schriften in einer Box verwendet, um das Laden der Schriftdefinitionen am Dokumentbeginn zu forcieren.

```
\sbox\z@{\tud@dinbn$ $}
```

Dieser Befehl prüft die aktuelle Definition von `\Gamma` und vergleicht diese mit den kursiven Varianten des Buchstabens. Sind diese identisch, wird davon ausgegangen, dass ein Paket mit der Option `slantedGreek` oder ähnlich geladen wurde. In diesem Fall wird der voreingestellte Wert für die entsprechende TUD-Script-Option gesetzt.

```
\newcommand*{\tud@font@greek@check@slanted}{\
\let@tempb\relax\
\let@tempc\relax\
\DeclareMathSymbol{\@tempb}{\mathalpha}{letters}{0}\
\DeclareMathSymbol{\@tempc}{\mathord}{letters}{0}\
\@tempswafalse\ifnum\Gamma=\@tempb\relax\@tempswatrue\fi
```

File c: tudscr-fonts.dtx
Da die Definition der griechischen Buchstaben durch das Paket `flexisym` stark verändert wird, wird die Prüfung in jedem Fall vor besagtem Paket ausgeführt.

```
\BeforePackage{flexisym}{\tud@font@greek@check@slanted}
```

Dieses Makro sorgt dafür, dass neben den normalen Makros für die griechischen Majuskeln die zusätzlichen Befehle für das explizite Auswählen aufrechter und kursiver Buchstaben.

```
\newcommand*{\tud@font@greek@init}[1]{% 
\ifcsundef{up#1}{% 
\if@tempswa% 
\csletcs{up#1}{tud@up#1}%
\else% 
\csletcs{up#1}{#1}%
\fi% 
\else% 
\csletcs{var#1}{#1}%
\fi% 
\if@tempswa% 
\csletcs{up#1}{tud@up#1}%
\else% 
\csletcs{up#1}{#1}%
\fi% 
\fi% 
\tud@font@math@set
```


```
\newcommand*{\tud@font@math@set}{% 
\if@atdocument% 
\if@tud@cdmath% 
\if@tud@universmath\else% 
\tud@font@math@missing@wrn{Univers}%
\fi% 
\if@tud@dinboldmath\else% 
\tud@font@math@missing@wrn{DIN-Bold}%
\fi% 
\fi% 
\ifboolexpr{bool {@tud@universmath} and bool {@tud@dinboldmath}}{}{% 
\bool@set@lock{@tud@cdmath}{false}%
```

Wurde per Option die Verwendung der Mathematiksschriften im Corporate Design aktiviert, wird für den Fall, dass die Schriften des Corporate Designs nicht installiert sind, eine Warnung erzeugt.

```
\if@atdocument% 
\if@tud@cdmath% 
\if@tud@universmath\else% 
\tud@font@math@missing@wrn{Univers}%
\fi% 
\if@tud@dinboldmath\else% 
\tud@font@math@missing@wrn{DIN-Bold}%
\fi% 
\fi% 
\ifboolexpr{bool {@tud@universmath} and bool {@tud@dinboldmath}}{}{% 
\bool@set@lock{@tud@cdmath}{false}%
```

Sind die Mathematiksschriften im Corporate Design nicht vorhanden, wird die entsprechende Option zurückgesetzt.
Mit diesen beiden Befehlen kann innerhalb des Dokumentes problemlos zwischen griechischen Buchstaben für die CD-Schriften und den normalen gewechselt werden.

Mit den beiden Befehlen kann die Definition der griechischen Buchstaben entweder aufrecht bzw. kursiv gesetzt werden.

Samtliche Schrifteinstellungen erfolgen erst am Ende der Präambel, um etwaige Einstellungen irgendwelcher Schriftpakete beachten zu können. Dabei erfolgt das Setzen der Mathematiksschriften durch \AtBeginDocument innerhalb von \AtEndPreamble am Ende aller \AtBeginDocument.

Falls das Paket mweights geladen wurde, werden im Bedarfsfall die vom Paket erwarteten Makros für die Schriftstärken der einzelnen Schriftfamilien definiert. Beim Aufruf von \tud@font@set werden diese anschließend gesichert.
Damit \texttt{tudscrfonts} reibungslos mit den nachfolgenden Definitionen funktioniert, wird im Bedarfsfall der Befehl \texttt{sectfont} vorgehalten.

Nachdem auf die Schriften geprüft wurde, werden noch die Schriftschalter und die dazugehörigen Befehle definiert.

Wurde das Paket \texttt{siunitx} geladen, so werden die Schrifteinstellungen mit $\texttt{cdfont=false}$ deaktiviert und zu Beginn des Dokumentes auf die aktuelle Einstellung gesetzt.

Damit die Schrift im Dokument später noch umgestellt werden kann, werden sämtliche Einstellungen erst am Ende der Präambel gesichert. Deshalb wird der dafür verantwortliche Befehl \texttt{tud@font@set} hier zum ersten Mal verwendet. Damit ist es möglich, die Schriften eventuell geladener Pakete wie \texttt{lmodern} oder \texttt{libertine} vorher zu sichern und später wieder zu aktivieren.

Die Einstellungen für den Mathematiksatz erfolgen erst zu Dokumentbeginn, um insbesondere auf Einstellungen von \texttt{fontspec} reagieren zu können. Da es so gut wie keine freien OpenType-Schriften für den mathematischen Formelsatz gibt, werden \textit{immer} die PostScript-Schriften verwendet, falls diese installiert sind. Wurde jedoch das Paket \texttt{bm} angefordert, so müssen die Mathematikschriften spätestens zum Ende der Präambel initialisiert werden.
6.5.3 Schrifteinstellungen für Überschriften

\ifdin Der Befehl \ifdin prüft auf die Verwendung von DIN BOLD. Davon abhängig wird entweder das erste oder das zweite Argument ausgeführt. Dies kann für die Befehle aller Gliederungsebenen genutzt werden, um zwischen der Ausgabe im Dokument sowie Inhaltsverzeichnis und/oder Kolumnentitel zu unterscheiden.

Mit \tud@sec@fontface wird die Schriftart für Überschriften im Layout des Corporate Designs definiert. Normalerweise wird hierfür DIN BOLD verwendet, was allerdings per Option auch deaktiviert werden kann.

Für Überschriften in Majuskeln\footnote{Großbuchstaben} wird der Befehl \MakeTextUppercase, was den internen \LaTeX-Befehl \MakeUppercase verbessert, aus dem Paket textcase genutzt.

\tud@makeuppercase Der Befehl führt \MakeTextUppercase{(Text)} nur dann aus, wenn die richtige Schriftfamilie – sprich DIN BOLD – verwendet wird.

Aufgrund eines Fehlers im \LaTeX-Kernels liegt die Grundlinie für die beiden Gliederungsebenen section und subsection zu hoch. Mit dem Einfügen des vertikalen Freiraums für die Umlaute wird diese automatisch nach unten verschoben. Allerdings ist das ein ziemlich übler Hack.

\protect\vphantom{"A"O"U}
7 Anwenderbefehle für Eingabefelder

Das TUD-Script-Bundle für das Corporate Design der Technischen Universität Dresden definieren mehrere Felder, welche durch den Anwender festgelegt werden können und dadurch auf der Titelseite bzw. auf der Aufgabenstellung – falls das Paket \tudscr\supervisor zum Einsatz kommt – ausgegeben werden. Ein Großteil der definierten Felder wird unter anderem für den Satz der Titelseite benötigt.

Das Setzen einer speziellen Titelseite mit \LaTeX ist eines der häufigsten Anliegen. Dafür ist von Markus Kohm\(^4\) das \texttt{titlepage}-Paket entworfen worden. Um gleichzeitig konsistent zu diesem Paket zu sein, werden für die entsprechenden Felder Alias-Befehle definiert.

7.1 Textfelder für die TUD-Script-Klassen

Von den Klassen benötigte Formularfelder werden definiert. Für das Setzen von Feldern werden mit dem Makro \texttt{\trim@spaces} aus dem Paket \texttt{trimspaces} bei einem übergebenen Argument führende und angehängte Leerzeichen beseitigt.

Für die für die TUD-Kopfzeile kann mit \texttt{\faculty\{Fakultät\}} die Fakultät angegeben werden, welche im Makro \texttt{\@faculty} gespeichert wird. Das gilt ebenso für die Angabe von Einrichtung, Institut und Lehrstuhl bzw. Professur. Dies erfolgt mit den Makros \texttt{\department\{Fachrichtung\}}, \texttt{\institute\{Institut\}} sowie \texttt{\chair\{Lehrstuhl\}}, welche in den Feldern \texttt{\@department}, \texttt{\@institute} und \texttt{\@chair} gespeichert werden.

Das optionale Argument wird zur Kompatibilität zur Klasse \texttt{tudscr\poster} vorgehalten. Wird das Paket geladen, kann mit dem optionalen Argument die Angabe der Struktureinheiten im Fußbereich variiert werden. Dafür werden die Felder \texttt{\@faculty}, \texttt{\@department}, \texttt{\@institute} sowie \texttt{\@chair} definiert.

\begin{verbatim}
\newcommand*{\@faculty}{\tud@head@text@field{faculty}}
\newcommand*{\@department}{\tud@head@text@field{department}}
\newcommand*{\@institute}{\tud@head@text@field{institute}}
\newcommand*{\@chair}{\tud@head@text@field{chair}}
\end{verbatim}

\(^4\)Autor von KOMA-Script
Für die Angabe weiteren, freien Textzeile im Kopf. Dies ist laut Corporate Design nur in besonderen Ausnahmefällen gestattet.

Mit \texttt{\@extraheadline} wird der Inhalt eines Feldes in \texttt{@(Feld)} gespeichert. Der Befehl erwartet als erstes obligatorisches Argument den Feldnamen und als zweites den Inhalt. Entsprecht das dritte obligatorische Argument \texttt{@empty}, so wird in \texttt{@@(Feld)} ebenfalls das zweite Argument abgelegt, andernfalls das dritte.

Damit wird es für Poster möglich, die Befehle \texttt{\faculty, \department, \institute, \chair} und \texttt{\professor} dahingehend zu erweitern, dass unterschiedliche Angaben für die Kopf- und Fußzeile gemacht werden können. Wird eines der zuvor genannten Makros lediglich mit einem obligatorischen Argument verwendet, so enthalten Kopf und Fuß den gleichen Eintrag. Wird jedoch zusätzlich das optionale Argument genutzt, so wird dessen Inhalt im Fußbereich mit \texttt{\@footline\@write} ausgegeben.

Für die spätere Verwendung im Dokument des Titels – beispielsweise für die Aufgabenstellung oder die Selbstständigkeitserklärung – wird das Feld \texttt{@@title} definiert. In diesem wird der mit \texttt{\title} gesicherte Eintrag ohne die etwaigen Fußnoten gespeichert.
Die Ausgabe einer zusätzlichen Zeile mit \authormore{Textzeile} direkt unterhalb der Angabe des Autors auf der Titelseite, wird im Makro \@authormore gespeichert.

Das Feld \@@author soll lediglich die Autoren ohne weitere Anmerkungen enthalten. Deshalb werden die gewöhnlichen Formatierungsbefehle des Titels temporär unschädlich gemacht.

Die Makros aus \tud@split@author@list werden zu \@tempc gesetzt, um diese nach der Expansion weiter zu behandeln.

Da die Befehle für Zusatzinformationen unter Umständen ungewollte Leerzeichen im Feld \@@author hinterlassen, werden diese entfernt.

Studiengang für den Titel sowie den Kopf der Aufgabenstellung, wird im Makro \@course gespeichert.
\textbf{disziplinen} Studienrichtung bzw. Fachrichtung für Titel und Kopf der Aufgabenstellung, wird im Makro \texttt{\@discipline} gespeichert.

\begin{verbatim}
90 \newcommand*{\@discipline}{}
91 \newrobustcmd*{\discipline}[1]{\gdef{\@discipline}{#1}\ignorespaces}
\end{verbatim}

\textbf{publisher} Kleine Korrektur für KOMA-Script, der Befehl sollte im Singular stehen.

\begin{verbatim}
92 \providecommand*{\publisher}[1]{\publishers{#1}}
\end{verbatim}

\textbf{thesis} Art bzw. Typ der Abschlussarbeit kann \texttt{\thesis{\langle Abschlussarbeit\rangle}} angegeben werden und wird im Makro \texttt{\@thesis} gespeichert. Alternativ dazu kann auch der Befehl \texttt{\subject} verwendet werden. Mit dem Befehl \texttt{tud@thesis@do} wird in den Feldern \texttt{\@thesis} respektive \texttt{\@subject} nach bestimmten Schlagwörtern für Abschlussarbeiten o.ä. gesucht. Wird eines dieser Wörter gefunden, wird der entsprechende Bezeichner für dieses Feld gesetzt. Zusätzlich wird durch \texttt{tud@thesis@@do} dafür gesorgt, dass gegebenenfalls der Inhalt von \texttt{\@subject} in \texttt{\@thesis} verschoben und entsprechend die Option \texttt{subjectthesis} gesetzt wird. In \texttt{\@thesis} wird die angegebene Abschlussarbeit ohne etwaige Fußnoten gespeichert.

\begin{verbatim}
93 \newcommand*{\thesis}{}
94 \newcommand*{\@thesis}{}
95 \newcommand*{\thesis}[1]{\tud@thesis@do{thesis}{#1}}
96 \renewcommand*{\subject}[1]{\tud@thesis@do{subject}{#1}}
97 \newcommand*{\tud@thanks}{}
98 \newcommand*{\tud@thesis@do}[2]{{% 
99 \begingroup% 
100 \global\let\tud@thanks\relax% 
101 \def{\thanks}{\ifx{#1}{thesis}\protect{\gdef{\@thesis}{#2}}{}% 
102 \global\let\@tempa\@tempa% 
103 \endgroup% 
104 \ifstr{#1}{diss}{\tud@thesis@@do{#1}{\dissertationname}}{% 
105 \ifstr{#1}{doctoral}{\tud@thesis@@do{#1}{\dissertationname}}{% 
106 \ifstr{#1}{diploma}{\tud@thesis@@do{#1}{\dissertationname}}{% 
107 \ifstr{#1}{master}{\tud@thesis@@do{#1}{\dissertationname}}{% 
108 \ifstr{#1}{student}{\tud@thesis@@do{#1}{\dissertationname}}{% 
109 \ifstr{#1}{evidence}{\tud@thesis@@do{#1}{\dissertationname}}{% 
110 \ifstr{#1}{project}{\tud@thesis@@do{#1}{\projectpapername}}{% 
111 \ifstr{#1}{seminar}{\tud@thesis@@do{#1}{\seminarpapername}}{% 
112 \ifstr{#1}{term}{\tud@thesis@@do{#1}{\termpapername}}{% 
113 \ifstr{#1}{research}{\tud@thesis@@do{#1}{\researchname}}{% 
114 \ifstr{#1}{log}{\tud@thesis@@do{#1}{\logname}}{% 
115 \ifstr{#1}{report}{\tud@thesis@@do{#1}{\reportname}}{% 
116 \ifstr{#1}{internship}{\tud@thesis@@do{#1}{\internshiptitle}}{% 
117 \ifstr{#1}{#2}{}}}}}}))))))))}}% 
\end{verbatim}

File d: tudscr-fields.dtx
Der angestrebte Abschluss bzw. der zu erwerbende akademische Grad, welcher auf der Titelseite ausgegeben werden soll, wird im Makro \@graduation gespeichert. Zusätzlich kann als optionales Argument die Kurzform des akademischen Grades angegeben werden, wird in \@graduation gespeichert.

\newcommand*{\@graduation}{\nothing}
\newcommand*{\graduation}[2]{  \ifx\empty{#1}  \gdef{\@graduation}{\nothing}  \else  \gdef{\@graduation}{(#1)}  \fi  \gdef{\@graduation}{#2}}

Angabe des verantwortlichen Hochschullehrers für Titel und Aufgabenstellung, wird im Makro \@professor gespeichert.

\newcommand*{\@professor}{\nothing}
\newcommand*{\professor}[1]{\gdef{\@professor}{#1}}

(Erst- und Zweit-)Betreuer bei Abschlussarbeiten, wird in \@supervisor gespeichert. Mehrere Betreuer werden durch \and getrennt. Für Poster kann der Befehl als Äquivalent zu \contactperson genutzt werden.

\newcommand*{\@supervisor}{\nothing}
\newcommand*{\supervisor}[1]{\gdef{\@supervisor}{#1}}
Diese Feld ist für die Hilfesteller bei der Anfertigung der Abschlussarbeit, welche auf der Selbstständigkeitserklärung aufgeführt werden. Mehrere Hilfesteller werden durch \and voneinander getrennt.

Angabe einer externen Firma, wird im Makro \@company gespeichert.


Fachreferenten bei einer Dissertation, werden im Makro \@advisor gespeichert. Mehrere Fachreferenten werden durch \and getrennt.

Angabe der Matrikelnummer für Titelseite und Aufgabenstellung, wird in dem Makro \@matriculationid gespeichert.

Das Immatrikulationsjahr für den Titel wird in \@matriculationyear gespeichert.

Der Geburtsort für den Titel wird in \@placeofbirth gespeichert.

Für alle Formatvorlagen, welche eine Unterschriftenzeile bereitstellen, wird ein einheitlicher Abstand verwendet.

Als Abschluss der Selbstständigkeitserklärung für Ort und Unterschrift.
Die Angabe des Ortes mit \place für die Selbstständigkeitserklärung und ggf. das Datum wird im Makro \@place gespeichert und standardmäßig mit „Dresden“ gesetzt.

\newcommand*{\@place}{Dresden}
\newcommand*{\place}[1]{\gdef\@place{#1}}

Im Fußbereich eines Posters kann zusätzlich eine Web-Seite angegeben werden.

\newcommand*{\@webpage{}}
\newcommand*{\webpage}{%\begin{verbatim}
  \kernel@ifstar{\tud@webpage@simple}{\tud@webpage@hyper}{%\gdef\@webpage{#1}}\end{verbatim}\par}
\newcommand*{\tud@webpage@simple}[1]{\gdef\@webpage{#1}}
\newcommand*{\tud@webpage@hyper}[2]{\gdef\@webpage{#2}}
\AfterPackage{hyperref}{%\begin{verbatim}
  \renewcommand*{\tud@webpage@hyper}[#2][hidelinks]{%\begin{verbatim}
    \begin{verbatim}
    \begingroup
    \urlstyle{same}
    \hypersetup{#1}
    \url{#2}
    \endgroup
    \end{verbatim}%
  \end{verbatim}\end{verbatim}}%
%

7.2 Datumsfelder für die TUD-Script-Klassen

\printdate
Im Folgenden werden mehrere Datumsfelder definiert. Damit diese optional durch das isodate-Paket formatiert werden können, wird der zu diesem Paket gehörende Befehl \printdate in die Definition der eigentlichen Datumsfelder integriert. Sollte das Paket nicht geladen werden, so muss dieser Befehl trotzdem definiert sein.

\TUD@UnwindPackage{isodate}{\providecommand*{\printdate}[1]{#1}}
\tud@isodate@printdate

Damit die Datumsfelder definiert werden können und das isodate-Paket unterstützen, muss beim Festlegen der Datumsfelder einiges beachtet werden. So müssen beispielsweise leere Argumente und Sonderfälle separat betrachtet werden. Damit dies einheitlich für alle Felder geschehen kann, wird dieser Befehl genutzt. Dabei wird als erstes Argument der Befehlsname für das Datumsfeld übergeben, als zweites Argument der gewünschte Inhalt.

\newcommand*{\tud@isodate@printdate}[2]{%\begin{verbatim}
  \ifx\today#2\relax%\gdef#1{#2}\else%\ifblank{#2}{%\gdef#1{}}{\gdef#1{\printdate{#2}}}%\end{verbatim}\end{verbatim}%

File d: tudscr-fields.dtx
Das Makro wird vor der Nutzung des Datumfeldes genutzt um zu prüfen, ob ein selbiges explizit angegeben wurde. Falls dies nicht der Fall ist, wird eine Warnung ausgegeben.

```latex
\newcommand*{\tud@date@check}{%
  \ifdefvoid{\@duedate}{%
    \ClassWarning{\TUD@Class@Name}{%\string\date' was not given}%
    Since a thesis is a self-contained work, an end%\string\date should be specified by the author.\MessageBreak%
    Nevertheless, today's date is used%\MessageBreak%
  }{%}
}%
\renewcommand*{\date}{%
  \kernel@ifstar{%
    \TUD@parameter@set{date}{pre=\noexpand\@place{,}}%
    \tud@date@set%  
  }{\tud@date@set}%
}\
```

Das Abgabedatum der Arbeit für den Titel wird im originalen Makro `\date` gespeichert. Zusätzlich kann über die Parameter im optionalen Argument eine Ergänzung hinzugefügt werden – beispielsweise als Erklärung für eine verspätete Abgabe aufgrund einer Verlängerung der Bearbeitungszeit – welche in den Feldern `\tud@date@pre` und `\tud@date@app` gespeichert wird. Hierfür wird der originale Befehl für das Datum `\date` erweitert, wobei die Sternversion `\date*` den angegebenen Ort als Präfix verwendet.

```latex
\renewcommand*{\tud@date@set}{%\
  \TUD@parameter@set{date}{#1}\
  \tud@x@isodate@printdate{\@date}{#2}\
  \global\let\tud@date@check\relax%\
}\n\newcommand*{\tud@date@pre}{}
\newcommand*{\tud@date@app}{}
\TUD@parameter@family{date}{%\
  \TUD@parameter@def{before}{\def\tud@date@pre{#1}}%\
  \TUD@parameter@let{pre}{before}\
  \TUD@parameter@def{prefix}{before}\
  \TUD@parameter@def{after}{\def\tud@date@app{#1}}%\
  \TUD@parameter@let{app}{after}\
  \TUD@parameter@let{suffix}{after}\
  \TUD@parameter@def{place}{\def\@place{#1}\def\tud@date@pre{#1,}}%\
  \TUD@parameter@handler@value{after}\
}\n```

Mit diesem Makro erfolgt die eigentliche Ausgabe des Datums und ggf. der optionalen Ergänzungen.
Das Verteidigungsdatum erscheint auf dem Titel und wird in \@defensedate gespeichert.  
\defensedate \@defensedate (Feld)

Angabe des Geburtstages für die Titelseite, wird im Makro \@dateofbirth gespeichert.  
\dateofbirth \@dateofbirth (Feld)

Die Klasse \texttt{tudscrposter} und das Paket \texttt{tudscrsupervisor} stellen einige weitere Felder bereit bzw. erweitern deren Funktionalitäten.  

Die Angabe des Prüfungsausschussvorsitzenden für die Aufgabenstellung, wird im Makro \@chairman gespeichert.  
\chairman \@chairman (Feld)

Die Befehle dienen zum Abspeichern der entsprechenden Parameter innerhalb der neu definierten Umgebungen aus dem Paket \texttt{tudscrsupervisor}.  
\grade \@grade (Feld)\@headline (Feld)

Angabe des Anfangsdatums für die Aufgabenstellung, wird im Makro \@issuedate gespeichert.  
\issuedate \@issuedate (Feld)

Angabe des geplanten Abgabedatums für die Aufgabenstellung, wird im Makro \@duedate gespeichert.  
\duedate \@duedate (Feld)
Einige Befehle werden nur definiert, falls das Paket nicht mit der Klasse \texttt{tudscrposter} verwendet wird – was allerdings nicht sinnvoll ist.

\begin{verbatim}
\*supervisor
\@ifclassloaded{tudscrposter}{\@tempswafalse}{\@tempswatrue}
\if@tempswa
\*/supervisor
\@contactperson
\@office
\@telephone
\@telefax
\@emailaddress
\contactperson(Feld)
\office(Feld)
\telephone(Feld)
\telefax(Feld)
\emailaddress(Feld)
\tud@emailaddress@simple
\tud@emailaddress@hyper
\end{verbatim}

Für einen Aushang bzw. ein Poster kann eine oder mehrere Kontaktpersonen angegeben werden. Zusätzlich lassen sich für jede einzelne Person ein Raum, eine Telefonnummer und die E-Mail-Adresse hinzufügen.

\begin{verbatim}
\newcommand*\@contactperson{}
\newcommand*\contactperson[1]{\gdef\@contactperson{#1}}
\let\supervisor\contactperson
\newcommand*\@office{}
\newrobustcmd*\office[1]{\gdef\@office{#1}}
\newcommand*\@telephone{}
\newrobustcmd*\telephone[1]{\gdef\@telephone{#1}}
\newcommand*\@telefax{}
\newrobustcmd*\telefax[1]{\gdef\@telefax{#1}}
\newcommand*\@emailaddress{}
\newrobustcmd*\emailaddress{\afterPackage*{hyperref}{\renewrobustcmd*\emailaddress[2][hidelinks]{\begingroup\urlstyle{same}\hypersetup{#1}\href{mailto:#2}{\nolinkurl{#2}}\endgroup}}}\newrobustcmd*\tud@emailaddress@simple[1]{\gdef\@emailaddress{#1}}
\newrobustcmd*\tud@emailaddress@hyper[2][2]{\gdef\@emailaddress{#2}}
\newcommand*\tud@multiple@split
\newcommand*\tud@multiple@@split
\newcommand*\tud@multiple@@@@split
\end{verbatim}

\textbf{7.4 Verarbeitung bestimmter Feldinhalte für die Ausgabe}

Einige Felder sind darauf ausgelegt, dass deren kommagetrennte Inhalte einzeln abgearbeitet und ausgewertet werden können. Die folgenden Makros übernehmen diese Aufgabe.

\begin{verbatim}
\tud@multiple@split
\tud@multiple@@split
\tud@multiple@@@@split
\end{verbatim}

Für Felder, die mehrere Personen – getrennt durch \texttt{\&} – beinhalten können und für die zusätzliche Angaben durch die Verwendung weiterer Makros innerhalb des Feldbefehlargumentes möglich sind, werden diese Befehle zum Aufteilen der Angaben bereitgestellt. Mit diesen wird es möglich, die durch \texttt{\&} getrennten Teile separat auszuwerten. Für diese Unterfangen wird der Befehl \texttt{\tud@multiple@@@@split} definiert. Dessen obligatorisches Argument ist dabei das Feld mit dem auszuwertenden Inhalt.

Dafür müssen für jedes so auszuwertende Feld zum einen zum Zeitpunkt der Ausgabe das entsprechende Makro \texttt{\tud@split(Feld)} und zum anderen eine Liste der auszuwertenden lokalen Angaben \texttt{\tud@split(Feld)@list} definiert sein. Momentan werden die beiden Felder...
\@author und – für die Klasse \texttt{tudscrposter} sowie das Paket Pakete \texttt{tudscrsupervisor} – \@contactperson zur Angabe zusätzlicher Informationen unterstützt.

Sollte ein Feld verwendet werden, welches initial eine Fehlermeldung enthält, so die vorhandene Warnung ausgegeben und danach das Feld als leer definiert.

Das Makro \texttt{tud@multiple@split} wird so definiert, dass der Befehl \texttt{\and} als Separator für die einzelnen Argumente dient. Mit den beiden freigestellten Argumenten kann das Makro zur eigentlichen Ausgabe aufgerufen werden, welches aus dem obligatorischen Argument \texttt{#1} konstruiert wird (\texttt{\tud@split(Feld)}). An dieses wird der jeweils aktuelle Autor im ersten Argument und die restlichen im zweiten Argument übergeben.

Das Feld wird mit \texttt{\and} terminiert, um der Definition von \texttt{tud@multiple@split} in jedem Fall zu entsprechen.

Der Befehl \texttt{tud@multiple@split} prüft zum Schluss, ob noch weitere Autoren angegeben sind. Sollte dies der Fall sein, so wird der Inhalt des zweiten Argumentes ausgeführt und \texttt{tud@multiple@split} ein weiteres Mal aufgerufen, um so sequentiell alle Autoren abzuarbeiten. Dafür muss \texttt{tud@multiple@split} innerhalb des verarbeitenden Makro (\texttt{\tud@split(Feld)}) aufgerufen werden.

Mit \texttt{tud@multiple@fields@store} und \texttt{tud@multiple@fields@restore} werden Hilfsmakros definiert, um einzelne Feldinhalte lokal ändern und nach der Verarbeitung auf den ursprünglichen Wert zurücksetzen zu können.

Nach der Sicherung der globalen Feldinhalte wird der übergebene Teilinhalt des zweiten Argumentes mit \texttt{\sbox\z@{(#2)}} in eine Box expandiert. Der Teilinhalt entspricht dabei dem
aktuellen Teil des Feldes vor dem nächsten \and. Damit werden die ggf. angegebenen lokalen Felder gesetzt, welche in \tud@split(Feld)@list aufgelistet sind ohne eine Ausgabe zu erzeugen.

\newcommand*{\tud@multiple@fields@store}[2]{% 
\def\@tempa##1{% 
  \ifcsdef{@##1}{\tud@cmd@store{@##1}}{% 
\letcs\@tempb{tud@split#1@list}% 
\let\and\relax% 
\expandafter\forcsvlist\expandafter\@tempa\expandafter{\@tempb}%%
\begingroup% 
\let\thanks\@gobble% 
\let\footnote\@gobble% 
\sbox\z@{#2}% 
\endgroup% 
}\}%
}

Nach dem Verarbeiten und der Ausgabe der lokalen Felder werden die zuvor bestehenden Feldwerte zurückgesetzt.

\newcommand*{\tud@multiple@fields@restore}[1]{% 
def\@tempa##1{% 
  \ifcsdef{@##1}{\tud@cmd@restore{@##1} \global\csletcs{@##1}{@##1}}{% 
}\}%
\letcs\@tempb{tud@split#1@list}% 
\let\and\relax% 
\expandafter\forcsvlist\expandafter\@tempa\expandafter{\@tempb}%%
\begingroup% 
\let\thanks\@gobble% 
\let\footnote\@gobble% 
\sbox\z@{#1}% 
\endgroup% 
}

Mit diesem Makro wird es möglich, die Inhalte bestimmter Felder aus einer definierter Liste \tud@split(Feld)@list auf Standardwerte zu setzen.

\newcommand*{\tud@multiple@fields@preset}[3]{% 
def\@tempa##1{% 
  \edef\@tempc{\csuse{##1}}% 
  \ifstr{#2}{*}{\csdef{@##1}{} '% 
  \begingroup% 
  \protected@expandtwoargs\in@{\@tempc}{\csuse{#1}}% 
  \ifin% 
  \ifcsdef{##1}{\@tempc{#2}}{% 
  \fi% 
  \endgroup% 
  \}%
  \}}% 
  \begingroup% 
  \letcs\@tempb{tud@split#1@list}% 
  \let\and\relax% 
  \expandafter\forcsvlist\expandafter\@tempa\expandafter{\@tempb}%%
  \begingroup% 
  \let\thanks\@gobble% 
  \let\footnote\@gobble% 
  \sbox\z@{#3}% 
  \endgroup% 
}\}%

File d: tudscr-fields.dtx
Auf der Titelseite sowie für die Aufgabenstellung (tudscrsupervisor) und den Seitenfuß von Postern (tudscrpster) wird die Angabe einer kollaborativen Autorenschaft ermöglicht, wo für jeden einzelnen Autor weitere Angaben (Matrikelnummer etc.) gemacht werden können. Hierfür werden die folgenden Makros bereitgestellt.

\begin{verbatim}
\newcommand*{\tud@split@author@do}[2]{}
\newcommand*{\tud@split@author@list}{authormore, course, discipline,%
matriculationyear, enrolmentyear,%
matriculationnumber, studentid, matriculationid,%
placeofbirth, birthplace, dateofbirth, birthday,%
}
\fi
\end{verbatim}

Die beiden Makros \tud@split@author@do und \tud@split@contactperson@do werden zunächst jeweils als Dummy initialisiert und an den entsprechenden Stellen umdefiniert. Dies betrifft den Titel, die Aufgabenstellung und den Aushang in den Umgebungen task und notice sowie den Seitenfuß bei Postern. Im Makro \tud@split@author@list werden alle Felder hinterlegt, die durch \tud@multiple@fields@... geprüft und ggf. initialisiert werden sollen.

Für die Angabe mehrerer Kontaktpersonen im Posterfuß (tudscrpster) oder auf einem Aushang (tudscrsupervisor) wird äquivalent verfahren.

\begin{verbatim}
\newcommand*{\tud@split@author@list}{authormore, course, discipline, office, telephone, telefax, emailaddress}
\end{verbatim}

Der Befehl \tud@split@contactperson@list für die Klasse tudscrpster.

\begin{verbatim}
\newcommand*{\tud@split@contactperson@do}[2]{}
\newcommand*{\tud@split@contactperson@list}{office, telephone, telefax, emailaddress}
\fi
\end{verbatim}

Damit ist der Teil für das Paket tudscrsupervisor abgeschlossen, wo bei der Verwendung von tudscrpster die vorhergenden Befehle nicht definiert werden, beendet.
8 Lokalisierung mithilfe sprachabhängiger Bezeichner

Das TUD-Script-Bundle ist für die deutsche und englische Sprache lokalisiert. Dies bedeutet, dass abhängig von der gewählten Sprache die entsprechenden Bezeichner gesetzt werden. Hierfür werden die Möglichkeiten von KOMA-Script in Form des Befehls \providecaptionname genutzt.

8.1 Definition der sprachabhängigen Bezeichner

\tud@locale@define


Diese Bezeichner existieren nur für die drei Hauptklassen.

File e: tudscr-locale.dtx
Diese Bezeichner stehen zusätzlich auch für \texttt{tudscrposter} zur Verfügung.

\begin{itemize}
\item \texttt{coursename} (Lok.)
\item \texttt{disciplinename} (Lok.)
\item \texttt{listingname} (Lok.)
\item \texttt{listlistingname} (Lok.)
\item \texttt{dissertationname} (Lok.)
\item \texttt{diplomathesisname} (Lok.)
\item \texttt{masterthesisname} (Lok.)
\item \texttt{bachelorthesisname} (Lok.)
\item \texttt{studentthesisname} (Lok.)
\item \texttt{studentresearchname} (Lok.)
\item \texttt{projectpapername} (Lok.)
\item \texttt{seminarpapername} (Lok.)
\item \texttt{termpapername} (Lok.)
\item \texttt{researchname} (Lok.)
\item \texttt{logname} (Lok.)
\item \texttt{internshipname} (Lok.)
\item \texttt{reportname} (Lok.)
\end{itemize}

\begin{itemize}
\item \texttt{authorname} (Lok.)
\item \texttt{contactname} (Lok.)
\item \texttt{contactpersonname} (Lok.)
\end{itemize}

\begin{itemize}
\item \texttt{taskname} (Lok.)
\item \texttt{tasktext} (Lok.)
\item \texttt{namesname} (Lok.)
\item \texttt{issuedatetext} (Lok.)
\item \texttt{duedatetext} (Lok.)
\item \texttt{chairmanname} (Lok.)
\item \texttt{focusname} (Lok.)
\item \texttt{objectivesname} (Lok.)
\item \texttt{evaluationname} (Lok.)
\item \texttt{evaluationtext} (Lok.)
\item \texttt{contentname} (Lok.)
\item \texttt{assessmentname} (Lok.)
\item \texttt{gradetext} (Lok.)
\item \texttt{noticename} (Lok.)
\end{itemize}

\begin{itemize}
\item \texttt{tud@general@name} (Lok.)
\item \texttt{tud@implementation@name} (Lok.)
\item \texttt{tud@changes@name} (Lok.)
\item \texttt{tud@todo@name} (Lok.)
\item \texttt{tud@environments@name} (Lok.)
\item \texttt{tud@option@name} (Lok.)
\item \texttt{tud@options@name} (Lok.)
\item \texttt{tud@layerpagestyle@name} (Lok.)
\item \texttt{tud@layerpagestyles@name} (Lok.)
\item \texttt{tud@layer@name} (Lok.)
\end{itemize}

Diesen Bezeichner werden von der Klasse \texttt{tudscrdoc} genutzt.

\begin{itemize}
\item \texttt{tud@general@name} (Lok.)
\item \texttt{tud@implementation@name} (Lok.)
\item \texttt{tud@changes@name} (Lok.)
\end{itemize}

\begin{itemize}
\item \texttt{tud@general@name} (Lok.)
\item \texttt{tud@implementation@name} (Lok.)
\item \texttt{tud@changes@name} (Lok.)
\end{itemize}

\texttt{File e: tudscr-locale.dtx}
8.2 Hilfsmakros für selektive Bezeichner

Einige Bezeichner verhalten sich je nach der Angabe für einzelne Felder selektiv, die zur Auswahl notwendigen Makros werden hier definiert.

\tud@ifin@and

Dieser Befehl prüft, ob innerhalb eines Felder, welches im ersten Argument angegeben werden muss, `\and` verwendet wurde. Ist dies der Fall, wird das zweite Argument ausgeführt, andernfalls das dritte.

\newcommand*{\tud@ifin@and}[1]{%  
  \begingroup  
  \let\and\relax  
  \protected@edef\@tempb{#1}  
  \def\@tempa##1\and##2\relax{%  
    \IfArgIsEmpty{##2}{%  
      \aftergroup\@secondoftwo  
    }{%  
      \aftergroup\@firstoftwo  
    }%  
  }%  
  \expandafter\@tempa\@tempb\and\relax  
  \endgroup  
}

8.3 Deutschsprachige Bezeichner

\tud@locale@german

Dieser Befehl dient zur Definition der deutschsprachigen Bezeichner. Dabei müssen als Argumente der Bezeichnerbefehl selbst sowie die dazugehörige Definition angegeben werden. Intern wird dabei `\providecaptionname` verwendet.
Hier erfolgt die eigentliche Definition der sprachabhängigen Bezeichner für die deutsche Sprache und ihre Dialekte.

Für die nachfolgenden Felder, für die es bedarfsweise einen Bezeichner für eine zweite Person gibt (\...othername), werden jeweils verschiedene Varianten definiert. Existiert in einem Feld nur eine Person, wird der Singular der Bezeichnung verwendet. Wurden mindestens zwei Personen angegeben (\and), so wird geprüft, ob der Bezeichner für die zusätzlichen Personen nicht leer ist. Ist dies der Fall, wird die alternative Form des Bezeichners der ersten Person verwendet, andernfalls wird der Bezeichner im Plural verwendet.

Es waren keine weiteren Personen an der geistigen Herstellung der vorliegenden Arbeit beteiligt.

\if\@supporter\else\end{quote} \noindent Weitere Personen waren an der geistigen Herstellung der vorliegenden Arbeit nicht beteiligt.

\if\@supporter\else\während der Anfertigung dieser Arbeit wurde ich nur von folgenden Personen unterstellt: \fi

Mir ist bekannt, dass die Nichteinhaltung dieser Erklärung zum nachträglichen Entzug des Hochschulabschlusses führen kann.


\if\@supporter\else\während der Anfertigung dieser Arbeit wurde ich nur von folgenden Personen unterstellt: \fi

Weitere Personen waren an der geistigen Herstellung der vorliegenden Arbeit nicht beteiligt.

\if\@supporter\else\während der Anfertigung dieser Arbeit wurde ich nur von folgenden Personen unterstellt: \fi

Mir ist bekannt, dass die Nichteinhaltung dieser Erklärung zum nachträglichen Entzug des Hochschulabschlusses führen kann.

Hier erfolgen für die Klasse \texttt{tudscrposter} sowie das Paket \texttt{tudscrsupervisor} weitere Definitionen.

Hier erfolgen für das Paket \texttt{tudscrsupervisor} weitere Definitionen.

Dies sind die Bezeichner für die Quelltextdokumentation.


Kursive Zahlen entsprechen der Seite, auf welcher der korrespondierende Eintrag beschrieben wird. Unterstrichene Zahlen verweisen auf die Codezeile der \fi Definition.

Alle weiteren Einträge sind Zeilennummern, wo der jeweilige Eintrag verwendet wird.

8.4 Englischsprachige Bezeichner

Dieser Befehl dient zur Definition der englischsprachigen Bezeichner. Dabei müssen als Argumente der Bezeichnerbefehl selbst sowie die dazugehörige Definition angegeben werden. Intern wird dabei \providecaptionname verwendet.

Hier erfolgt die eigentliche Definition der sprachabhängigen Bezeichner für die deutsche Sprache und ihre Dialetke.

\newcommand*{\tudlocaleenglish}{%
I hereby certify that I have authored this thesis independently and without undue assistance from third parties. No other than the resources and references indicated in this thesis have been used. I have marked both literal and accordingly adopted quotations as such. During the preparation of this thesis I was only supported by the following persons:

There were no additional persons involved in the intellectual preparation of the present thesis. I am aware that violations of this declaration may lead to subsequent withdrawal of the degree.

This contains confidential data, disclosed by . Publications, duplications and inspections—even in part—are prohibited without explicit permission, as well as publications about the content of this thesis. This thesis may only be made accessible to the supervisor at Technische Universität "at Dresden, the reviewers and also the members of the examination board.
Hier erfolgen für die Klasse \texttt{tudscrposter} sowie das Paket \texttt{tudscrsupervisor} weitere Definitionen.

Hier erfolgen für das Paket \texttt{tudscrsupervisor} weitere Definitionen.

Dies sind die Bezeichner für die Quelltextdokumentation.
8.5 Kompatibilität der Bezeichner mit verschiedenen Pakete

8.5.1 Unterstützung des Paketes listings

Die Bezeichner des Paketes werden auf die bereits definierten gesetzt.

\AfterPackage{listings}{%
  \renewcommand*{\lstlistingname}{\listingname}%
  \renewcommand*{\lstlistlistingname}{\listlistingname}%
}%

8.5.2 Unterstützung des Paketes mathswap

Wird das Paket mathswap verwendet, werden die Ersetzungen für Punkt und Komma im Mathematikmodus sprachspezifisch angepasst.

\AfterPackage{mathswap}{%
  \tud@locale@german{\@commaswap}{,}%
  \tud@locale@german{\@dotswap}{.}%
  \tud@locale@english{\@commaswap}{\,}%
  \tud@locale@english{\@dotswap}{.}%
}%
9 Der Satzspiegel für TUD-Script


9.1 Kompatibilität der Satzspiegeleinstellungen mit weiteren Paketen

9.1.1 Unterstützung von Schnittmarken mit dem Paket crop

Insbesondere für das Erstelen von Postern sollen häufig Schnittmarken auf dem ausgegebenen Papierbogen erscheinen. Dies lässt sich entweder mit den Mitteln des Paketes `geometry` oder aber durch das Paket `crop` realisieren. Für letzteres ist dafür ein kleiner Patch notwendig.

```
\if@tud@x@crop@center
   \CROP@center
\fi
```

Das Paket `crop` bietet die Option, den Satzspiegel auf dem Papierbogen zu zentrieren. Diese Einstellung geht beim Laden eines neuen Satzspiegels mit `\loadgeometry` verloren. Hiermit wird das Problem behoben.

```
\newif\if@tud@x@crop@center
\AfterPackage{\textfamilydefault}{%}
   \CheckCommand*{\CROP@center}{%}
      \voffset\stockheight
      \advance\voffset-\paperheight
      \voffset.5\voffset
      \hoffset\stockwidth
      \advance\hoffset-\paperwidth
      \hoffset.5\hoffset
   }%
\pretocmd{\CROP@center}{%}
   \@tud@x@crop@centertrue%
}\if@tud@x@crop@center
\fi
```

9.1.2 Unterstützung der Klasse standalone


```
\if@tud@x@standalone@crop
   Mit diesem Makro kann die Verwendung der Option `crop` mit der Klasse `standalone` geprüft werden. Das erste Argument wird ausgeführt, wenn die Klasse geladen und die Option aktiv
```
ist. Das zweite Argument wird ausgeführt, wenn entweder die Option `crop` deaktiviert ist oder die Klasse `standalone` gar nicht geladen wurde.

\newif{\if@tud@x@standalone@crop}
\AfterClass*{standalone}{\let\if@tud@x@standalone@crop\ifs@crop}

9.2 Definition der Größen und Maße in abhängig vom Papierformat

Die TUD-Script-Klassen sollen neben den vom Corporate Design vorgegebenen Seitenrändern weitere Satzspiegel ermöglichen und u. a. auch die Nutzung des Paketes `typearea` erlauben. Damit ein einheitliches Verfahren zur Wahl bzw. Umschaltung des Satzspiegels genutzt werden kann, wird momentan das Paket `geometry` genutzt. Dabei werden entweder die gewünschten Maße der Seitenränder direkt gewählt oder aber die mit `typearea` berechneten Maße an `geometry` durchgereicht.

\newlength{\tud@len@widemargin}
\newlength{\tud@len@slimmargin}
\newlength{\tud@len@both}

Die Seitenränder für links und rechts sowie deren Summe, welche später für weitere Satzspiegel- einstellungen dazu verwendet wird, diese anteilig zu verteilen.

\newlength{\tud@len@topmargin}
\newlength{\tud@len@barheight}
\newlength{\tud@len@headsep}
\newlength{\tud@len@footsep}

Es folgen die speziellen Maße für die Seiten mit dem TUD-Kopf, also die Höhe oberhalb des Querbalkens, die Höhe des Querbalkens selbst sowie der vertikale (Mindest-)Abstand zwischen Querbalken und Textkörper sowie der Fußzeile.

\newlength{\tud@len@logox}
\newlength{\tud@len@logoy}
\newlength{\tud@len@logowidth}

Außerdem müssen Abmessungen und Position des TUD-Logos definiert werden, genauer der horizontale (Soll-)Abstand zwischen linkem Seitenrand und Logo, der vertikale (Soll-)Abstand zwischen oberem Seitenrand und Logo sowie die Breite und Höhe\(^5\) des Logos.

\newlength{\tud@len@line}
\newlength{\tud@len@heavyline}
\newlength{\tud@len@thinline}

Die Linienstärke der Outline im Kopf ist für ein monochromes Layout schmaler als im Layout mit farbigem Hintergrund.\(^6\) Da im Dokument beide Varianten vorkommen können, wird die Breite `\tud@len@line` durch den Seitenstil situativ entweder auf den Wert von `\tud@len@thinline` oder aber `\tud@len@heavyline` gesetzt.

\newlength{\tud@len@line}
\newlength{\tud@len@heavyline}
\newlength{\tud@len@thinline}

\(^5\)Proportionen sind durch die Grafik vorgegeben
\(^6\)monochrom `\tud@len@thinline`, koloriert `\tud@len@heavyline`
Die Längen für den vertikalen Standardversatz der Überschriften sowie die Differenz der Kopfhöhen von normalem und TUD-Kopf-Satzspiegel. Die Differenz zwischen den unterschiedlichen Kopfhöhen der Satzspiegelvarianten wird durch \texttt{tud@cdgeometry@process} berechnet und dabei in \texttt{tud@len@areadiff} gespeichert. Dabei muss unterschieden werden, ob der Satzspiegel des Corporate Designs oder der durch \texttt{typearea} berechnete respektive durch den Anwender mit \texttt{geometry} vorgegebene genutzt wird. Davon abhängig wird zum Ausgleich die Länge \texttt{tud@len@areavskip} definiert.

\begin{verbatim}
30 \newlength{tud@len@areaheadvskip}
31 \newlength{tud@len@areadiff}
32 \newlength{tud@len@areavskip}
\end{verbatim}

Zu guter Letzt noch die Länge \texttt{tud@len@ddcdiff} für den Höhenausgleich des Seitenfußes, welcher bei der Verwendung des DRESDEN-concept-Logos in der Fußzeile bei Version v2.02 für den Satzspiegel notwendig war. Dieser etwas verschrobene Satzspiegel wird im Kompatibilitätsmodus weiterhin angeboten.

\begin{verbatim}
33 \if@v@lower{2.03}{\newlength{tud@len@ddcdiff}}{}
\end{verbatim}

Hiermit werden in Abhängigkeit der gewählten Papiergröße die Maße für die Größe der Kopfzeile, die Position des Logos, die Seitenränder sowie die Schriftgröße in der Kopfzeile definiert. Aufgerufen wird der Befehl durch \texttt{tud@cdgeometry@process}. Dabei fungiert \texttt{tud@len@widemargin} als Bezugslänge. Die maßgebliche Papiergröße ist DIN A4 bzw. DIN B4. Die Längen für alle anderen Papiermaße unterscheiden sich um den Faktor $\sqrt{2}$, wobei das Format DIN A5 nach den Vorgaben des Corporate Designs übersprungen wird.

Um die Überschriften und den Satzspiegel auch in diesem Fall vertikal richtig zu positionieren, wird der Schalter \texttt{\if@tud@cdgeometry@adjust} benötigt. Mit der Einstellung \texttt{cdgeometry=adapted} wird auch das Format DIN A5 mit den skalierten Seitenrändern erstellt.

\begin{verbatim}
34 \newif{\if@tud@cdgeometry@adjust}
35 \newlength{tud@len@layoutheight}
36 \newlength{tud@len@layoutwidth}
37 \newlength{tud@len@layouthoffset}
38 \newlength{tud@len@layoutvoffset}
39 \newcommand*{tud@cdgeometry@set}{%
40 \ifGm@layout%
41 \setlength{tud@len@layoutheight}{\Gm@layoutheight}%
42 \setlength{tud@len@layoutwidth}{\Gm@layoutwidth}%
43 \else%
44 \setlength{tud@len@layoutheight}{\paperheight}%
45 \setlength{tud@len@layoutwidth}{\paperwidth}%
46 \fi%
47 \setlength{tud@len@layouthoffset}{\Gm@layouthoffset}%
48 \setlength{tud@len@layoutvoffset}{\Gm@layoutvoffset}%
49 \tud@if@v@lower{2.05}{%
50 \ifdim{tud@len@layoutheight}<\dimexpr105mm+1.5mm\relax%
51 \def{\tempa}{0}%
52 \else\ifdim{tud@len@layoutheight}<\dimexpr148mm+1.5mm\relax%
53 \def{\tempa}{1}%
54 \else\ifdim{tud@len@layoutheight}<\dimexpr229mm+2mm\relax%
55 \def{\tempa}{2}%
56 \else\ifdim{tud@len@layoutheight}<\dimexpr297mm+2mm\relax%
57 \def{\tempa}{3}%
58 \else\ifdim{tud@len@layoutheight}<\dimexpr420mm+2mm\relax%
59 \def{\tempa}{4}%
\end{verbatim}

File: tudscr-area.dtx
In Abhängigkeit von der gefundenen Gestaltungshöhe wird nun der Satzspiegel definiert. Dabei wird geprüft, ob dieser im entweder innerhalb des durch das Corporate Design vorgegebenen Rasters erzeugt oder anhand der Gestaltungshöhe berechnet werden soll.

Eine Berechnung des Satzspiegels findet in jedem Fall außerhalb des Rasters statt. Ansonsten wird DIN A4 als Referenzformat ausgewählt.

Bei Formaten, welche kleiner als das Raster sind, wird die Gestaltungshöhe für DIN A6 als Referenz genommen, um den Satzspiegel zu berechnen.
Für Formate der Klasse 5 gibt es zwei Varianten. Entweder, es wird sich an das Raster des Corporate Designs geahtlen oder es wird ein eigener Satzspiegel für das Papierformat definiert. Wird das Raster gehalten, muss bei den Überschriften etwas getrickst werden, weshalb der Schalter \if@tud@cdgeometry@adjust gesetzt wird.

\ifnum\tud@cdgeometry@calc@num=\z@\relax% restricted
\tud@cdgeometry@adjusttrue%
\setlength\tud@len@widemargin{30mm}%
\setlength\tud@len@heavyline{1pt}%
\setlength\tud@len@thinline{0.5pt}%
\renewcommand*\tud@head@fontsize{9}%
\else%
\setlength\tud@len@widemargin{21.213203mm}%
\setlength\tud@len@heavyline{0.707107pt}%
\setlength\tud@len@thinline{0.353553pt}%
\renewcommand*\tud@head@fontsize{6.363961}%
\fi%
\orX <=A4/B4
\setlength\tud@len@widemargin{60mm}%
\setlength\tud@len@heavyline{2pt}%
\setlength\tud@len@thinline{1pt}%
\renewcommand*\tud@head@fontsize{12}%
\orX <=A3/B3
\setlength\tud@len@widemargin{42.426407mm}%
\setlength\tud@len@heavyline{1.414214pt}%
\setlength\tud@len@thinline{0.707107pt}%
\renewcommand*\tud@head@fontsize{12.727922}%
\orX <=A2/B2
\setlength\tud@len@widemargin{84.852814mm}%
\setlength\tud@len@heavyline{2.828427pt}%
\setlength\tud@len@thinline{1.414214pt}%
\renewcommand*\tud@head@fontsize{18}%
\orX <=A1/B1
\setlength\tud@len@widemargin{120mm}%
\setlength\tud@len@heavyline{4pt}%
\setlength\tud@len@thinline{2pt}%
Wenn die Berechnung aktiviert, dann erfolgt diese anhand der Referenz im Format DIN A4.

Zum Schluss werden die restlichen Längen aus der Referenzlänge berechnet und das Erstellen der Kopfzeile aufgrund der möglichen Änderung der Schriftgröße forciert.

Wird die Gestaltungshöhe DIN A5 im Raster verwendet, werden die Abstände zwischen Kopf- und Fußzeile angepasst.
9.3 Optionen für den Satzspiegel

Für das Corporate Design sind standardmäßig feste Seitenränder vorgegeben. Diese sind jedoch leider nur aus gestalterischen Motiven und ohne die Beachtung typographischer Belange gewählt und festgelegt worden. Beispielweise ist ein doppelseitiger Satz einer Abschlussarbeit im Corporate Design nur mit einem grauenhaften Satzspiegel möglich. Um dem Anwender einen gewissen Spielraum zu geben, wird alternativ zum asymmetrischen (geometry=on) ein symmetrischer Satzspiegel bereitgestellt (geometry=normal), welcher bei einseitigem Satz zentriert ist und bei zweiseitigem Satz den inneren Rand kleiner setzt als den äußeren. Des Weiteren kann die Satzspiegelberechnung auch direkt durch das Paket typearea aus dem KOMA-Script-Paket erfolgen (geometry=no). Mit \tud@cdgeometry@process werden alle getroffenen Satzspiegeloptionen umgesetzt und nachfolgend definiert.
Diese Option dient dazu, die Höhe des Fußes anzupassen. Dies ist jedoch nur möglich, wenn der Satzspiegel des Corporate Designs verwendet wird. Kommt \texttt{typearea} zum Einsatz, ist die Option wirkungslos. Da für die Länge ggf. auch Werte angegeben werden können, die abhängig von der Schriftgröße sind, wird die Ausführung dieser Option verzögert.

Wird entweder das Paket \texttt{crop} oder aber die Option \texttt{layout} des Paketes \texttt{geometry} für Schnittmarken verwendet, werden alle farbigen Elemente des Layouts dahingehend vergrößert, dass beim Zuschneiden des Papierbogens \textit{in die Farbe} geschnitten wird. Verantwortlich hierfür ist das Makro \texttt{\tud@dim@bleedmargin}, welches natürlich auch vom Anwender mit der Option \texttt{bleedmargin} beliebig gesetzt werden kann. Als Standardwert werden 5\% der größten Randbreite gesetzt.

Negative Werte sind nicht möglich, diese werden automatisch korrigiert.

Sollte einer der beiden KOMA-Script-Optionen im Dokument nach der Präambel verwendet werden, erfolgt die Anpassung des Satzspiegels.
Sollte das Paket \texttt{silence} vorhanden sein wird es geladen, um die vom Paket \texttt{typearea} erzeugten Warnungen beim Laden einer TUD-Script-Klasse zu unterdrücken, da im Normalfall das Paket gar nicht für die Berechnung des Satzspiegels zum Einsatz kommt. Sollte dies vom Anwender so eingestellt werden, wird vor dem Beginn des Dokumentes \texttt{\recalctypearea} aufgerufen und die entsprechenden Warnungen werden etwas später und in Abhängigkeit der verwendeten Schriftart erzeugt.

\begin{verbatim}
\newcommand*{\tud@x@ta@warnings@off}{}
\newcommand*{\tud@x@ta@warnings@on}{
  \IfFileExists{silence.sty}{%\RequirePackage{silence}[2012/07/02]%
    \WarningFilter[typearea]{typearea}{DIV for}%
    \WarningFilter[typearea]{typearea}{Bad type area settings!}%
    \WarningFilter[typearea]{typearea}{Maybe no optimal type area settings!}%
    \WarningFilter[typearea]{typearea}{Very low DIV value!}%
    \BeforePackage{typearea}{\ActivateWarningFilters[typearea]}%
    \AfterPackage{typearea}{\DeactivateWarningFilters[typearea]}%
  }{%\renewcommand*{\tud@x@ta@warnings@off}{\ActivateWarningFilters[typearea]}
    \renewcommand*{\tud@x@ta@warnings@on}{\DeactivateWarningFilters[typearea]}
}
\end{verbatim}

\section{Realisierung der Satzspiegelleinstellungen}

Das Paket \texttt{geometry} erlaubt eine genaue Vorgabe des Satzspiegels und der Seitenränder, wie es vom Corporate Design der Technischen Universität Dresden verlangt wird.

\begin{verbatim}
\newcommand*{\tud@cdgeometry@process}{}
\newcommand*{\tud@cdgeometry@@process}{
  \if@tud@x@standalone@crop
    \PreventPackageFromLoading{geometry}%
  \else
    \RequirePackage{geometry}[2010/09/12]%
  \fi
  \AfterPackage*{geometry}{%\CheckCommand*{\Gm@initnewgm}{%
    \Gm@passfalse
    \Gm@swap@papersizefalse
    \Gm@dimlist={}
    \Gm@hbodyfalse
    \Gm@vbodyfalse
    \Gm@heightroundedfalse
    \Gm@includeheadfalse
    \Gm@includefootfalse
    \Gm@includempfalse
    \let\Gm@width\@undefined
    \let\Gm@height\@undefined
    \let\Gm@textwidth\@undefined
    \let\Gm@textheight\@undefined
    \let\Gm@lines\@undefined
    \let\Gm@hscale\@undefined
    \let\Gm@vscale\@undefined
    \let\Gm@hmarginratio\@undefined
    \let\Gm@vmarginratio\@undefined
  \}}
\end{verbatim}
\leq \textwidth
\setlength{\@colht}{\textheight}
\setlength{\@colroom}{\textheight}
\setlength{\vsize}{\textheight}
\setlength{\columnwidth}{\textwidth}
\if@twocolumn
\advance\columnwidth-\columnsep
\divide\columnwidth\tw@
\@firstcolumntrue
\fi
\setlength{\hsize}{\columnwidth}
\setlength{\linewidth}{\hsize}
\ifGm@includemp
\addtolength{\oddsidemargin}{-\Gm@wd@mp}
\addtolength{\evensidemargin}{\Gm@odd@mp}
\fi
\if@mparswitch
\setlength{\evensidemargin}{\Gm@rmargin}
\addtolength{\evensidemargin}{-1\Gm@truedimen}\in}
\ifGm@includemp
\advance\evensidemargin{\Gm@even@mp}
\fi
\else
\evensidemargin\oddsidemargin
\fi
\evensidemargin\oddsidemargin
\ifGm@includehead
\addtolength{\topmargin}{-\headheight}
\addtolength{\topmargin}{-\headsep}
\else
\addtolength{\topmargin}{-\headheight}
\addtolength{\topmargin}{-\headsep}
\fi
\else
\addtolength{\topmargin}{-\headheight}
\addtolength{\topmargin}{-\headsep}
\fi
\ifGm@includefoot

File f: tudscr-area.dtx
Für einen benutzerdefinierten Satzspiegel ist es notwendig, die durch den Anwender in der Präambel getätigten Einstellungen zu sichern und am Ende der Präambel für die unterschiedlichen, durch die Klasse benötigten Satzspiegel der einzelnen Seitenstile umzusetzen. Dabei werden auch etwaige Optionen von KOMA-Script beachtet. Hierfür werden die originalen Befehle \geometry und \newgeometry angepasst, weshalb diese vorher gesichert werden.

Mit dem Befehl \tud@setgeometry wird abhängig davon, ob dieser in der Präambel oder im Dokument von \tud@cdgeometry@process verwendet wird, entweder \geometry oder \newgeometry aufgerufen. Mit dem Schalter \if@tud@x@geometry@enabled wird im Fall, dass ein der Satzspiegel durch den Anwender eingestellt wurde, das rekursive Aufrufen von \geometry bzw. \newgeometry und einer daraus resultierenden Endlosschleife unterbunden.

Unabhängig von der gewählten Option cdgeometry für den Satzspiegel sollen die von geometry bereitgestellten Möglichkeiten für den Papierbogen nutzbar bleiben.

Um etwaige Änderungen am Papierformat innerhalb des Dokumentes beachten zu können, werden die aktuellen Maße für das Papierformat an \Gm@restore@org angehängt.

Zu Beginn des Dokumentes werden mit \Gm@initnewgm vor jedem Satzspiegel die zuvor gemachten Einstellungen zurückgesetzt.
Beim Speichern eines Satzspiegels sichert geometry tatsächlich nicht alle notwendigen Einstellungen und Befehle. Deshalb wird etwas nachgeholfen.

\newcommand*{\tud@savegeometry}[1]{% 
def\Gm@restore{}% 
\savegeometry{#1}% 
\csepreto{Gm@restore@@#1}{% 
\etex@unexpanded{\def\Gm@lmargin}{\Gm@lmargin}% 
\etex@unexpanded{\def\Gm@rmargin}{\Gm@rmargin}% 
\ifGm@includemp% 
\noexpand\Gm@includemptrue% 
\else% 
\noexpand\Gm@includempfalse% 
\fi% 
}%
\newcommand*{\tud@loadgeometry}[1]{% 
\@tud@x@geometry@enabledfalse% 
\loadgeometry{#1}%
@currsize%
@if@tud@x@crop@center{\CROP@center% 
}@tud@x@geometry@enabledtrue%
}\geometry
\newgeometry
\tud@geometry
paper (Param.)
papername (Param.)
paperwidth (Param.)
paperheight (Param.)
papersize (Param.)
layout (Param.)
layoutname (Param.)
layoutwidth (Param.)
layoutheight (Param.)
layoutsize (Param.)
bleedmargin (Param.)
\tud@area@def
\tud@area@set
\tud@x@geometry@init
\tud@x@geometry@paper
\tud@x@geometry@layout
\tud@x@geometry@parameter

Die nächsten beiden Makros müssen nach der Satzspiegeländerung des ausgeführt werden, um den Durchschuss und eine etwaige Zentrierung durch das Paket crop wiederherzustellen.

\@currsize%
@if@tud@x@crop@center{\CROP@center% 
}@tud@x@geometry@enabledtrue%

Das folgende Konstrukt ist aus der Idee entstanden, die Parameter paper sowie layout des Paketes geometry auf die passenden Optionen von KOMA-Script abbilden zu können und insbesondere Schnittmarken für alle Seitenstile bereitzustellen.

\TUD@parameter@family{geometry}{% 
\TUD@parameter@def{paper}{\tud@area@def{paper}{#1}}% 
\TUD@parameter@let{papername}{paper}%
\TUD@parameter@def{paperwidth}{\tud@area@def{paper}{#1:	he\paperheight}}% 
\TUD@parameter@def{paperheight}{\tud@area@def{paper}{\the\paperwidth:#1}}% 
\TUD@parameter@def{papersize}{% 
\def\@tempa##1,##2,##3\@nil{% 
\IfArgIsEmpty{##2}{% 
\tud@area@def{paper}{##1:##1}% 
}\else% 
\tud@area@def{paper}{##1:##2}% 
\fi% 
}% 
\@tempa#1,,\@nil% 
}\TUD@parameter@def{layout}{\tud@area@def{layout}{#1}}% 
\TUD@parameter@let{layoutname}{layout}%
\TUD@parameter@def{layoutwidth}{% 
\tud@area@def{layout}{#1:\the\tud@len\layoutheight}}% 
\TUD@parameter@def{layoutheight}{% 
\tud@area@def{paper}{##1:\##2}\@nil{% 
\IfArgIsEmpty{##2}{% 
\tud@area@def{paper}{##1:##1}\}% 
}\else% 
\tud@area@def{paper}{##1:##2}\}% 
\fi% 
}% 
\@tempa##1,,\@nil% 
}% 
\TUD@parameter@def{layout}{% 
\tud@area@def{layout}{#1}\}%

File: tudscr-area.dtx
 Wenig Kommentar dazu, weil eigentlich schon wieder obsolet! Die möglichen Werte für \paper und \layout werden abgefangen und für die beiden Befehle \geometry und \newgeometry aufgearbeitet. Alle anderen werden einfach an geometry durchgereicht. Dies geschieht über die jeweiligen Hilfsmakros \tud@x@geometry@... für \paper, \layout sowie parameter für alles Übrige.
Der Befehl \td@area@set bewerkstelligt die korrekte Abarbeitung der zuvor gesammelten Schlüssel.
Bei der redefinition von `\geometry` und `\newgeometry` kommt intern `\tud@geometry` zum Einsatz, womit kleinere Feinheiten zwischen den beiden Befehlen unterschieden werden können.

```latex
\newcommand*{\tud@x@geometry@init}{}
\newcommand*{\tud@x@geometry@paper}{}
\newcommand*{\tud@x@geometry@layout}{}
\newcommand*{\tud@x@geometry@parameter}{}
\newcommand*{\tud@geometry}[2]{%\if\Gm@layout%
\setlength{\tud@len@layoutheight}{\Gm@layoutheight}%
\setlength{\tud@len@layoutwidth}{\Gm@layoutwidth}%
\else%
\setlength{\tud@len@layoutheight}{\paperheight}%
\setlength{\tud@len@layoutwidth}{\paperwidth}%
\fi%
\let{\tud@x@geometry@paper}@empty%
\let{\tud@x@geometry@layout}@empty%
\let{\tud@x@geometry@parameter}@empty%
\TUD@parameter@set{geometry}{#1}%
\tud@area@set{paper}%
\ifstr{#2}{newgeometry}{%\eappto{\Gm@restore@org}{%\paperwidth=\the\paperwidth\relax%\paperheight=\the\paperheight\relax%}}%\ifx{\tud@x@geometry@layout}@empty%\begingroup%\setlength{\paperwidth}{\Gm@layoutwidth}%\setlength{\paperheight}{\Gm@layoutheight}%\tud@area@set{layout}%
\edef{\tud@reserved}{%\endgroup%\etex@unexpanded{%\epreto{\tud@x@geometry@parameter}}{%\paperwidth=\the\paperwidth,paperheight=\the\paperheight,}%}}%\fi\endgroup%\etex@unexpanded{%\epreto{\tud@x@geometry@parameter}}{%}
\if\Gm@layout%\setlength{\tud@len@layoutheight}{\Gm@layoutheight}%\setlength{\tud@len@layoutwidth}{\Gm@layoutwidth}%\else%\setlength{\tud@len@layoutheight}{\paperheight}%\setlength{\tud@len@layoutwidth}{\paperwidth}%\fi%\let{\tud@x@geometry@paper}@empty%\let{\tud@x@geometry@layout}@empty%\let{\tud@x@geometry@parameter}@empty%
\TUD@parameter@set{geometry}{#1}%
\tud@area@set{paper}%
\ifstr{#2}{newgeometry}{%\eappto{\Gm@restore@org}{%\paperwidth=\the\paperwidth\relax%\paperheight=\the\paperheight\relax%}}%\ifx{\tud@x@geometry@layout}@empty%\begingroup%\setlength{\paperwidth}{\Gm@layoutwidth}%\setlength{\paperheight}{\Gm@layoutheight}%\tud@area@set{layout}%
\edef{\tud@reserved}{%\endgroup%\etex@unexpanded{%\epreto{\tud@x@geometry@parameter}}{%\paperwidth=\the\paperwidth,paperheight=\the\paperheight,}%}}%\fi\endgroup%\etex@unexpanded{%\epreto{\tud@x@geometry@parameter}}{%}
```

File: tudscr-area.dtx

89
Sollte im Dokument durch den Anwender der Satzspiegel manuell geändert werden, muss darauf reagiert und die unterschiedlichen Satzspiegel der Klasse angepasst werden. Hierfür wird am Ende von \Gm@changelayout ein Patch eingehängt.

Bei der Verwendung von \restoregeometry wird der am Ende der Präambel der passend zur entsprechende Einstellung von cdgeometry gesicherte Satzspiegel geladen. Über das optinale Argument kann der gewünschte Wert für die Option cdgeometry angegeben werden.

Dies sind Hilfsmakros für die Definition der unterschiedlichen Seitenlayouts (horizontale und vertikale Ränder). Es handelt sich dabei um die Seitenränder für den normalen Textbereich sowie die angepassten vertikalen Einstellungen für die Seitenstilvarianten mit TUD-Kopf. Im Kompatibilitätsmodus für die Version v2.02 gibt es außerdem einen separaten Satzspiegel für den DRESDEN-concept-Fuß. Gesetzt werden diese in Abhängigkeit von der Option cdgeometry im Makro \tud@cdgeometry@@@process bzw. \tud@cdgeometry@@process.
Nur falls \texttt{typearea} zum Einsatz kommt, wird am Ende der Präambel die Satzspiegelberechnung mit \texttt{recalctypearea} angestoßen, ansonsten wird bloß \texttt{tud@AfterChangingArea} ausgeführt.

Das Makro \texttt{\tud@cdgeometry@@process} setzt die Option für den gewünschten Satzspiegel um, sowohl für die Seitenränder als auch zur Einberechnung der Kopf- und/oder Fußzeile. Um mehrere Satzspiegel verwenden zu können – was für die unterschiedlichen Höhen für Kopf- und Fußzeile nötig ist – wird das Paket \texttt{geometry} verwendet. Soll \texttt{typearea} zur Satzspiegelerestellung genutzt werden, so werden die damit berechneten Werte an \texttt{geometry} weitergereicht. Es werden drei Layouts erstellt: normaler Satzspiegel, nur TUD-Kopf sowie TUD-Kopf und -Fuß und mit \texttt{\savegeometry{⟨Stil⟩}} gesichert. Damit kann innerhalb des Dokumentes mit dem Befehl \texttt{\loadgeometry{⟨Stil⟩}} oberer sowie ggf. untere Seitenrand geändert werden.

Die Maßvorgaben werden entsprechend der Seitengröße gesetzt.

Für den Fall, dass \texttt{typearea} die Satzspiegelberechnung übernimmt oder dieser durch den Benutzerdefiniert wurde, werden die Ergebnisse aus der Berechnung respektive die aktiven Einstellungen direkt an \texttt{geometry} weitergereicht. Dazu werden die berechneten Werte in die entsprechenden Hilfsmakros für die Erstellung der \texttt{geometry}-Satzspiegel übergeben.

Anschließend erfolgen die horizontalen und vertikalen Randeinstellungen. Zunächst für \texttt{typearea}. Hierfür ist eine Sonderbehandlung notwendig, falls mit \texttt{\geometry} eine Größe für den Druckbereich/das Layout angegeben wurde. Dafür wird das Papierformat temporär auf die Größe des angegebenen Layouts geändert und der Satzspiegel neu berechnet. Damit diese Berechnung jedoch nicht umgesetzt wird, wird \texttt{\activateareas} unschädlich gemacht.
Nachdem der Satzspiegel im Zweifelsfall neu berechnet wurde, werden die von typearea berechneten Seitenränder an geometry weitergereicht.

\edef\tud@cdgeometry@hmargin{%
  left=\the\dimexpr\oddsidemargin+1in-\ta@bcor\relax,%
  textwidth=\the\textwidth,%
  marginparwidth=\the\marginparwidth,%
  marginparsep=\the\marginparsep,%
  reversemarginpar=true,%
  bindingoffset=\the\ta@bcor%
}%
\edef\tud@cdgeometry@vmargin{%
  includehead=true,%
  top=\the\dimexpr\topmargin+1in\relax,%
  headheight=\the\headheight,%
  headsep=\the\headsep,%
  textheight=\the\textheight,%
  footskip=\the\footskip%
}%

Und nun für den benutzerdefinierten Satzspiegel.

\edef\tud@cdgeometry@hmargin{%
  left=\Gm@lmargin,%
  right=\Gm@rmargin,%
  includemp=true,%
  bindingoffset=\the\ta@bcor%
}%

File f: tudscr-area.dtx
Es wird die Höhendifferenz zwischen TUD-Kopf und Standardkopfzeile für den benutzer-definierten bzw. typearea-Satzspiegel berechnet. Für den Satzspiegel mit TUD-Kopf muss unterschieden werden, ob der Abstand zwischen Kopf und Textbereich vergrößert wurde. Ist dies der Fall, wird die Höhe des Textbereiches über \@tempdima entsprechend verkleinert. Sollte der benutzerdefinierten bzw. typearea-Satzspiegel noch unterhalb des TUD-Kopfes liegen, wird der Abstand zum Kopf vergrößert.

Für den Kompatibilitätsmodus der Version v2.02 gibt es für den DRESDEN-concept-Fuß einen separaten Satzspiegel.
Sollte die Option extrabottommargin verwendet worden sein, wird eine Warnung ausgegeben, dass diese für den typearea-Satzspiegel wirkungslos ist.

Hier erfolgt die Definition der Hilfsmakros für das CD-konforme asymmetrische bzw. an das Corporate Design angelehnte Layout für einseitigen und zweiseitigen Satz. Zuerst werden die verschiedenen unterschiedlichen horizontalen Ränder für die unterschiedlichen Optionen definiert. Dies betrifft im einzelnen den äußeren Seitenrand (\@tempdima), die Textbreite (\@tempdimb) sowie die Breite der Randnotizen (\@tempdimc). Danach kommt der Gleichanteil.

Die zweite Variante ist eigentlich nicht konform mit dem Corporate Design. Sie ist sowohl im einseitigen als auch im zweizeitigen Satz symmetrisch.

Eigentlich ist auch die dritte Variante nach dem Corporate Design nicht zulässig. Sie ist im einseitigen Satz symmetrisch, im zweiteiligen Satz wird die innere Seite schmaler gesetzt als die äußere, wobei hier auf das in den Seitenrand ragende TUD-Logo geachtet werden muss.
Die *Berechnung* der Textbreite ist für alle Varianten identisch. Danach erfolgt die Zuweisung zum Makro.

Für alle drei Varianten der vertikale Gleichanteil. Der Satzspiegel der Version v2.02 wird aus Gründen der Kompatibilität weiterhin vorgehalten.
Es wird die Differenz der Höhen zwischen TUD-Kopf und Standardkopfzeile für den Satzspiegel des Corporate Designs berechnet.

Für die Kompatibilitätsvariante wieder mal die Extrawurst.

Wird die Fußzeile zum Satzspiegel gerechnet, ist für Seiten mit DRESDEN-concept-Fuß etwas Handarbeit notwendig, damit dieser nicht allzu weit nach oben ragt.

Es wird die Differenz der Höhen zwischen TUD-Kopf und Standardkopfzeile für den Satzspiegel des Corporate Designs berechnet. Außerdem wird die standardmäßige vertikale Verschiebung der Überschriften festgelegt, wobei hier insbesondere auf die Gestaltungshöhe DIN A5 geachtet werden muss.
Für den Satzspiegel der Version v2.02 wird außerdem die Länge für den höheren Seitenfuß gesetzt.

980 \ifnum\v@lower>2.03\relax
981 \ifdim\v@ddcdiff<\z@ \relax\setlength\v@ddcdiff{\z@} \fi
982 \global\v@ddcdiff=\v@ddcdiff
983 \}

Es wird \tud@cdgeometry@@@process aufgerufen, was die zuvor definierten Hilfsmakros nutzt, um alle benötigten Seitenlayouts optionsabhängig zu erstellen.

984 \tud@cdgeometry@@@process
985 \}

Der Befehl \tud@cdgeometry@@@process setzt mit den zuvor definierten Hilfsmakros die geometry-Optionen für sowohl die seitlichen als auch die oberen sowie unteren Seitenränder und inkludiert ggf. Kopf- und Fußzeile in den Satzspiegel. Die unterschiedlichen Layouts sind dabei tudareaddc für TUD-Kopf und DRESDEN-concept-Fuß, tudarea für den alleinigen TUD-Kopf sowie stdarea für den normalen bzw. mit typearea berechneten Satzspiegel.

986 \newcommand*\tud@cdgeometry@@@process{%

Für den Satzspiegel der Version v2.02 wird als erstes der Satzspiegel für den TUD-Kopf zusammen mit dem DRESDEN-concept-Fuß erstellt. Das Einbeziehen der Fußzeile in den Satzspiegel erfolgt nicht ggf. über die Option includefoot sondern manuell bei der Definition von \tud@cdgeometry@ddcmargin.

987 \eappto\tud@cdgeometry@hmargin{,layouthoffset=\the\Gm@layouthoffset}%
988 \eappto\tud@cdgeometry@vmargin{,layoutvoffset=\the\Gm@layoutvoffset}%
989 \tud@if@v@lower{2.03}{%
990 \edef\@tempa{%
991 \tud@cdgeometry@hmargin,%
992 \tud@cdgeometry@vmargin,%
993 \tud@cdgeometry@tudmargin,%
994 \tud@cdgeometry@ddcmargin,%
995 ignorehead,ignorefoot%
996 }
997 \tud@setgeometry{\@tempa}%
998 \tud@savegeometry{tudareaddc}%
999 %}

Danach folgt der Seitenstil, mit dem TUD-Kopf und der Standardfußzeile.

1000 \edef\@tempa{%
1001 \tud@cdgeometry@hmargin,%
1002 \tud@cdgeometry@vmargin,%
1003 \tud@cdgeometry@tudmargin%
1004 %}
1005 \tud@setgeometry{\@tempa}%
1006 \tud@savegeometry{tudarea}%

Als letztes wird der Standardsatzspiegel erstellt.

1007 \edef\@tempa{%
1008 \tud@cdgeometry@hmargin,%
1009 \tud@cdgeometry@vmargin%
1010 %}
1011 \tud@setgeometry{\@tempa}%
1012 \tud@savegeometry{stdarea}％
Falls die (abermalige) Erstellung der Satzspiegel im Dokument erfolgt, sollte anschließend auch der richtige Satzspiegel wieder ausgewählt werden.

\if@atdocument
\ifstr{\tud@currentgeometry}{stdarea}
{expandafter\tud@loadgeometry\expandafter{\tud@currentgeometry}\
}%
\fi
%
\tud@cdgeometry@hincl
\tud@cdgeometry@fincl
\tud@cdgeometry@mpincl

Diese Hilfsmakros werten die Optionen für das Einbeziehen von Kopf- und Fußzeile sowie der Randnotizen aus.

\newcommand{\tud@cdgeometry@hincl}{\if@hincl\includehead=true\else\includehead=false\fi\}
\newcommand{\tud@cdgeometry@fincl}{\if@fincl\includefoot=true\else\includefoot=false\fi\}
\newcommand{\tud@cdgeometry@mpincl}{\if@mincl\includemp=true\else\includemp=false\fi\}

Damit sind alle notwendigen Satzspiegel für die unterschiedlichen Seitenstile definiert. Im Normalfall werden durch \tud@cdgeometry@@process alle benötigten Satzspiegel erstellt. Mit dem Setzen des Seitenstils wird der jeweils richtige bzw. benötigte Satzspiegel ausgewählt.

\newcommand{\tud@currentgeometry}{}
\newcommand*{\tud@BeforeSelectAnyPageStyle}{\tud@currentgeometry}

Mit \tud@BeforeSelectAnyPageStyle wird beim Umschalten des Seitenstils ggf. der Satzspiegel geändert. Es wird der aktivierte Satzspiegel in dem Makro \tud@currentgeometry gesichert, um darauf zu einem späteren Zeitpunkt prüfen zu können.

Für den Fall, dass das Dokument nicht ausschließlich mit dem Kopf im Corporate Design auf jeder Seite gesetzt wird, muss der komplette vertikale Satzspiegel angepasst werden, sonst wäre der obere Rand optisch viel zu groß. Hierfür wird das Paket geometry genutzt. Durch \tud@cdgeometry@@process werden zwei – für die KOmpatibilitätsvariante drei – Layouts erstellt. Diese können mit \loadgeometry{\{Stil\}} geladen werden.

\newcommand*{\tud@currentgeometry}{}
\newcommand*{\tud@BeforeSelectAnyPageStyle}{\tud@currentgeometry[1]}%
Der \texttt{tudheadings}-Seitenstil wird mit dem Befehl \texttt{\newpairofpagestyles} derart definiert, dass zwischen \texttt{tudheadings} und \texttt{plain.tudheadings} auch mit \LaTeX-Standardseitenstilen \texttt{headings} bzw. \texttt{plain} umgeschaltet werden kann. Dies wird hiermit beachtet.

\begin{verbatim}
\ifstr{#1}{\GetRealPageStyle{#1}}{% 
\tud@if@tudheadings{#1}{% 
  ... wobei in der Version v2.02 zwei unterschiedliche existierten, und abhängig von der Verwendung des DRESDEN-concept-Logos im Fuß waren.

\tud@ddc@check\%  
\% Das DRESDEN-concept-Logo im Fuß ist nicht aktiviert.
\% Ab der Version v2.03 ist nur noch ein Satzspiegel vonnöten.
\ifcase\@tempb\relax% \tud@ddc@foot@num=false
\ifstr{\tud@currentgeometry}{tudarea}{}{\% 
\tud@loadgeometry{tudarea}\%
\gdef\tud@currentgeometry{tudarea}\%
\}%
\else% \tud@ddc@foot@num!=false
\ifstr{\tud@currentgeometry}{tudareaddc}{}{\% 
\tud@loadgeometry{tudareaddc}\%
\gdef\tud@currentgeometry{tudareaddc}\%
\}%
\fi%
\}
\% Die Länge \texttt{\tud@len@areavskip} gibt an, wie groß die Differenz zwischen Kopfhöhe zwischen aktuellem und dem speziellen \texttt{tudheadings}-Seitenstil ist. Da diese in diesem Fall identisch sind, wird die Länge zu \texttt{\z@} gesetzt.
\setlength{\tud@len@areavskip}{\z@}\%
\% Im Seitenfuß wird für die Seitenzahl und ggf. die Kolumnentitel die passende Schrift verwendet.
\tud@font@koma@set{pagefoot}{\usekomafont{tudheadings}}\%
\tud@font@koma@set{pagenumber}{\usekomafont{tudheadings}}\%
\end{verbatim}

File f: tudscr-area.dtx
Dies ist der Fall, wenn kein `tudheadings`-Seitenstil geladen werden soll. Zusätzlich zum Satzspiegel wird außerdem der Seitenstil `empty` zurückgesetzt.

Die Länge \texttt{\tud@len@areavskip} wird auf den berechneten Wert gesetzt. Nur für den Fall, dass die Kopfzeile zum Satzspiegel gerechnet wird und dieser nicht durch \texttt{typearea} berechnet wurde, muss diese Länge etwas angepasst werden.

Die Schriften für Seitenzahl und Kolumnentitel werden zurückgesetzt.

Die Länge \texttt{\tud@len@areavskip} wird vorsichtshalber global gesetzt.

\texttt{\tud@AfterChangingArea} Der Befehl `\AfterCalculatingTypearea` wird durch das Paket `typearea` bereitgestellt. Die durch `typearea` berechneten Längenwerte für den Satzspiegel werden zur Weiterverarbeitung gesichert. Für die Erstellung des Satzspiegels wird das Paket `geometry` verwendet. Bei der entsprechenden Option (\texttt{cdgeometry=no}) wird jedoch der von \texttt{typearea} berechnete Satzspiegel an \texttt{geometry} weitergereicht. Somit wird es möglich, unabhängig vom genutzten Paket (`typearea` oder `geometry`) zur Festlegung des Satzspiegels, diesen innerhalb des Dokumentes zu ändern.

Nach der Änderung des Satzspiegels werden alle Logoboxen neu erstellt.

Damit Änderungen am Satzspiegel im Dokument sicher übernommen werden, wird das Setzen des Satzspiegels forciert. Dafür wird \texttt{\tud@currentgeometry} zurückgesetzt und anschließend der aktuelle Seitenstil erneut geladen, um das erneute und korrekte Ausführen von `\tud@BeforeSelectAnyPageStyle` zu erzwingen.
Damit alle Satzspiegelinstellungen korrekt vorgenommen und auch alle Klassenoptionen korrekt verarbeitet werden, wird die Berechnung erstmalig mit \AtEndPreamble am Ende der Präambel ausgeführt. Dies geschieht jedoch nicht, wenn die standalone-Klasse mit der Option crop geladen wurde, um die Seitenränder nicht zu ändern.

Am Ende der Präambel wird der Satzspiegel des Dokumentes gesetzt. Dabei werden auch sowohl der benutzerdefinierte Satzspiegel als auch die durch typearea berechneten Einstellungen gesichert.

Ist die Option cdgeometry=custom nicht aktiv, werden mit \geometry etwaig gemachte Einstellungen überschrieben.

Der Kram ist dafür da, etwaige Warnungen von typearea aufgrund der fehlenden Wahl für die Satzspiegelauflage zu unterdrücken.
Falls `cdgeometry=custom` aktiv ist, werden die Einstellungen übernommen.

Nach einer Neuberechnung oder Änderung des Satzspiegels durch `typearea` werden die für die Seitenstile benötigten Satzspiegel mit `geometry` neu erstellt.

Der Kopf im Corporate Design der Technischen Universität Dresden erfordert einen eigenen Satzspiegel, der ggf. durch `\BeforeSelectAnyPageStyle` aktiviert bzw. deaktiviert wird.

Sollte das Laden des Paketes `geometry` verhindert werden, sind einige wenige Befehle vorzuhalten.
Damit die Satzspiegelumstellungen nicht durchgeführt werden, wenn durch die folgenden KOMA-Script-Befehle Leerseiten erzeugt werden, müssen diese minimal angepasst werden.

\let\tud@BeforeSelectAnyPageStyle\@gobble
\providecommand*\tud@setgeometry[1]{}
\providecommand*\tud@savegeometry[1]{}
\tud@ddc@enlargepage

Der Befehl wird nur für den Kompatibilitätsmodus zur Version v2.02 benötigt und für die Titelkopf- und Kapitelseiten verwendet. Er verkleinert die entsprechenden Seiten, wenn für diese mit dem Befehl \thispagestyle einer der tudheadings-Seitenstile gewählt wird und das DRESDEN-concept-Logo im Fuß gesetzt werden soll. Der entsprechende Seitenstil kann im optionalen Argument angegeben werden, wenn der Befehl bedingt ausgeführt werden soll.

\tud@if@v@lower{2.03}\{
\newcommand*\tud@ddc@enlargepage[1][]{
\tud@if@tudheadings{#1}{\ifstr{\tud@currentgeometry}{tudareaddc}}{
\tud@ddc@check\%
\ifcase\@tempb\relax\else\tud@ddc@foot@num=false
\enlarge@thispage{\tud@len@ddcdiff}\%
\fi\%
\}}\%
}}

File: tudscr-area.dtx 103
10 Der Seitenstil des Corporate Designs

Es wird der Seitenstil des Corporate Design der Technischen Universität Dresden mit Logo und dem charakteristischen Querbalken im Kopfbereich definiert. Hierfür kommt das KOMA-Script-Paket `scrlayer-scrpage` zum Einsatz.

10.1 Definition des Seitenstils mit dem Paket `scrlayer-scrpage`

Ein zentraler Bestandteil von TUD-Script ist der Seitenkopf des Corporate Designs. Dieser wird ab der Version v2.02 mit Hilfe des Paketes `scrlayer-scrpage` erzeugt. Dafür werden einzelne Seitenstile erstellt, welche zum einen von verschiedenen Befehlen wie beispielsweise von `\maketitle` oder `\part` bzw. `\addpart` genutzt werden. Zum anderen kann der Anwender selbst diese entweder direkt über die Wahl des Seitenstils oder aber mit der `tudpage`-Umgebung nutzen.

Ist die Klasse `standalone` zusammen mit der Option `crop` aktiv, werden die Seitenränder auf die Einstellungen dieser Klasse gesetzt und das Laden von `scrlayer-scrpage` verhindert.

Es wird ein neuer Seitenstil kreiert, der das Corporate Design der Technischen Universität Dresden mit der prägenden Kopfzeile umsetzt. Dabei soll dieser Kopf auch verwendbar sein, wenn nicht die vom Corporate Design vorgeschriebenen Seitenränder sondern das Paket `typearea` genutzt wird. Dafür wird das Logo im Kopf so wie im Corporate Design vorgesehen mit dem Logo der Technischen Universität Dresden in den Rand und dem Schriftzug bündig zum Textblock gesetzt.

Dies sind das links aus dem Satzspiegel verschobene Logo der Technischen Universität Dresden und das Zweitlogo. Diese werden seit der Version v2.04 nicht mehr in zwei getrennten sondern in einer gemeinsamen Ebene gesetzt.
Vor der Ausgabe der Logoboxen werden diese im Bedarfsfall neu gesetzt.


Die Ebenen für obere und untere Linie des Querbalkens sowohl in der normalen Version, welche den Textbereich überspannt als auch in der Variante über die komplette Seitenbreite.

Fakultät, Einrichtung, Institut und Lehrstuhl als Inhalt des Querbalkens.
Dabei muss für beide Varianten der entsprechenden horizontale Versatz beachtet werden. Dafür wird ggf. der Parameter `head` von `scrlayer-scrpage` verwendet.

```latex
\DeclareNewLayer[%
  background,headsep,%
  voffset=\dimexpr\tud@len@topmargin-.5\tud@len@line\relax,%
  addvoffset=\tud@len@layoutvoffset,%
  contents=%
  \ifnum\tud@head@bar@num<\tw@\relax% cdhead=nocolor/litecolor
  \if@tud@head@widebar\else\tud@head@rule{\layerwidth}\fi%
  \fi%]
  {tudheadings.head.upline}%
\DeclareNewLayer[%
  background,headsep,%
  voffset=\dimexpr\tud@len@topmargin+\tud@len@barheight% 
  -.5\tud@len@line\relax,%
  addvoffset=\tud@len@layoutvoffset,%
  contents=%
  \ifnum\tud@head@bar@num<\tw@\relax% cdhead=nocolor/litecolor
  \if@tud@head@widebar\else\tud@head@rule{\layerwidth}\fi%
  \fi%]
  {tudheadings.head.lowline}%
\DeclareNewLayer[%
  background,%
  width=\dimexpr\tud@len@layoutwidth+\tud@dim@bleedmargin*2\relax,%
  addhoffset=\dimexpr\tud@len@layouthoffset-\tud@dim@bleedmargin\relax,%
  voffset=\dimexpr\tud@len@topmargin-.5\tud@len@line\relax,%
  addvoffset=\tud@len@layoutvoffset,%
  contents=%
  \ifnum\tud@head@bar@num<\tw@\relax% cdhead=nocolor/litecolor
  \if@tud@head@widebar\tud@head@rule{\layerwidth}\fi%
  \fi%]
  {tudheadings.head.uplinewide}%
\DeclareNewLayer[%
  background,%
  width=\dimexpr\tud@len@layoutwidth+\tud@dim@bleedmargin*2\relax,%
  addhoffset=\dimexpr\tud@len@layouthoffset-\tud@dim@bleedmargin\relax,%
  voffset=\dimexpr\tud@len@topmargin-.5\tud@len@line\relax,%
  addvoffset=\tud@len@layoutvoffset,%
  contents=%
  \ifnum\tud@head@bar@num<\tw@\relax% cdhead=nocolor/litecolor
  \if@tud@head@widebar\tud@head@rule{\layerwidth}\fi%
  \fi%]
  {tudheadings.head.lowlinewide}%
\DeclareNewLayer[%
  foreground,headsep,%
  addvoffset=.5\tud@len@headsep,%
  height=\baselineskip,%
  contents={\if@tud@head@date\hfill\tud@date@print\fi},%
  ]
  {tudheadings.head.date}%
```

`tudheadings.head.date` (Layer)

Optionales Datum rechts oberhalb des Textbereiches.

\DeclareNewLayer[%
  background,foot,%
  width=\dimexpr\tud@len@layoutwidth+\tud@dim@bleedmargin*2\relax,%
  hoffset=\dimexpr\tud@len@layoutoffset-\tud@dim@bleedmargin\relax,%
  height=\dimexpr\tud@len@layoutheight-\layeryoffset%
  +\tud@len@layoutoffset+\tud@dim@bleedmargin\relax,%
  addvoffset=-\dp\strutbox,%
  contents={%
    \if@tud@foot@colored%
      \let\@tempc\relax%
      \ifdefvoid{\tud@pagecolor}{%
        \ifcase\tud@head@bar@numelax\or
          \def\@tempc{\color{HKS41!60}}%
          \or
          \def\@tempc{\color{HKS41}}%
        \fi%
      }{%
        \def\@tempc{\color{HKS41}}%
        \ifstr{\tud@pagecolor}{HKS41}{\def\@tempc{\color{HKS41!60}}}{}
      }%
      \ifx\@tempc\relax\else%
        \tud@ddc@check%
        \ifcase\@tempbelax% 	ud@ddc@foot@num=false
          \vbox to \layerheight{%
            \@tempc\rule{\layerwidth}{\layerheight}%
          }
        \fi%
      \fi%
    \}
    },%
  ]{tudheadings.foot.back}%

Bis zur Version v2.02 wurde für die Verwendung des DRESDEN-concept-Logos der Satzspiegel geändert. Hier wird dem Rechnung getragen, indem der farbige Hintergrund im Zweifelsfall vergrößert wird.

\tud@if@v@lower{2.03}{%
  \ModifyLayer[%
    addvoffset=-\tud@len@ddcdiff+\ht\strutbox,%
    contents={%
      \if@tud@foot@colored%
        \let\@tempc\relax%
        \ifdefvoid{\tud@pagecolor}{%
          \ifcase\tud@head@bar@numelax\or
            \def\@tempc{\color{HKS41!60}}%
            \or
            \def\@tempc{\color{HKS41}}%
          \fi%
        }{%
          \def\@tempc{\color{HKS41}}%
          \ifstr{\tud@pagecolor}{HKS41}{\def\@tempc{\color{HKS41!60}}}{}
        }%
        \ifx\@tempc\relax\else%
          \tud@ddc@check%
          \ifcase\@tempbelax% 	ud@ddc@foot@num=false
            \vbox to \layerheight{%
              \@tempc\rule{\layerwidth}{\layerheight}%
            }
          \fi%
        \fi%
      \}
      },%
  ]{tudheadings.foot.back}
Die zweite Ebene wird für die Ausgabe von Drittlogos und des DRESDEN-concept-Logos sowie freien Inhalten im Seitenfuß erstellt. Ab Version v2.03 wird der Satzspiegel so gewählt, dass das DRESDEN-concept-Logo und andere Inhalten in den normalen Seitenfuß passen.

Falls die Höhe der Logos im Fuß nicht durch den Anwender festgelegt wurden, werden passende Standardwerte gesetzt. Außerdem wird die genutzte Höhe des Layers nicht vollständig genutzt, um bei einem farbigen Fußbereich einen gewissen Abstand vom Inhalt zur Kante zu erhalten.

Im Bedarfsfall werden die Boxen vor der Ausgabe durch die einzelnen \...@use-Befehle neu gesetzt. Danach erfolgt die Ausgabe, wobei die Boxen übereinander gelegt werden.

Bis zur Version v2.02 wurde für die Verwendung des DRESDEN-concept-Logos der Satzspiegel geändert. Hier wird dem Rechnung getragen.
Die letzte Ebene dient nicht der Ausgabe eines bestimmten Inhaltes sondern wird für evtl. notwendige Aufräumarbeiten etc. bereitgehalten.

Falls ein Kapitel ohne die Option chapterpage gesetzt wird und dennoch die Gestalt des Kopfes des tudheadings-Seitenstils sich vom restlichen Dokument unterscheiden soll (cd bzw. cdchapter), so dient das Makro \tud@head@bar@restore zum Zurücksetzen auf das normale Verhalten. Siehe \tud@chapter@app.

Aus den zuvor erstellten Ebenen werden jetzt die eigentlichen Seitenstile zusammengesetzt. Als erstes wird der headings-Seitenstil definiert. Dieser verwendet den TUD-Kopf und die Fußzeilen Ebenen des Seitenstils scrheadings, welche zur Definition des Seitenstils verwendet werden. Beim Aktivieren des Seitenstils wird außerdem \tud@ps@init ausgeführt. Damit die Benutzerschnittstelle von scrlayer-scrpage für die Fußzeile weiterhin durch den Anwender verwendet werden kann, werden wie erwähnt die entsprechenden scrheadings-Ebenen verwendet.

Zuletzt werden Aliasnamen für den Seitenstil definiert.
Für den plain-Seitenstil wird ebenso verfahren. Hier werden für die Fußzeilenebenen des Seitenstils plain.scrheadings hinzugefügt.

```
\DeclareNewPageStyleByLayers[%
onselect=tud@ps@onselect,onbackground=tud@ps@onbackground%]
{plain.tudheadings}{%
tudheadings.head.back,%
tudheadings.head.content,%
tudheadings.head.bar,%
tudheadings.head.text,%
tudheadings.head.upline,%
tudheadings.head.lowline,%
tudheadings.head.uplinewide,%
tudheadings.head.lowlinewide,%
tudheadings.head.date,%
tudheadings.foot.back,%
tudheadings.foot.content,%
tudheadings.last,%
plain.scrheadings.foot.odd,%
plain.scrheadings.foot.even,%
plain.scrheadings.foot.oneside,%
plain.scrheadings.foot.above.line,%
plain.scrheadings.foot.below.line%
}%
```


```
\DeclareNewPageStyleByLayers[%
onselect=tud@ps@onselect,onbackground=tud@ps@onbackground%]
{empty.tudheadings}{%
tudheadings.head.back,%
tudheadings.head.content,%
tudheadings.head.bar,%
tudheadings.head.text,%
tudheadings.head.upline,%
tudheadings.head.lowline,%
tudheadings.head.uplinewide,%
tudheadings.head.lowlinewide,%
tudheadings.head.date,%
tudheadings.foot.back,%
tudheadings.foot.content,%
tudheadings.last%
}%
```

Um farbige Titel- Teil- und Kapitelseiten unabhängig vom aktuell verwendeten Seitenstil erstellen zu können, wird die zusätzliche Ebene tudheadings.pagecolor definiert, die allen Seitenstilen hinzugefügt wird. Dabei dient das Makro \tud@pagecolor zum Umschalten der
Hintergrundfarben und wird durch die Auswertung der Hilfsmakros \tud@layout\(\text{Typ}\)\num optionsabhängig für Titel-, Teile- sowie Kapitelseiten gesetzt.

\begin{verbatim}
279  \DeclareNewLayer[%
280  background,%
281  area={\dimexpr\tud@len@layouthoffset-\tud@dim@bleedmargin*2\relax}%
282  \dimexpr\tud@len@layoutwidth+\tud@dim@bleedmargin*2\relax}%
283  \dimexpr\tud@len@layoutheight+\tud@dim@bleedmargin*2\relax}%,%
284  contents={%
285  \ifdefvoid{\tud@pagecolor}{}{%
286  \color{\tud@pagecolor}%
287  \rule{\layerwidth}{\layerheight}%
288  }%}
289  }%,%
290  }{tudheadings.pagecolor}%
291  \AddLayersToPageStyle{@everystyle@}{tudheadings.pagecolor}%
\end{verbatim}

Damit wurden alle Ebenen und die darauf aufbauenden Seitenstile deklariert.

Damit ist die Deklaration der Seitenstile mit \texttt{scrlayer-scrpage} beendet.

\section*{10.1.1 Erweiterung der Seitenstilauswahl}

Da sich die zuvor definierten Seitenstile von den Standardseitenstilen stark unterscheiden und auch einen separaten Satzspiegel benötigen, sind einige Makros zur Kontrollstrukturierung notwendig.

\begin{verbatim}
\tud@if@tudheadings
\tud@ps@list
Mit dem Befehl \texttt{tud@if@tudheadings} kann geprüft werden, ob der im ersten Argument gegebene Seitenstil – was auch das Makro \texttt{currentpagestyle} sein kann, welches durch das Paket \texttt{scrlayer-scrpage} definiert wird und den aktuellen Seitenstil beinhaltet – einem aus der Liste \texttt{tud@ps@list} entspricht. Ist dies der Fall, wird das zweite Argument ausgeführt, andernfalls das dritte. In die Liste \texttt{tud@ps@list} werden die drei zuvor definierten Seitenstile eingetragen.

296  \newcommand*{\tud@ps@list}{}
297  \listadd{\tud@ps@list}{tudheadings}
298  \listadd{\tud@ps@list}{plain.tudheadings}
299  \listadd{\tud@ps@list}{empty.tudheadings}
300  \newcommand*{\tud@if@tudheadings}[3]{%
301  \xifinlist{\GetRealPageStyle(#1)}{\tud@ps@list}{#2}{#3}%
302  }
\end{verbatim}

\begin{verbatim}
\thispagestyle
Wird der Befehl \texttt{thispagestyle} verwendet, muss dafür Sorge getragen werden, dass kein Alias-Seitenstil expandiert wird.

303  \patchcmd{\thispagestyle}{\gdef@specialstyle{#1}}{%
304  \gdef@specialstyle{\GetRealPageStyle{#1}}%
305  }{%}
\end{verbatim}

File g: tudscr-pagestyle.dtx
Für den Fall, dass das Paket `scrlayer-scrpage` nicht geladen wird und dennoch das Makro `\currentpagestyle` wie erwartet definiert ist, wird der originale Befehl etwas angepasst.

```latex
\newcommand*{\tud@pagestyle}{}
\let{\tud@pagestyle}{\pagestyle}
\pretocmd{\tud@pagestyle}{%\tud@BeforeSelectAnyPageStyle{#1}%\edef{\currentpagestyle}{#1}{}%\tud@patch@wrn{\pagestyle}}{}{\tud@patch@wrn{\pagestyle}}
```

Damit die neuen Seitenstile auch vor dem Laden von `scrlayer-scrpage` bereits durch den Anwender ausgewählt werden können, wird `\pagestyle` temporär zurechtgebogen. Die Definition des Makros wird gesichert und direkt vor dem Laden des Paketes wiederhergestellt. Damit werden die Seitenstile erst aktiviert, nachdem diese auch tatsächlich erstellt wurden.

```latex
\tud@cmd@store{\pagestyle}
\renewcommand*{\pagestyle}[1]{\AfterPackage{\scrlayer-scrpage}{\pagestyle{\#1}}}
\BeforePackage{\scrlayer-scrpage}{%\tud@cmd@restore{\pagestyle}%%\edef{\tud@pagestyle}{}%}
```

Um die rudimentäre Verwendung der Klassen auch zu ermöglichen, wenn das Laden des Paketes `scrlayer-scrpage` verhindert wurde, werden für diesen Fall einige notwendige Makros definiert.

```latex
\TUD@UnwindPackage{\scrlayer-scrpage}{%\ifcsndef{if@chapter}{%\newif{if@chapter}\ifcsndef{chapter}{@chapterfalse}{@chaptertrue}%\ifcsndef{chapter}{@chapterfalse}{@chaptertrue}%\cslet{ps@tudheadings}{ps@headings}\cslet{ps@plain.tudheadings}{ps@plain}\cslet{ps@empty.tudheadings}{ps@empty}\providecommand*{\currentpagestyle}{plain}\providecommand*{\BeforeSelectAnyPageStyle}[1]{}\providecommand*{\GetRealPageStyle}[1]{#1}%\tud@ps@onselect
```

Außerdem wird die angepasste Definition von `\pagestyle` verwendet.

```latex
\let{\pagestyle}{\tud@pagestyle}\edef{\tud@pagestyle}{}\tud@ps@onselect
\let{\pagestyle}{\tud@pagestyle}\edef{\tud@pagestyle}{}\tud@ps@onselect
```

Das Erstellen des Satzsiegels erfolgt normalerweise nach dem Paket.

```latex
\tud@cmdgeometry@init\tud@ps@onselect
```

Der Befehl wird bei der Initialisierung der Seitenstile aufgerufen.

```latex
\newcommand*{\tud@ps@onselect}{%\ps@tudheadings\ps@plain.tudheadings\ps@empty.tudheadings\currentpagestyle\BeforeSelectAnyPageStyle\GetRealPageStyle\tud@ps@onselect
```

Normalerweise werden durch `scrlayer-scrpage` bei der Umschaltung auf einen mit dem Befehl `\newpairofpagestyles{\langle\text{Seitenstil}\rangle}\langle\text{Seitenstil}\rangle` definierten Seitenstil, die Seitenstile `headings` und `plain` als Aliase für `\langle\text{Seitenstil}\rangle` und `\langle\text{Seitenstil}\rangle` festgelegt. Allerdings definiert dieser Befehl zusätzliche Ebenen für Kopf- und Fußzeilen, weshalb auf die Verwendung verzichtet wird. Das automatische Umschalten soll dennoch ermöglicht werden.
10.1.2 Gestaltungsvarianten für Kopf- und Fußzeile

Das Makro \tud@head@logocolor enthält die Farbe des Kopfes. Die Gestalt des Querbalkens kann über die Option cdhead geändert werden. Für Titel, Teil und Kapitel gibt es speziell dazugehörige Optionen, womit das Aussehen des Kopfes ggf. angepasst werden kann. Mit \tud@foot@logocolor kann die Farbe eines etwaigen DRESDEN-concept-Logo gleichermaßen festgelegt werden, in \tud@foot@fontcolor wird ggf. die Schriftfarbe für den Fuß gespeichert. Um die Optionen innerhalb des Dokumentes ändern zu können, werden die Anpassungen bei jeder Ausgabe eines Seitenstiles mit \tud@ps@onbackground ausgeführt.

\newcommand*{\tud@head@logocolor}{black}
\newcommand*{\tud@head@fontcolor}{}
\newcommand*{\tud@foot@logocolor}{black}
\newcommand*{\tud@foot@fontcolor}{}
\newcommand*{\tud@ps@onbackground}{%}

Zunächst die Einstellungen für Seiten ohne spezielle Hintergrundfarbe für den Kopf…

\ifdefvoid{\tud@pagecolor}{%
  \ifcase{\tud@head@bar@num}\relax cdhead=nocolor
    \renewcommand*{\tud@head@logocolor}{black}%
    \renewcommand*{\tud@head@fontcolor}{}
    \or cdhead=litecolor
    \renewcommand*{\tud@head@logocolor}{HKS41}%
    \renewcommand*{\tud@head@fontcolor}{HKS41}%
    \or cdhead=barcolor
    \renewcommand*{\tud@head@logocolor}{HKS41}%
    \renewcommand*{\tud@head@fontcolor}{white}%
    \or cdhead=color
    \renewcommand*{\tud@head@logocolor}{white}%
    \renewcommand*{\tud@head@fontcolor}{white}%
  \fi%
%
  \ifdefvoid{\tud@foot@colored}{%
    \ifcase{\tud@head@bar@num}\relax cdhead=nocolor
      \renewcommand*{\tud@foot@logocolor}{black}%
      \renewcommand*{\tud@foot@fontcolor}{}
      \else cdhead=color
        \renewcommand*{\tud@foot@logocolor}{HKS41}%
        \renewcommand*{\tud@foot@fontcolor}{HKS41}%
        \else cdhead=**color
          \renewcommand*{\tud@foot@logocolor}{white}%
          \renewcommand*{\tud@foot@fontcolor}{white}%
          \else
            \renewcommand*{\tud@foot@logocolor}{HKS41}%
            \renewcommand*{\tud@foot@fontcolor}{HKS41}%
          \fi
  }%

...und den Fuß.

\ifdefvoid{\tud@pagecolor}{%
  \ifcase{\tud@head@bar@num}\relax cdhead=nocolor
    \renewcommand*{\tud@foot@logocolor}{black}%
    \renewcommand*{\tud@foot@fontcolor}{}
    \else cdhead=**color
      \renewcommand*{\tud@foot@logocolor}{white}%
      \renewcommand*{\tud@foot@fontcolor}{white}%
      \else
        \renewcommand*{\tud@foot@logocolor}{HKS41}%
        \renewcommand*{\tud@foot@fontcolor}{HKS41}%
      \fi
  \fi%

File g: tudscr-pagestyle.dtx
Die Einstellungen für Seiten mit Hintergrundfarbe variieren davon. Zuerst die für farbige Titel- und Teileseiten...

Die Farbe für die Schrift im Fuß wird für Poster immer weiß gesetzt, falls über die Option \backgroundcolor für den Seitenhintergrund die primäre Hausfarbe genutzt wird.

...sowie die für die farbigen Kapitelseiten.

Abhängig von der eingestzten Schriftfarbe des Kopfes wird die Linienstärke des Querbalkens gesetzt, die Schriftfarbe des Fußes wird global geändert, damit Seitenzahl und Kolumnentitel gegebenenfalls angepasst werden.
10.1.3 Gestaltung des Querbalkens in der Kopfzeile

Die Schrift der Kopfzeile wird entweder in den Schriften des Corporate Designs oder aber in den serifenlosen Standardschriften gesetzt, wobei für beide Varianten sowohl eine fettgedruckte als auch eine normale Version benötigt wird. Die Schriftgröße ist durch das Corporate Design unabhängig von der gewählten Schriftgröße im Dokument vorgegeben und wird in Abhängigkeit von der Papiergröße gewählt und in \texttt{\tud@head@fontsize} gespeichert. Das Makro \texttt{\tud@head@font@@bold} wird verwendet, um die Einstellung der fetten Schriften ohne die fixierte Schriftgröße nutzen zu können.

\begin{verbatim}
\newcommand*{\tud@head@fontsize}{\}
\newcommand*{\tud@head@font@light}{\}
\newcommand*{\tud@head@font@bold}{\}
\newcommand*{\tud@head@font@@bold}{\}
\newcommand*{\tud@head@font@set}{\}
\end{verbatim}

Die Option \texttt{cdhead} wird ausgewertet. Sollte diese inkompatibel gesetzt sein, wird eine Warnung ausgegeben.

\begin{verbatim}
\ifcase\tud@head@font@num\relax false
  \ifnum\tud@cdfont@num>0\relax true/heavy
    \ClassWarning{\TUD@ClassName}{It isn't possible to use 'cdhead=false'
      together with 'cdfont=true'}
  \fi
  \TUDoptions{cdhead=true}
\fi
\end{verbatim}

Die Schriften für den Kopf im Corporate Design werden nachfolgend definiert. Abhängig von \texttt{\tud@cdfont@num} werden entweder die Schriften des Corporate Designs oder aber die serifenlose Standardschrift für die Kopfzeile verwendet. Die Rückfallebene:

\begin{verbatim}
\ifcase\tud@head@font@set\relax
  \DeclareFixedFont{\tud@head@font@light}{\encodingdefault}{\sfdefault}{\mddefault}{\updefault}{\tud@head@fontsize}{\}
  \DeclareFixedFont{\tud@head@font@bold}{\encodingdefault}{\sfdefault}{\bfdefault}{\updefault}{\tud@head@fontsize}{\}
\fi
\end{verbatim}

Und jetzt die eigentlichen Schriften im Corporate Design.

\begin{verbatim}
\ifcase\tud@head@font@num\relax\else true/heavy
  \ifcase\tud@univers\relax
    \DeclareFixedFont{\tud@head@font@light}{\encodingdefault}{\sfdefault}{\mddefault}{\updefault}{\tud@head@fontsize}{\}
  \fi
\fi
\end{verbatim}
Diese Befehle sind Hilfsmakros, die bei der Erzeugung der Textzeile inner- und unterhalb des Querbalkens des TUD-Kopfes dienen.

Dieser Befehl dient zum Füllen einer Liste für die spätere Ausgabe der Kopfzeilenfelder. Es werden sequentiell alle potenziellen Felder hinzugefügt. Sollte das hinzuzufügende Feld dazu führen, dass der Text der Kopfzeile über den Seitenrand hinausragen würde, so wird eine weitere Zeile begonnen. Das Trennzeichen zwischen einzelnen Feldern muss gepuffert werden, da dieses nur gesetzt werden soll, wenn ein weiteres Feld nachfolgt.

Sollte das hinzuzufügende Feld die aktuelle Kopfzeile über die Textbreite hinaus erweitern, wird der bisherige Inhalt in die Liste gespeichert und eine neue mit dem aktuellen Inhalt begonnen. Andernfalls wird der aktuelle Inhalt dem bestehenden hinzugefügt.

Für die Felder im Kopf wird bei einer Änderung dieser die Warnung bezüglich der zu großen Breite der Kopfinformationen, was bei kleinen Papierformaten geschehen kann, (re-)definiert.
Der Inhalt der Kopfzeile wird durch die angegeben Fakultät etc. vorgegeben. Sollte der Platz in einer Zeile dafür nicht ausreichen, wird eine weitere Zeile begonnen. Ob diese benötigt wird, ist abhängig vom Inhalt und von der sich daraus ergebenden Breite der Textzeile. Das Erzeugen der einzelnen Textzeilen im Kopf, die in \tud@head@text@list gesichert werden, erfolgt mit \tud@head@text@add[[ Trennzeichen]]{{Schrift}}{{Feld}}. Die dafür benötigten Hilfsmakros werden vor der Verwendung initialisiert.

Ganz zum Schluss wird der Rest aus \tud@head@text@line ebenso wie ggf. die zusätzliche Zeile in die Liste expandiert.

Dieser Befehl ist der Inhalt der Ebene für den Text der Kopfzeile.

Sollte keine Fakultät angegeben sein, wird eine Warnung ausgegeben.

Die Ausgabe der Textzeile(n) mit Fakultät etc. im Kopfbereiche erfolgt über das Makro \tud@head@text@write. Hierbei ist zu beachten, dass für die erste Zeile bei einem zweifarbigem Kopf eine andere Textfarbe als für die restlichen benötigt wird. Außerdem ist für diese aufgrund der Linien im Querbalken ein etwas größerer Abstand zur zweiten Zeile notwendig, weshalb die temporären Makros \@tempb und \@tempc einmalig genutzt werden. Die Farbe des Textes der ersten Zeile entspricht immer der des Kopfes.
Das Makro \texttt{@tempa} wird zur Abbreitung der Liste \texttt{@head@text@list} definiert.

Sollte eine der Textzeilen zu lang sein, was insbesondere bei kleineren Papierformaten vorkommen kann, dann wird eine Warnung ausgegeben.

Die weiteren Zeilen sind bei zweifarbigem Kopf normalerweise blau, nur für einen dunklen Seitenhintergrund weiß.

Dies ist nun die eigentliche Ausgabe, welche in einer vertikalen Box erfolgt.

Das Paket \texttt{ragged2e} ändert im Zweifelsfall die Länge \texttt{spaceskip}. Um den Kopf unbeeinflusst davon immer in der gleichen Gestalt erscheinen zu lassen, wird dies hier temporär unterdrückt.

\begin{verbatim}
\texttt{tud@head@text@write}
\end{verbatim}

\begin{verbatim}
\texttt{tud@head@rule}
\end{verbatim}
10.1.4 Boxen für Layerinhalte

Die Inhalte für Kopf und Fuß der \texttt{tudheadings}-Seitenstile – sprich das Logo der Technischen Universität Dresden, die DRESDEN-concept-Logos, ein mögliches Zweilogo sowie die Logos und Inhalte im Fuß werden in Boxen gesetzt, um die Anzahl der benötigten Aufrufe von \texttt{\includegraphics} möglichst gering zu halten.

\newcommand*{\tud@newlayerbox}[1]{\expandafter\newsavebox\csname tud@layer@#1\endcsname}
\newcommand*{\tud@savelayerbox}[1]{\global\expandafter\sbox\csname tud@layer@#1\endcsname}
\newcommand*{\tud@uselayerbox}[1]{\expandafter\usebox\csname tud@layer@#1\endcsname}
\tud@vlayerbox

Für die Ausgabe der Logos und Inhalte wird dieser Befehl definiert. Dieser setzt die im zweiten Argument angegebenen Inhalte in eine vertikale Box, deren gewünschte Höhe im ersten Argument angegeben wird. Für den Fall, dass die gewünschten Inhalte die maximale Höhe überschreiten, wird eine Warnung erzeugt. Der Inhalt dieser Warnung wird im dritten Argument angegeben. Damit soll dem Anwender geholfen werden, in diesem Fall die richtigen Maßnahmen zu ergreifen und nicht lediglich eine Meldung einer zu übervollen \texttt{vbox} zu erhalten.

\newcommand*{\tud@vlayerbox}[3]{\begingroup\setbox\z@\vbox{#2}\setlength\@tempdima{\dimexpr\ht\z@+\dp\z@-#1\relax}\ifdim\@tempdima>\z@\relax\ClassWarning{\TUD@Class@Name}{The maximum height is exceeded by \the\@tempdima\MessageBreak}\fi\endgroup\vbox to #1{#2}}

10.2 Der Kopfbereich der Seitenstile

Es folgen Option und Befehle zur Gestaltung des Kopfbereichs mit Hauptlogo, Querbalken und ggf. Zweilogo.

Mit dieser Option wird die zentrale Benutzerschnittstelle für Einstellungen des typischen Querbalkens für den TUD-Kopf geschaffen. Durch sie können sowohl die verwendete Schrift als auch die Stärke für den Balkentext (Institut, Fakultät etc.) geändert werden. Zusätzlich ist die Laufweite des Querbalkens bzw. die Farbe einstellbar.
Diese drei Boxen werden für das Logo der Technischen Universität Dresden reserviert. Abhängig von der gewählten Option des Layouts und der Seitenfarbe wird eine der Logos in der notwendigen Farbe ausgewählt.
10.2.1 Das Hauptlogo der Technischen Universität Dresden

Das Makro \texttt{\tud@mainlogo@set} setzt sowohl das Logo der Technischen Universität Dresden als auch – durch den Aufruf der entsprechenden Befehle zum Schluss – alle weiteren Logos und Inhalte im Kopf und Fuß. Dies geschieht nach jeder Änderung des Satzspiegels beim Ausführen der einzelnen Seitenstil-Layer. Dafür wird der Schalter \texttt{\if@tud@mainlogo@set} im Hook \texttt{\tud@AfterChangingArea} gesetzt.

\begin{verbatim}
\newlength{\tud@len@mainlogoheight}
\newcommand{\tud@mainlogo@wrn}[1]{
  \ifdim#1<\ta@bcor
  \ClassWarning{TUD\Class@Name}{
    The selected page layout means that the logo of TUD extends beyond the printing area.
    The inner margin is smaller than BCOR ('BCOR=\the\ta@bcor', inner margin is \the#1)\MessageBreak%
    Maybe you should decrease the current value of DIV ('DIV=\the\ta@div')%}
  \fi}
\global\let\tud@mainlogo@wrn\@gobble
\def\@tempa{\includegraphics[keepaspectratio, width=\tud@len@logowidth]}
\tud@savelayerbox{main@black}{\@tempa{TUD-black}}
\tud@savelayerbox{main@HKS41}{\@tempa{TUD-blue}}
\tud@savelayerbox{main@white}{\@tempa{TUD-white}}
\settoheight{\tud@len@mainlogoheight}{\tud@uselayerbox{main@black}}
\global\tud@len@mainlogoheight=\tud@len@mainlogoheight
\end{verbatim}

Es soll sichergestellt werden, dass das bei der Verwendung von \texttt{typearea} über den Satzspiegel in den Seitenrand hinausragende TUD-Logo nicht außerhalb des Druckrandes liegt. Dies kann beispielsweise bei kleinen Papierformaten, zweisitzigem Satz und/oder einem zu großen DIV-Wertes passieren. Es wird in diesem Fall eine Warnung ausgegeben, damit das Problem durch den Anwender behoben werden kann.

\begin{verbatim}
\gdef\tud@mainlogo@wrn##1{\ifdim##1<\ta@bcor
  \ClassWarning{TUD\Class@Name}{
    The selected page layout means that the logo of TUD extends beyond the printing area.
    The inner margin is smaller than BCOR ('BCOR=\the\ta@bcor', inner margin is \the#1)\MessageBreak%
    Maybe you should decrease the current value of DIV ('DIV=\the\ta@div')%}
  \fi}
\end{verbatim}

Hier werden die beiden Makros aufgerufen, um die Inhalte von Kopf und Fuß in den Boxen in der richtigen Größe neu zu setzen.

\begin{verbatim}
\tud@headlogo@set
\tud@footlogo@option@setfalse
\end{verbatim}

\texttt{\tud@mainlogo@use} Hiermit erfolgt die Ausgabe der gespeicherten Boxen für das Logo der Technischen Universität Dresden, wobei \texttt{\tud@head@logocolor} die Farbe festlegt.
10.2.2 Optionales Zweit- oder DRESDEN-concept-Logo

Diese Boxen werden für das DRESDEN-concept-Logo und das Zweitlogo im Kopf definiert. Der Box \tud@layer@head@option fällt dabei die Rolle zu, entweder ein – per Option gewähltes – DRESDEN-concept-Logo oder aber das mit \headlogo angegebene Zweitlogo zusichern und auszugeben. Die anderen drei Boxen speichern das Logo von DRESDEN-concept in den drei für die unterschiedlichen Layoutausprägung benötigten Varianten.


Äquivalent zum Logo der Technischen Universität Dresden werden auch für das Logo von DRESDEN-concept drei Farbvarianten erstellt.

File g: tudcr-pagestyle.dtx
Hiermit werden die DRESDEN-concept-Optionen ddc bzw. ddchead abgearbeitet sowie ein ggf. gegebenes Zweitlogo gespeichert. Dies geschieht beim Aufruf des passenden Layers, falls der Schalter `if@tud@headlogo@option@set` gesetzt wurde.

\newcommand*{\tud@headlogo@option@set}{%
  \def\tempc##1##2{%\tud@savelayerbox{head@option}{%\includegraphics[keepaspectratio,totalheight=\tud@len@mainlogoheight,##2]{##1}}%
  \tud@savelayerbox{head@option}{}
  \tud@ddc@check%
  \ifcase\@tempa\relax%
    \tud@ddc@head@num=false
    \ifdefvoid{\tud@headlogo@filename}{}{%
      \protected@edef\tud@headlogo@filename{\expandafter\trim@spaces\expandafter{\tud@headlogo@filename}}%
      \protected@edef\@tempb{\@tempc{\tud@headlogo@filename}{\tud@headlogo@fileoptions}}%
      \@tempb%
    }%
    \or\or\or\or\or\or\
  \tud@ddc@head@num=color
  \@tempc{DDC-03}{}
  \tud@ddc@head@num=colorblack
  \@tempc{DDC-09}{}
  \tud@ddc@head@num=gray
  \@tempc{DDC-21}{}
  \tud@ddc@head@num=black
  \@tempc{DDC-24}{}
  \tud@ddc@head@num=blue
  \@tempc{DDC-27}{}
  \tud@ddc@head@num=white
  \@tempc{DDC-30}{}
  \fi%
  \global\tud@headlogo@option@setfalse%
%}
\tud@headlogo@use

Die Ausgabe von DRESDEN-concept- oder Zweitlogo im Kopf erfolgt mit diesem Makro. Dabei werden diese in einer Box vertikal zentriert.

\newcommand*{\tud@headlogo@use}{%
  \tud@vlayerbox{\tud@len@mainlogoheight}{%\vss%
    \hbox{%\tud@ddc@check%
      \ifnum\@tempa=\@ne\relax\tud@ddc@head@num=true
10.3 Der Fußbereich der Seitenstile

Es folgen Option und Befehle zur Gestaltung des Fußbereichs mit optionalem Inhalt im Fuß, Drittoligos und DRESDEN-concept-Logo.

Dieser Seitenstil mit dem Kolumnentitel im Fuß wurde bereits für die alte Tudbook-Klasse bereitgestellt und soll auch hier optional angeboten werden. Zusätzlich kann mit dieser Option ggf. ein farbiger Hintergrund in der Fußzeile aktiviert werden.

Einige Einstellungen sind abhängig vom Paket scrlayer-scrpage, welches unter Umständen nicht geladen wird. Mit dem Befehl \TUD@AfterPackage@do kann Quelltext nur dann ausgeführt werden, wenn ein Paket auch tatsächlich geladen wurde. Hierfür ist die Initialisierung mit \TUD@AfterPackage@set nötig.

Die ursprünglichen Kopf- und Fußzeilen werden mit \tud@ps@store gesichert, um gegebenenfalls zurückzuschalten zu können. Anschließend werden diese mit dem neuen Stil überschrieben. Dies wird allerdings frühestens nach dem Laden von scrlayer-scrpage durchgeführt.
Hier kann die Hintergrundfarbe des Fußes aktiviert bzw. deaktiviert werden.

Außerdem kann mit der Option `extrabottommargin` gesetzt werden.

Die nächsten beiden Befehle dienen zum Sichern der Kopf- und Fußzeilen...

..., wobei hierfür auf die internen Befehle von `scrlayer-scrpage` zurückgegriffen werden muss.
10.3.1 Optionales DRESDEN-concept-Logo oder Drittlogos

Für den Anwender besteht mit der Option `footlogoheight` die Möglichkeit, die Höhe aller Logos im Fußbereich – also eventuell das DRESDEN-concept-Logo sowie vom Anwender mit dem Befehl `\footlogo` angegebene Logos – zentral festzulegen. Dabei wird der gewünschte Wert in `\tud@dim@footlogoheight` abgelegt.

Da sich mit der Version v2.03 die Standardhöhe des DRESDEN-concept-Logos im Fußbereich geändert hat, wird dieser Wert abhängig von der Einstellung für die Option `tudscrver` nach der Abarbeitung der Optionen am Ende der Klasse gesetzt.

Im Fußbereich kann das DRESDEN-concept-Logo rechtsbündig ausgegeben. Hierfür werden die entsprechenden Boxen für alle Farbvarianten definiert. Diese werden genutzt, wenn
entweder die Option ddcfoot oder aber ddc in Verbindung mit einem Zweitlogo (\headlogo) genutzt wird.

\begin{verbatim}
859 \tud@newlayerbox{foot@black}
860 \tud@newlayerbox{foot@HKS41}
861 \tud@newlayerbox{foot@white}
862 \tud@newlayerbox{foot@option}
\end{verbatim}

Auch im Fuß werden für das DRESDEN-concept-Logo verschiedene Boxen für die Farbvarianten black, HKS41 und white sowie eine für die gezielte Auswahl über die Optionen ddc bzw. ddcfoot erstellt. Dafür wird zunächst das Makro \tud@footlogo@ddc@set definiert.

\begin{verbatim}
863 \newcommand*{\tud@footlogo@ddc@set}[2]{%
864 \tud@savelayerbox{foot@#1}{%
865 \includegraphics[keepaspectratio,totalheight=\tud@dim@footlogoheight]{#2}%
866 }%
867 }
\end{verbatim}

Hiermit werden die DRESDEN-concept-Optionen ddc bzw. ddcfoot abgearbeitet. Dies geschieht bei der Ausgabe des dazugehörigen Layers, falls hierfür der entsprechende Schalter \if@tud@footlogo@option@set gesetzt wurde.

\begin{verbatim}
868 \newcommand*{\tud@footlogo@option@set}{%
869 \if@tud@footlogo@option@set%
870 \tud@footlogo@ddc@set{black}{DDC-22}%
871 \tud@footlogo@ddc@set{HKS41}{DDC-25}%
872 \tud@footlogo@ddc@set{white}{DDC-28}%
873 \tud@ddc@check%
874 \ifcase@\@tempb\relax\or\or% \tud@ddc@foot@num=color
875 \tud@footlogo@ddc@set{option}{DDC-01}%
876 \or% \tud@ddc@foot@num=colorblack
877 \tud@footlogo@ddc@set{option}{DDC-07}%
878 \or% \tud@ddc@foot@num=gray
879 \tud@footlogo@ddc@set{option}{DDC-19}%
880 \or% \tud@ddc@foot@num=black
881 \tud@footlogo@ddc@set{option}{DDC-22}%
882 \or% \tud@ddc@foot@num=blue
883 \tud@footlogo@ddc@set{option}{DDC-25}%
884 \or% \tud@ddc@foot@num=white
885 \tud@footlogo@ddc@set{option}{DDC-28}%
886 \fi%
887 \ifcase@\@tempb\relax\or\or% \tud@ddc@foot@num=!false
888 \tud@footlogo@ddc@setfalse%
889 \fi%
90 }%
\end{verbatim}


\begin{verbatim}
890 \newcommand*{\tud@footlogo@ddc@use}[2]{%
891 \tud@ddc@check%
892 \ifcase@\@tempb\relax\else% \tud@ddc@foot@num=!false
893 \tud@footlogo@option@set%
894 \setboxz@hbox{%
895 \tud@vlayerbox{#1}{%
896 \vss%
897 \hbox to #2%
898 \hss%
899 \ifcase@\@tempb\relax\or% \tud@ddc@foot@num=true
\end{verbatim}
Nach der Ausgabe wird ein negativer horizontaler Abstand in der Boxbreite eingefügt, damit nachfolgende Boxen überlagert werden können.

\usebox\z@\hspace*{-\wd\z@}%
\fi%
}
\tud@layer@foot@cmd

Für das Setzen von benutzerdefinierten Logos im Fußbereich wird diese Box verwendet.

\tud@newlayerbox{foot@cmd}

Mit dem Befehl \footlogo kann eine kommasparierte Liste für Logos im Fuß angegeben werden, die mit \footlogosep voneinander getrennt werden.

\newcommand*{\footlogosep}{\hfill}
\newcommand*{\tud@footlogo@filenames}{}
\newcommand*{\tud@footlogo@fileoptions}{}
\newcommand*{\footlogo}[2][{}]{%
\tud@comp@clearpage%
\renewcommand*{\tud@footlogo@filenames}{#2}%
\renewcommand*{\tud@footlogo@fileoptions}{#1}%
\global{\tud@footlogo@cmd@settrue%}
\tud@comp@resetpagestyle%
%
Die mit \footlogo angegebene Logos und die dazugehörigen Einstellungen, welche entweder allgemein über das optionale Argument oder aber individuell durch : an den Dateinamen angehängt übergeben wurden, werden hier in der Box \tud@layer@foot@cmd gespeichert.

\newif{\if@tud@footlogo@cmd@set}
\newcommand{\tud@footlogo@cmd@set}[2][{}]{%
\if@tud@footlogo@cmd@set%
\tud@savelayerbox{foot@cmd}{%
Der Befehl \tud@footlogo@cmd@set wird genutzt, um die einzelnen Logos für den Fuß nacheinander in der Box \tud@layer@foot@cmd zu setzen.

\newcommand*{\tud@footlogo@cmd@@set}[2]{%
\begingroup%
\def\@tempa{}
\def\@tempb{}
\@tempswatrue%
\def\do##1{%\if@tempswa\def\@tempa{##1}\@tempswafalse\else\expandafter\def\expandafter\@tempb\expandafter{\@tempb,##1}\fi%}
\let\@tempc\relax%
\DeclareListParser{\@tempc}{:}%\@tempc{#1}%
\preto{\tud@footlogo@fileoptions}{%%
\def\@tempa{}%\def\@tempb{}%\@tempswatrue%
\def\do##1{%\if@tempswa\def\@tempa{##1}\@tempswafalse\else\expandafter\def\expandafter\@tempb\expandafter{\@tempb,##1}\fi%}
\let\@tempc\relax%
\DeclareListParser{\@tempc}{:}%\@tempc{#1}%
\def\@tempa{}
\def\@tempb{}
\def\do##1{%\if\@tempswa\def\@tempa{##1}\@tempswafalse\else\expandafter\def\expandafter\@tempb\expandafter{\@tempb,##1}\fi%}
\let\@tempc\relax%
\DeclareListParser{\@tempc}{:}%\@tempc{#1}%
\protected@edef\@tempc{%\noexpand\includegraphics[\@tempa]{\@tempb,##1}%%
\fi%}
\let\@tempc\relax%
\DeclareListParser{\@tempc}{:}%\@tempc{#1}%
\preto{\tud@footlogo@fileoptions}{%%
\keepaspectratio, totalheight=\tud@dim@footlogoheight,%
\}%
\appto{\tud@footlogo@fileoptions}{\@tempb}%
\protected@edef\@tempc{%\noexpand\includegraphics[\@tempa]{\@tempb}%%
\fi%}
\let\@tempc\relax%
\DeclareListParser{\@tempc}{:}%\@tempc{#1}%

In jedem Fall werden die Grundeinstellungen als erstes ausgeführt.

\preto{\tud@footlogo@fileoptions}{%\keepaspectratio, totalheight=\tud@dim@footlogoheight,%
}%

Wurden tatsächlich individuelle Parameter übergeben, so werden diese als letztes ausgeführt, um vorige Einstellungen überschreiben zu können.

\appto{\tud@footlogo@fileoptions}{\@tempb}%\protected@edef\@tempc{%\noexpand\includegraphics[\@tempa]{\@tempb}%%
}%

Die Logos werden – wie auch das DRESDEN-concept-Logo im Fuß – individuell optisch vertikal zentriert.

\tud@vlayerbox{#2}{\vss\hbox{\@tempc}\vss\vss}{%
Der Befehl \tud@footlogo@cmd@use wird genutzt, um die einzelnen Logos für den Fuß innerhalb der Ebene tudheadings.foot.content auszugeben.

Mit dem Befehl \footcontent kann der Inhalt für den Fußbereich der Seiten im Stil tudheadings festgelegt werden. Dieser wird mit passenden Schrifteinstellungen entweder einspaltig oder zweispaltig gesetzt. Die Sternversion des Befehls nimmt keinerlei Einstellungen für die Schrift vor.

Die mit \footcontent angegebenen Inhalte werden in diesen drei Makros gesichert, weshalb diese vorher alloziert werden. Anschließend erfolgt die Definition des Befehls.

Die Sternversion ändert die Schriftgröße vom Inhalt nicht. Die Normalversion nutzt die Schriftgröße der Kopfzeile mit einem passenden Durchschuss.

Normalerweise wird der Inhalt vollständig über die verfügbare Seitenfußbreite gesetzt. Wird das optionale Argument nach dem obligatorischen genutzt, so erfolgt die Ausgabe zweispaltig. Das optionale Argument kann für zusätzliche Formatierungsanweisungen genutzt werden.
Wurde in einem der Argumente für den linken und/oder rechten Inhalt ein Stern verwendet, bleibt die bisherige Definition bestehen.

Mit diesen beiden Hilfsmakros werden die linke und die rechte Spalte des Standard-Seitenfußes eines Posters festgelegt. In der linken Spalte werden dabei Fakultät, Einrichtung, Institut und Lehrstuhl sowie der Professor ausgegeben, wobei die Angaben über das optionale Argument der entsprechenden Feldbefehle, die in den Klassen normalerweise nur für den Seitenkopf genutzt werden, variert werden können.

In der rechten Spalte werden der Autor oder die Autoren (\author) und die Kontaktperson(en) (\contactperson) ausgegeben. Zu jeder Person können individuelle Angaben bzgl. Büro, Telefonnummer und E-Mail-Adresse gemacht werden.

Wurde kein Autor angegeben, wird in diesem Fall die normalerweise erzeugte Warnung bei der Verwendung des Feldes \author unterdrückt.
Das Makro zum Aufteilen der Autorenangaben wird für die hier benötigte Form definiert. Dabei wird die Ausgabe aller nicht lokal angegebenen Felder unterdrückt, indem der Befehl `\tud@multiple@fields@preset` im zweiten Argument mit einem `*` aufgerufen wird.

Wurde gültige Felder außerhalb von `@author` global angegeben, so werden diese nach allen Autoren ausgegeben.

Wenden Autor und Kontaktperson ausgegeben, ist eine Leerzeile einzufügen.

Die Ausgabe der Kontaktperson(en) erfolgt analog zu der Autorenausgabe.

Zu guter letzt noch eine mögliche Homepage.
Mit \texttt{\tud@footcontent@use} erfolgt die Ausgabe der mit \texttt{\footcontent} definierten Inhalte. Dabei wird darauf geachtet, ob ein DRESDEN-concept-Logo im Fuß verwendet wird. Die Ausgabe erfolgt jedoch nur, falls tatsächlich etwas definiert wurde. Andernfalls wird lediglich eine leere Box der geforderten Breite ausgegeben. Dies ist notwendig, da nachgelagerte Funktionen diese Box fester Breite erwarten.

Äquivalent zu den Logos wird auch der durch den Anwender frei definierbare Inhalt des Fußes in einer vertikalen Box gesetzt. Mit \texttt{\@tempdimc} wird im Bedarfsfall der Freiraum für das DRESDEN-concept-Logo bereitgestellt.

Wurde das optionale Argument nach dem obligatorischen nicht genutzt, so wird der Inhalt über die komplette Breite des Textbereiches ausgegeben.

Im zweispaltigen Modus werden zwei vertikale Boxen erzeugt, zwischen denen der Abstand \texttt{\columnsep} eingefügt wird.
10.4 Optionen für das DRESDEN-concept-Logo

Dieser Schalter wird intern von den Optionen ddc, ddchead und ddcfoot verwendet, um rekursive Aufrufe der Optionen zu verhindern.

Bis zur Version v2.02 wurden unterschiedliche Satzsiegel für den Seitenfuß mit und ohne DRESDEN-concept-Logo verwendet. Damit dieser im Zweifelsfall umgestellt werden kann, wurde vor dem Ausführen der entsprechenden Optionen ein Seitenumbruch erzwungen und nach dem Verarbeiten der Optionen der aktuelle Seitenstil erneut aufgerufen. Die beiden folgenden Makros werden für dieses Unterfangen definiert, jedoch bei der Abarbeitung der Optionen nur noch im Kompatibilitätmodus ausgeführt.

\newcommand*{\tud@comp@clearpage}{}% \newcommand*{\tud@comp@resetpagestyle}{}%
Diese Optionen dienen zur Auswahl des DRESDEN-concept-Logos auf Seiten mit dem Stil \headings. Die Option \ddchead setzt das Logo dabei immer in den Kopf, wobei ein mit \headlogo definiertes Zweitlogo gegebenenfalls überschrieben wird. Die Option \ddcfoot setzt das Logo immer in den Fuß. Mit der Option ddc wird das Logo nur in den Kopf gesetzt, wenn kein Zweitlogo angegeben ist. Ist dies jedoch der Fall, wird das Logo stattdessen im Fuß ausgegeben.

\begin{verbatim}
\newcommand*{\tud@ddc@auto@num}{0}
\if@tud@ddc@auto
\if@tud@headlogo@option@set
\if@tud@footlogo@option@set
\TUD@key{ddc}[true]{}
\end{verbatim}

Die Option ddc.

Im Kompatibilitätsmodus wird ein Seitenumbruch erzwungen. Anschließend wird die Option abgearbeitet.

\begin{verbatim}
\if\tud@comp@clearpage%
\TUD@set@numkey{ddc}{\tud@ddc@auto@num}{\tud@ddc@switch}{#1}%
\fi
\end{verbatim}

Wurde die Option ddc intern durch \ddchead oder \ddcfoot aufgerufen, wird die automatische Auswahl des DRESDEN-concept-Logos deaktiviert.

\begin{verbatim}
\if@tud@ddc@internal%
\TUDoptions{ddchead=false}%
\TUDoptions{ddcfoot=false}%
\else%
\end{verbatim}

Andernfalls wird die automatische Auswahl aktiviert und die beiden Optionen \ddchead sowie \ddcfoot intern zurückgesetzt.

\begin{verbatim}
\tud@ddc@auto@true%
\tud@ddc@internaltrue%
\TUDoptions{ddchead=false}%
\TUDoptions{ddcfoot=false}%
\end{verbatim}

File g: tudscr-pagestyle.dtx
Sowohl die DRESDEN-concept-Logos als auch ein mögliches Zwei-logo (\headlogo) werden in den entsprechenden Layern durch Boxen ausgegeben. Die beiden Schalter führen zum Erneuern der dazugehörigen Logoboxen. Siehe die Beschreibung der Makros \tud@headlogo\option@set sowie \tud@footlogo\option@set.

\global\tud@headlogo\option@settrue\%
\global\tud@footlogo\option@settrue\%
\fi%

Abschließend wird im Kompatibilitätsmodus der aktuelle Seitenstil aufgerufen, um bei einem geänderten Fuß den notwendigen Satzspiegel zu laden.

\tud@comp\resetpagestyle%

Die Option ddchead dient zum zwingenden Setzen des DRESDEN-concept-Logos im Kopf der Seiten im Stil von tudheadings.

\newcommand*{\tud@ddc@head@num}{0}
\TUD@key{ddchead}[true]{% 
Der Seitenumbruch im Kompatibilitätsmodus.

\tud@comp\clearpage%
\TUD@set@numkey{ddchead}{\tud@ddc@head@num}{\tud@ddc@switch}{#1}% 
Beim internen Aufruf soll lediglich der Wert des Schlüssels geändert werden. Wurde die Option durch den Anwender genutzt, werden die korrespondierenden Optionen zurückgesetzt. Der Aufruf der Option ddc führt zum erneuten setzen der Logoboxen.

\ifx\FamilyKeyState\FamilyKeyStateProcessed%
\if\tud@ddc@internal\else%
\@tud@ddc@internaltrue%
\TUDoptions{ddcfoot=false}%
\TUDoptions{ddc=false}%
\@tud@ddc@internalfalse%
\fi%
\fi%

Das Setzen des Seitenstils im Kompatibilitätsmodus.

\tud@comp\resetpagestyle%
}


\newcommand*{\tud@ddc@foot@num}{0}
\TUD@key{ddcfoot}[true]{% 
Der Seitenumbruch im Kompatibilitätsmodus.

\tud@comp\clearpage%
\TUD@set@numkey{ddcfoot}{\tud@ddc@foot@num}{\tud@ddc@switch}{#1}%

File g: tudscr-pagestyle.dtx
Beim internen Aufruf soll lediglich der Wert des Schlüssels geändert werden. Wurde die Option durch den Anwender genutzt, werden die korrespondierenden Optionen zurückgesetzt. Der Aufruf der Option `ddc` führt zum erneuten Setzen der Logoboxen.

Der Schalter wird verwendet, um das explizite Setzen der Schrifteinstellungen über die Seitenstilparameter zu detektieren.

Das Setzen des Seitenstils im Kompatibilitätsmodus.

Dieses Hilfsmakro wird aufgerufen, wenn geprüft wird, welche Einstellungen für Kopf und Fuß des Seitenstiles `tudheadings` zum Tragen kommen sollen. Dabei wird über den Schalter `\if@tud@auto` festgelegt, ob die Auswahl des DRESDEN-concept-Logos automatisch erfolgen soll – Option `ddc` wurde verwendet. Ist dies der Fall, wird das Logo standardmäßig im Kopf gesetzt. Ist jedoch durch den Befehl `\headlogo` ein Zweitlogo angegeben worden, wird dieses in der Fußzeile ausgegeben. Die Verwendung einer der Optionen `ddchead` bzw. `ddcfoot` forciert die Ausgabe des DRESDEN-concept-Logos an der gewünschten Stelle. Die jeweils gültige Auswahl für Kopf bzw. Fuß wird in den Makros `\@tempa` bzw. `\@tempb` geschrieben und kann anschließend ausgewertet werden.

Für die Satzspiegelvergrößerung wird beim Vorhandensein eines oder mehrerer Logos im Fuß die Variable `\@tempb` auf einen negativen Wert gesetzt, um auch ohne die Ausgabe eines DRESDEN-concept-Logos den Fußbereich zu vergrößern.

10.5 Umgebungsparameter für die neuen Seitenstile

Der Schalter wird verwendet, um das explizite Setzen der Schrifteinstellungen über die Seitenstilparameter zu detektieren.
Im Folgenden werden die einzelnen Parameter für die \texttt{tudpage}-Umgebung definiert. Die Parameter für Kopf- und Fußzeile werden nicht nur für die \texttt{tudpage}-Umgebung sondern auch noch für Titel und Umschlagseite verwendet, weshalb deren Definitionen zur einfachen Wiederverwendung in das Makro \texttt{TUD\parameter\@ps\@def} ausgelagert werden.

\newcommand*{\TUD\parameter\@ps\@def}{[1]{\%}
\begin{tabular}{ll}
\textbf{cdfont} (Param.) & Die Option \texttt{cdfont} bestimmt, ob die Schriften des Corporate Designs oder aber die standardmäßigen Serifenlosen für die Auszeichnungen in der Kopfzeile und Serifen im Inhalt verwendet werden. \\
\TUD\parameter\@def{cdfont}{true}\% & \TUD\option{cdfont}{#1}\% & \@tud\parameter@ps@font@settrue\%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{cdhead} (Param.) & Mit dem Parameter \texttt{cdhead} kann die Verwendung der Schriften des Corporate Designs aktiviert werden, wenn diese im Fließtext nicht zum Einsatz kommen. Die Breite des Querbalkens kann über diesen Parameter ebenfalls geändert werden. \\
\TUD\parameter\@def{cdhead}{true}{\TUD\option{cdhead}{#1}}\%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{cdfoot} (Param.) & Über \texttt{cdfoot} kann gg. die Hintergrundfarbe sowie die Standardfußzeile des Corporate Designs aktiviert werden. \\
\TUD\parameter\@def{cdfoot}{true}{\TUD\option{cdfoot}{#1}}\%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{headlogo} (Param.) & Das Zweit- und die Drittlöge können ebenfalls lokal geändert werden. \\
\TUD\parameter\@def{headlogo}{\headlogo{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{footlogo} (Param.) & Mit diesen Parametern kann das DRESDEN-concept-Logo im Kopf bzw. Fuß eingebundet werden. \\
\TUD\parameter\@def{footlogo}{\footlogo{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{ddc} (Param.) & Für die Kompatibilität werden auch veraltete Parameter bereitgestellt. \\
\TUD\parameter\@def{ddc}{true}{\TUD\option{ddc}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{ddchead} (Param.) & \\
\TUD\parameter\@def{ddchead}{true}{\TUD\option{ddchead}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{ddcfoot} (Param.) & \\
\TUD\parameter\@def{ddcfoot}{true}{\TUD\option{ddcfoot}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{cdfonts} (Param.) & \\
\TUD\parameter\@def{cdfonts}{true}{\TUD\option{cdfonts}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{tudfonts} (Param.) & \\
\TUD\parameter\@def{tudfonts}{true}{\TUD\option{tudfonts}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{barfont} (Param.) & \\
\TUD\parameter\@def{barfont}{true}{\TUD\option{barfont}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{widehead} (Param.) & \\
\TUD\parameter\@def{widehead}{true}{\TUD\option{widehead}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{tudfoot} (Param.) & \\
\TUD\parameter\@def{tudfoot}{true}{\TUD\option{tudfoot}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{logo} (Param.) & \\
\TUD\parameter\@def{logo}{true}{\TUD\option{logo}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{ddchead} (Param.) & \\
\TUD\parameter\@def{ddchead}{true}{\TUD\option{ddchead}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{ddcfoot} (Param.) & \\
\TUD\parameter\@def{ddcfoot}{true}{\TUD\option{ddcfoot}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{cdfonts} (Param.) & \\
\TUD\parameter\@def{cdfonts}{true}{\TUD\option{cdfonts}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{tudfonts} (Param.) & \\
\TUD\parameter\@def{tudfonts}{true}{\TUD\option{tudfonts}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{barfont} (Param.) & \\
\TUD\parameter\@def{barfont}{true}{\TUD\option{barfont}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{widehead} (Param.) & \\
\TUD\parameter\@def{widehead}{true}{\TUD\option{widehead}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{tudfoot} (Param.) & \\
\TUD\parameter\@def{tudfoot}{true}{\TUD\option{tudfoot}{#1}}%
\end{tabular}\%}

\newcommand*{\TUD\parameter\@def}{[true]{\%}
\begin{tabular}{ll}
\textbf{logo} (Param.) & \\
\TUD\parameter\@def{logo}{true}{\TUD\option{logo}{#1}}%
\end{tabular}\%}
Damit sind alle notwendigen Parameter für die Kopf- und Fußzeile der Seiten im Stil \texttt{tudheadings} definiert.

Durch den hohen TUD-Kopf muss bei der Verwendung dessen das Seitenlayout temporär umgeschaltet werden, um die Höhe des Satzspiegels zu verringern. Dies geschieht innerhalb dieser Umgebung reversibel mit \texttt{\pagestyle}. Dabei wird mit \texttt{\loadgeometry} ein Seitenumbruch erzwungen und der benötigte Satzspiegel geladen.

Die Befehle \texttt{\tud@currentpagestyle@set} und \texttt{\tud@currentpagestyle@reset} sind für die Sicherung und Wiederherstellung des vor der Umgebung geladenen Seitenstils verantwortlich.

Zu Beginn der Umgebung werden die weiter unten beschriebenen Optionen für diese Umgebung mit \texttt{\TUD@parameter@set} ausgeführt. Nach dem Beenden der Umgebung wird nach einem weiteren Seitenumbruch das Standardseitenlayout wiederhergestellt. Die gewünschte Sprache der Umgebung kann als einzelner Wert ohne Schlüssel angegeben werden.

Um beim Umschalten des Seitenstils den aktuellen selbst zu behalten, wird dieser im Hilfsmakro \texttt{\tud@currentpagestyle@value} gespeichert.
Falls das `multicol`-Paket geladen ist und eine Spaltenanzahl angegeben ist, wird die entsprechende Umgebung gesetzt, mit `\BODY` der Inhalt der `tudpage`-Umbgebung ausgegeben und anschließend gegebenenfalls die `multicols`-Umbgebung beendet.

```
\ifnum\tud@x@multicol@num>\@ne\relax
\begin{multicols}{\tud@x@multicol@num}
\BODY
\ifnum\tud@x@multicol@num>\@ne\relax
\end{multicols}
\fi
\fi
```

Am Ende der Umgebung wird der vorhergehende Seitenstil zurückgesetzt und der dazugehörige Satzspiegel geladen. Dazu wird die Ausführung das Hilfsmakro `\@tempa` auf das Beenden der Umgebung mit `\aftergroup` verzögert.

```
\aftergroup\tud@currentpagestyle@reset
\clearpage
```

Mit `\TUD@parameter@family{⟨Family⟩}{⟨…⟩}` wird die Familie der Parameter festgelegt und anschließend die Definitionen getätigt.

```
\TUD@parameter@family{tudpage}{%}
```

Mit dem Parameter `language` kann die in der Umgebung verwendete Sprache umgeschaltet werden. Die Sprache kann auch ohne den entsprechenden Schlüssel direkt als Parameter angegeben werden.

```
\TUD@parameter@def{language}{\selectlanguage{#1}}%
```

Mit `columns` kann die Anzahl der Spalten für die Umgebung angegeben werden. Für mehr als zwei Spalten muss das Paket `multicol` geladen werden. Die Spaltenanzahl kann auch ohne den entsprechenden Schlüssel direkt Parameter angegeben werden.

```
\TUD@parameter@def{columns}{\cs@std@lock{tud@x@multicol@num}{#1}}%
```

Mit dem Parameter `pagestyle` kann der verwendete Seitenstil eingestellt werden, wobei einer der `tudheadings`-Seitenstile verwendet wird.

```
\TUD@parameter@def{pagestyle}{%}
\tud@if@tudheadings{#1}{\renewcommand*{\tud@envir@ps}{#1}}{%}
\ifstr{#1}{empty}{\renewcommand*{\tud@envir@ps}{empty.tudheadings}}{%}
\ifstr{#1}{plain}{\renewcommand*{\tud@envir@ps}{plain.tudheadings}}{%}
\ifstr{#1}{headings}{\renewcommand*{\tud@envir@ps}{tudheadings}}{%}
\TUD@parameter@error{pagestyle}{%}
\ifstr{#1}{headings, plain, empty or any tudheadings page style type}{%}
\}
\}
\}
```

Hier werden die Parameter für Kopf- und Fußzeile tatsächlich definiert.

```
\TUD@parameter@ps@def{#1}%
```
Für den Fall, dass ein Wert nicht in der Schlüssel-Wert-Notation gegeben wird, erfolgt eine Sonderbehandlung durch \texttt{TUD@parameter@handler@default}, bei der erst auf eine Zahl geprüft wird und anschließend versucht wird, das Argument als Sprache zu setzen. Nach der Definition aller Parameter wird der Befehl \texttt{TUD@parameter@family} und damit auch die aktuelle Parameterfamilie beendet.

\begin{verbatim}
1339 \TUD@parameter@handler@default{}
1340 }

Mit der Nutzung von \texttt{scrlayer-scrpage} ist die parallele Verwendung des Paketes \texttt{fancyhdr} nicht möglich.

\begin{verbatim}
1341 \PreventPackageFromLoading[%
1342 \ClassWarning{\TUD@Class@Name}{%
1343 The package ‘fancyhdr’ must not be used with a \MessageBreak%
1344 TUD-Script class. You should make use of the \MessageBreak%
1345 capabilities of package ‘scrlayer-scrpage’ instead%
1346 }%,
1347 ]{fancyhdr}
\end{verbatim}
11 Das Layout des Corporate Designs

Unter Layout wird die Wahl von Schriftart und -größe, die Positionierung verschiedener Textelemente sowie die farbliche Gestaltung verstanden. Für das Corporate Design gibt es dabei drei wesentliche Ausprägungen, die für die verschiedenen Elemente auch unabhängig gewählt werden können.


11.1 Gestalt von Umschlagseite, Titel, Teile und Kapitel

Ein zentraler Teil der Wrapper-Klassen ist die Auswahl der Darstellung des Corporate Designs. Dabei kann gewählt werden, ob es einfarbig, mit leichtem oder aber mit starkem Farbeinsatz genutzt werden soll. Die folgenden Optionen sind hierfür zuständig.

Diese Option dient für die Hauptklassen zur globalen Einstellung für Cover-, Titel-, Teil- und Kapitelseiten. Mit dem Schalter werden für alle Komponenten die gleichen Einstellungen gesetzt. Die Standardfarbe des Kopfes der Seiten im \texttt{tudheadings}-Stil ist schwarz und wird für die farbigen Layouts auf dunkelblau gesetzt. Abhängig von der gewählten Option – der numerische Wert ist über \texttt{tud@layout@switch} festgelegt – werden die sog. Layoutschlüssel auf einen Wert gesetzt (0...3), welcher später ausgewertet werden kann. Der Schalter für Kapitel wird lediglich für die Klassen \texttt{tudscrbook} und \texttt{tudscrreprt} definiert und genutzt. Damit werden alle Komponenten mit einer zentralen Option festgelegt, können aber auch durch den Nutzer explizit überschrieben werden.
Für die Klasse **tudscrposter** wird mit der Option die farbige Ausprägung eines Posters definiert. Dabei sind alle Werte gültig, die auch für die meisten anderen Layouteinstellungen verwendet werden können. Abhängig vom gewählten Wert, wird der Seitenstil, die Ausprägung der Fußzeile sowie die farbliche Gestalt von Kopf- und Fußzeile festgelegt. Da dabei unter anderem auch gegebenenfalls der Seitenstil auf einen vom Typ **tudheadings** festgelegt wird, erfolgt die Ausführung der Option ebenfalls frühestens zu Beginn des Dokumentes.

14 \{+book | report | article\}
15 \cs@lock{tud@layout@cover@num}{0}
16 {/book | report | article}
17 \cs@lock{tud@layout@title@num}{0}
18 \cs@lock{tud@layout@part@num}{0}
19 \{+book | report\}
20 \cs@lock{tud@layout@chapter@num}{0}
21 {/book | report\}
22 \cs@lock{tud@layout@section@num}{0}

Es folgt die eigentliche Option.

23 \newcommand*{\tud@cd@num}{0}
24 \TUD@key{cd}[true]{% 
25 \TUD@set@numkey{cd}{\tud@cd@num}{\tud@layout@switch}{#1}% 
26 \ifx\FamilyKeyState\FamilyKeyStateProcessed% 

Layout nicht im Corporate Design.

27 \ifcase\tud@cd@num\relax% false 
28 \{+book | report | article\}
29 \cs@std@lock{tud@layout@cover@num}{0}% 
30 {/book | report | article% 
31 \cs@std@lock{tud@layout@title@num}{0}% 
32 \cs@std@lock{tud@layout@part@num}{0}% 
33 \{+book | report\}
34 \cs@std@lock{tud@layout@chapter@num}{0}% 
35 \TUD@std@ifkey@lock{chapterpage}{@tud@chapterpage}{false}% 
36 {/book | report% 
37 \cs@std@lock{tud@layout@section@num}{0}%

Layout im Corporate Design mit schwarzer Schrift und schwarzem Kopf.

38 \or% true 
39 \{+book | report | article\}
40 \cs@std@lock{tud@layout@cover@num}{1}% 
41 {/book | report | article% 
42 \cs@std@lock{tud@layout@title@num}{1}% 
43 \cs@std@lock{tud@layout@part@num}{1}% 
44 \{+book | report\}
45 \cs@std@lock{tud@layout@chapter@num}{1}% 
46 \TUD@std@ifkey@lock{chapterpage}{@tud@chapterpage}{false}% 
47 {/book | report% 
48 \cs@std@lock{tud@layout@section@num}{1}%

Layout im Corporate Design mit blauer Schrift und blauem Kopf.

49 \or% litecolor 

Die Umschlagseite wird standardmäßig immer monochrom gesetzt.

50 \{+book | report | article\}
51 \cs@std@lock{tud@layout@cover@num}{1}%
Layout im Corporate Design mit blauer Schrift und farbigem Querbalken.

\or% barcolor
\cs@std@lock{tud@layout@cover@num}{1}%
\cs@std@lock{tud@layout@title@num}{3}%
\cs@std@lock{tud@layout@part@num}{3}%
\cs@std@lock{tud@layout@chapter@num}{3}%
\TUD@std@ifkey@lock{chapterpage}{@tud@chapterpage}{false}%
\cs@std@lock{tud@layout@section@num}{2}%

Farbiges Layout mit farbigem Hintergrund im Corporate Design mit weißer Schrift und Outline im Kopf.

\or% color
\cs@std@lock{tud@layout@cover@num}{1}%
\cs@std@lock{tud@layout@title@num}{5}%
\cs@std@lock{tud@layout@part@num}{5}%
\cs@std@lock{tud@layout@chapter@num}{5}%
\TUD@std@ifkey@lock{chapterpage}{@tud@chapterpage}{true}%
\cs@std@lock{tud@layout@section@num}{2}%

Farbiges Layout mit farbigem Hintergrund im Corporate Design mit weißer Schrift und farbigem Querbalken im Kopf.

\or% full
\cs@std@lock{tud@layout@cover@num}{1}%
\cs@std@lock{tud@layout@title@num}{6}%
Die Ausführung des Befehls `\tud@layout@process` sorgt dafür, dass alle Optionen auch verarbeitet und wirksam werden. Dieser wird standardmäßig nach der Abarbeitung aller Optionen innerhalb der Präambel einmalig ausgeführt. Deshalb wird er hier nur innerhalb des Dokumentes genutzt. Dies gilt auch für die folgenden Optionen.

Innerhalb der Präambel wird der Satzspiegel abhängig vom gewünschten Layout festgelegt. Dies kann vom Benutzer mit dem Setzen der Option `cdgeometry` überschrieben werden.

cdcover (Opt.) Das durch die Option `cd` insgesamt festgelegte Layout kann für die einzelnen Elemente Umschlagseite (cdcover),...
cdpart (Opt.)  ...Teile- (cdpart) und...

\TUD@key{cdpart}{true}{
  \TUD@set@numkey{cdpart}{@tempa}{\tud@layout@switch}{#1}\%
  \ifx\FamilyKeyState\FamilyKeyStateProcessed\%
    \ifcase\@tempa\relax
      \cs@set@lock{tud@layout@part@num}{0}\%
    \or\ true
      \cs@set@lock{tud@layout@part@num}{1}\%
    \or\ litecolor
      \cs@set@lock{tud@layout@part@num}{2}\%
    \or\ barcolor
      \cs@set@lock{tud@layout@part@num}{3}\%
    \or\ bicolor
      \cs@set@lock{tud@layout@part@num}{4}\%
    \or\ color
      \cs@set@lock{tud@layout@part@num}{5}\%
    \or\ full
      \cs@set@lock{tud@layout@part@num}{6}\%
    \fi
    \TUD@SpecialOptionAtDocument{tud@layout@process}\%
  \fi}

\TUD@key{cdchapter}{true}{
  \TUD@set@numkey{cdchapter}{@tempa}{\tud@layout@switch}{#1}\%
  \ifx\FamilyKeyState\FamilyKeyStateProcessed\%
    \ifcase\@tempa\relax
      \cs@set@lock{tud@layout@chapter@num}{0}\%
    \or\ true
      \TUD@std@ifkey@lock{chapterpage}{@tud@chapterpage}{false}\%
      \cs@set@lock{tud@layout@chapter@num}{1}\%
    \or\ litecolor
      \cs@set@lock{tud@layout@chapter@num}{2}\%
    \or\ barcolor
      \cs@set@lock{tud@layout@chapter@num}{3}\%
    \or\ bicolor
      \cs@set@lock{tud@layout@chapter@num}{4}\%
    \or\ color
      \cs@set@lock{tud@layout@chapter@num}{5}\%
    \or\ full
      \cs@set@lock{tud@layout@chapter@num}{6}\%
    \fi
    \TUD@SpecialOptionAtDocument{tud@layout@process}\%
  \fi}

\TUD@std@ifkey@lock{chapterpage}{@tud@chapterpage}{false}\%

\cdchapter (Opt.)  ...bei den entsprechenden Klassen (\texttt{tudscrbok} und \texttt{tudscrbreport}) auch für die Kapitelseiten (\texttt{cdchapter}) separat für die einzelnen Elemente geändert bzw. überschrieben werden.

\xspace\mbox{}
Zuletzt noch alle nachgelagerten Gliederungsebenen.

Die nun folgenden Optionen und Befehle werden nur für die beiden Klassen `tudscrbook` und `tudscrreport` definiert.

Der Schalter dient zur Steuerung, ob der Titel des Dokumentes wie im Handbuch des Corporate Designs auf den Teileseiten stehen soll.

Mit dieser Option können Kapitelüberschriften alleine auf einer Seite stehen. Es werden separate Kapitelseiten erzeugt.

File h: tudscr-layout.dtx
Für Klassen mit separaten Teileseiten und Kapiteln wird eine zusätzliche Option definiert, welche nur bei zweiseitigem Satz (twoside) und zusätzlich immer auf nur der rechten Seite öffnenden Teilen bzw. Kapiteln (open=right) zum Tragen kommen.


Der Befehl \texttt{\tud@cleardoublepage} enthält die Definition der zu erzeugenden Doppelseite.

Mit der Option cleardoublespecialpage kann dieser festgelegt werden.

Dabei wird die Rückseite entweder wie alle Vakatseiten oder leer gesetzt.

Zusätzlich wird die Seite ggf. auch farbig ausgegeben (s. color).

Außerdem kann ein Seitenstil gezielt ausgewählt werden.

Beim Laden der Klasse selbst sind die speziellen Seitenstile der KOMA-Script- bzw. TUD-Script-Klassen noch nicht definiert. Diese sollen als Werte jedoch trotzdem erlaubt sein. Deshalb wird für diesen Fall das Überprüfen mit \texttt{\AfterPreamble} auf den Beginn des Dokumentes verzögert.

\begin{verbatim}
\newcommand*{\tud@cleardoublepage}{\cleardoubleoddpage}
\end{verbatim}

Mit der Option cleardoublespecialpage kann dieser festgelegt werden.

\begin{verbatim}
\TUD@key{cleardoublespecialpage}[true]{%\TUD@set@ifkey{cleardoublespecialpage}{@tempswa}{#1}%
\ifx\FamilyKeyState\FamilyKeyStateProcessed%
\else%
\renewcommand*{\tud@cleardoublepage}{\null\thispagestyle{empty}\newpage}%
\fi%
\end{verbatim}

\begin{verbatim}
\renewcommand*{\tud@cleardoublepage}{\cleardoubleoddpageusingstyle{#1}}%
\end{verbatim}

\begin{verbatim}
\afterpreamble{%
\ifsundef{ps@#1}{%
\ClassError{\TUD@Class@Name}{`#1' is no valid pagestyle}{%You tried to use `#1' as a pagestyle for option\MessageBreak%`cleardoublespecialpage', but it was never defined.%
}}%\renewcommand*{\tud@cleardoublepage}{\cleardoubleoddstandardpage}%
\FamilyKeyStateProcessed%
}}%
\end{verbatim}

\begin{verbatim}
\renewcommand*{\tud@cleardoublepage}{\cleardoubleoddpageusingstyle{#1}}%
\end{verbatim}

\begin{verbatim}
\afterpreamble{%
\ifsundef{ps@#1}{%
\ClassError{\TUD@Class@Name}{`#1' is no valid pagestyle}{%You tried to use `#1' as a pagestyle for option\MessageBreak%`cleardoublespecialpage', but it was never defined.%
}}%\renewcommand*{\tud@cleardoublepage}{\cleardoubleoddpageusingstyle{#1}}%
\FamilyKeyStateProcessed%
}}%
\end{verbatim}

7bei aktivierter chapterpage-Option
Der Befehl \texttt{\tud@cleardoublespecialpage} dient zur eigentlichen Umsetzung. Das obligatorische Argument enthält alles, was innerhalb der Gruppe an lokalen Einstellungen vorgenommen werden soll. Das optionale Argument wird ggf. genutzt, um einen Inhalt auf der Rückseite auszugeben.

\newcommand*{\tud@cleardoublespecialpage}[2][]{%
\begingroup%
#2
\ifboolexpr{bool {@twoside} and bool {@openright}}{%
  \if@tud@clearcolor%
    \def\@tempa{%
      \tud@cleardoublepage%
    }%
  \else%
    \def\@tempa{%
      \endgroup%
    }%
  \fi%
  }{%
    \if@tud@clearcolor\tud@clearcolor@wrn\fi%
  }%
\@tempa%
\@afterindentfalse\@afterheading%
}\%}{%}

\newif{\if@tud@clearcolor}
\TUD@ifkey{clearcolor}{\@tud@clearcolor}
\TUD@ifkey{clearcolour}{\@tud@clearcolor}

Bei starkem Farbeinsatz im Stil des Corporate Designs auf Titel- Teil- oder Kapitelseiten (\texttt{cd...=color}) wird die entsprechende Rückseite in der gleichen Farbe wie die dazugehörige Vorderseite gestaltet. Für Teile- bzw. Kapitelseiten muss dafür die passende Option (\texttt{cleardoublespecialpage}) aktiviert sein.

\newif{\if@tud@clearcolor}
\TUD@ifkey{clearcolor}{\@tud@clearcolor}
\TUD@ifkey{clearcolour}{\@tud@clearcolor}\%}

\begin{Verbatim}
\texttt{\tud@pagecolor}
\end{Verbatim}

Der Befehl \texttt{\tud@pagecolor} dient für die Hauptklassen als Hilfsmakro, um je nach gewählter Ausprägung des Layouts die farblichen Seitenhintergründe zu aktivieren. Für die Klasse \texttt{tudscrposter} wird der Befehl zusätzlich für das benutzerdefinierte Festlegen einer standardmäßigen Hintergrundfarbe mit der Option \texttt{backgroundcolor} genutzt.

\newcommand*{\tud@pagecolor}{%}

\begin{Verbatim}
\texttt{\tud@pagecolor}
\end{Verbatim}

\begin{Verbatim}
\texttt{\tud@pagecolor}
\end{Verbatim}

Mit dieser Option kann die standardmäßige Hintergrundfarbe für Poster gesetzt werden.
Wird die Option innerhalb des Dokumentes verwendet, wird mit einer Box erst die gewählte Farbe verifiziert und anschließend auf die am Dokumentbeginn definierte Textfarbe umgestellt.

Für das Aktivieren der Farbe HKS41 wird \normalcolor erweitert, um hier in jedem Fall als Schriftfarbe weiß zu verwenden.

11.2 Schrifteinstellung und Positionierung der Überschriften

Die KOMA-Script-Option headings setzt für alle Überschriften die Formatierung zurück. Um die Einstellungen für das Corporate Design aktiv zu halten, werden nach der Ausführung der Option durch \tud@x@scr@headings@reset die mit \tud@font@koma@set definierten Makros \tud@font@koma@⟨Element⟩ erneut an die entsprechenden Schriftelemente angehängt. Bei der Verwendung der KOMA-Script-Option headings=standardclasses werden außerdem noch die Optionen open und chapterprefix ausgeführt, weshalb auch auf diese reagiert werden muss.

File h: tudscr-layout.dtx
Als erstes werden die Optionen verarbeitet, welche einen Einfluss auf die Formatierung respektive die Schriftgröße der Überschriften haben. Wird über die Option `headings` eine andere Überschriftengröße gewählt, werden mit dem Makro `\tud@x@scr@headings@set` die passenden Schriftgrößen sowohl für die Überschriften `\tud@sec@fontsize` als auch die Titelseite `\tud@thesis@fontsize` gesetzt. Dafür wird gewählte Schriftgröße der Überschriften im Hilfmakro `\tud@x@scr@headings@current` gesichert.

Wird `\tud@x@scr@headings@set` aufgerufen, so werden mit `\KOMAoptionOf` die aktuellen Werte der Option `headings` ausgelesen, welche in einer Liste ausgegeben werden. Diese wird durchlaufen und dabei auf die relevanten Werte reagiert.

Damit die Einstellungen für die Größe wirksam werden, wird das Makro einmalig zu Beginn des Dokumentes ausgeführt.
Mit diesen Optionen kann der Anwender die Überschriften von Titel, Teilen und Kapiteln vertikal von ihrer Standardposition verschieben, welche vom Corporate Design eigentlich vorgegeben ist. Die Option `pageheadingsvskip` verschiebt dabei Überschriften, welche allein auf einer Seite stehen, also Überschriften von Teilen oder auf Kapitelseiten. Auch die vertikale Position des Titels kann damit beeinflusst werden. Mit der Option `headingsvskip` werden die Überschriften von Kapiteln, bei denen direkt danach der Fließtext folgt, oder die des Titelkopfes (titlepage=false) verschoben.

Mit `\tud@headmidvskip@normal` wird der Abstand zwischen der Nummerierung eines Titels und der eigentlichen Bezeichnung gesetzt. Gleiches gilt für eine separate Kapitelnummernzeile (chapterprefix) und dem eigentlichen Kapiteltitel. Vorher kann ggf. mit dem Makro `\tud@headmidvskip@reverse` um den eingefügten Abstand nach oben verschoben werden.

Die Makros werden für die Farbe der Überschriften verwendet.

Dieses Makro wird für die Ausrichtung des Titels bereitgehalten.
Es wird das Schriftelement für die Option \texttt{parttitle} erstellt.

\begin{verbatim}
\texttt{\newkomafont{parttitle}\%
\tud@sec@fontface%
\tud@color{\tud@part@fontcolor}%
\usesizeofkomafont{section}%
}
\end{verbatim}

11.3 Umsetzung des Layouts

Dieser Befehl kümmert sich darum, dass die gewählten Optionen für den Seitenstil auch umgesetzt werden. Dabei wird insbesondere darauf Wert gelegt, dass die Optionen auch im Dokument geändert und wiederhergestellt werden können.

\begin{verbatim}
\texttt{\newcommand*{\tud@layout@process}{%
\texttt{\tud@cmd@store{⟨Befehl⟩}}
\texttt{\tud@cmd@restore{⟨Befehl⟩}}
\texttt{\tud@font@koma@unset{titlehead}}%
\texttt{\tud@font@koma@unset{title}}%
\texttt{\tud@font@koma@unset{subtitle}}%
\texttt{\tud@font@koma@unset{subject}}%
\texttt{\tud@font@koma@unset{author}}%
\texttt{\tud@font@koma@unset{date}}%
\texttt{\tud@font@koma@unset{publishers}}%
\texttt{\tud@cmd@restore{⟨Befehl⟩}}
\texttt{\tud@font@koma@set{titlehead}{\usekomafont{titlepage}}}%
\end{verbatim}
Das gleiche für die Teileseiten.

File h: tudscr-layout.dtx
Es wird die Schriftfarbe, -größe und -art angepasst, je nachdem ob der Titel des Dokumentes auf den Teileseiten auftauchen soll oder eben nicht. Für die Artikel-Klasse gibt es die `parttitle`-Option nicht.
nachgelagerten Gliederungsebenen orientiert für diese Klasse an den Einstellungen für die Teileseiten.

Auch hier werden die Standardbefehle für ein mögliches Deaktivieren der Optionen gesichert und anschließend geändert. Dabei erfolgt die Zuweisung der angepassten Befehle abermals über \let.

Auch hier werden die Standardbefehle für ein mögliches Deaktivieren der Optionen gesichert und anschließend geändert.

File h: tudscr-layout.dtx
Für Poster wird außerdem der passende Seitenstil und die Ausprägung von Kopf und Fuß sowie die Schriftfarbe festgelegt.

\ifcase\tud@cd@num\relax\else % *color
\ifcase\tud@layout@section@num\relax\or\else % *color
\renewcommand*{\tud@section@fontcolor}{HKS41}%
\fi%
\tud@sectioning@set{section}{{%}
\tud@sec@fontface%
\ifstr{\tud@pagecolor}{HKS41}{}{%}
\fi%
\tud@sectioning@set{subsection}{{%}
\tud@sec@fontface%
\ifstr{\tud@pagecolor}{HKS41}{}{%}
\fi%
\tud@sectioning@set{subsubsection}{{%}
\tud@sec@fontface%
\ifstr{\tud@pagecolor}{HKS41}{}{%}
\fi%
\tud@sectioning@set{minisec}{{%}
\tud@sec@fontface%
\ifstr{\tud@pagecolor}{HKS41}{}{%}
\fi%
\tud@sectioning@set{paragraph}{{%}
\tud@sec@fontface%
\ifstr{\tud@pagecolor}{HKS41}{}{%}
\fi%
\tud@sectioning@set{subparagraph}{{%}
\tud@sec@fontface%
\ifstr{\tud@pagecolor}{HKS41}{}{%}
\fi%
\tud@cmd@store{raggedsection}%
\renewcommand*{\raggedsection}{\tud@raggedright}%
\fi%
\fi%
\fi%

\ifcase\tud@cd@num\relax% false
\pagestyle{empty}%
\color{black}%
\else% !false
\pagestyle{empty.tudheadings}%
\ifcase\tud@empty.tudheadings\relax\or\relax \true
\cs@std@lock{tud@head@bar@num}{0}%
\bool@std@lock{tud@foot@colored}{false}%
\color{black}%
\or% litecolor
\cs@std@lock{tud@head@bar@num}{1}%
\bool@std@lock{tud@foot@colored}{false}%
\color{HKS41}%
\or% barcolor
\cs@std@lock{tud@head@bar@num}{2}%
\bool@std@lock{tud@foot@colored}{false}%
\color{HKS41}%
\else% bicolor/color/full
\cs@std@lock{tud@head@bar@num}{3}%
\bool@std@lock{tud@foot@colored}{true}%
\color{HKS41}%
\fi%
\fi%
\fi%
\fi%
\renewcommand*{\raggedsection}{\tud@raggedright}%
\fi

File h: tudscr-layout.dtx 157
Diese beiden Hilfsmakros werden für Titel-Teile und separate Kapitelseite benötigt, um die Fußnoten auf diesen speziellen Seiten in der gleichen Farbe wie den Rest der verwendeten Schriften zu setzen.

\newcommand{\tud@layout@specialpage@set}[1]{% 
  \tud@font@koma@set{footnote}{% 
    \edef\@tempa{% 
      \noexpand\tud@color{\expandafter\noexpand\csname tud@#1@fontcolor\endcsname}% 
    }% 
  }% 
  \tud@cmd@store{footnoterule}% 
  \let\footnoterule\relax% }
\newcommand{\tud@layout@specialpage@unset}{% 
  \tud@font@koma@unset{footnote}% 
  \tud@cmd@restore{footnoterule}% }

Dieses Makro ersetzt den Standardbefehl \parheadstartvskip für die CD-Teilseiten. Damit lassen sich insbesondere die Farbanpassungen für alle Klassen sowie die Option parttitle für \texttt{tudscrreprt} und \texttt{tudscrbook}-Klasse einfach realisieren und gleichzeitig das Zurückschalten auf das Standardverhalten gewährleisten.

Für die \texttt{tudscartcl}-Klasse wird der vertikale Standardabstand beibehalten und lediglich die Farbanpassung hinzugefügt.

Für \texttt{tudscrreprt} und \texttt{tudscrbook}-Klasse werden die Farben gesetzt und die vertikalen Abstände neu definiert.

File h: \texttt{tudscr-layout.dtx}
Sollte die `parttitle`-Option aktiviert sein, so wird der Titel des Dokumentes statt des Teiltitels mit `\tud@parttitle` auf die Seite gesetzt.

Der Titel des Teils wird auf der gleichen Höhe wie der Untertitel auf der Titelseite ausgegeben.

Hiermit wird die Seitenfarbe gesetzt sowie die Schrift für Kopf und Seitenzahl verändert. Die Änderungen sollen dabei lokal bleiben, wofür das obligatorische Argument des Befehls `\tud@cleardoublespecialpage` verwendet wird. Damit wird abhängig von den Einstellungen für die beiden Optionen `cleardoublespecialpage` und `clearcolor` die Rückseite farbig gesetzt.
11.3.1 Präambeln für Teile und Kapitel

\tud@partpreamble

Um die KOMA-Script-Befehle für die Teilepräambel auch für das farbige Layout nutzen zu können, müssen diese leicht angepasst werden. Für die Präambeln wird die Überschrift einer Zusammenfassung ausschließlich zentriert und relativ klein gesetzt. Außerdem wird für eine mögliche Zusammenfassung die titlepage-Option deaktiviert. Aufgerufen wird der Befehl durch die gepatchte Version von \set@@@@preamble.

\newcommand*{\tud@partpreamble}{%
  \ifnum\tud@abstract@level@num>1\relax% sec/chap
    \renewcommand*{\tud@abstract@level@num}{1}\%
  \fi%
  \KOMAoptions{titlepage=false}%
  \addtokomafont{disposition}{\tud@color{\tud@part@fontcolor}}%
  \addtokomafont{dictum}{\tud@color{\tud@part@fontcolor}}%
  \tud@color{\tud@part@fontcolor}%
}

Hiermit wird der unterschiedliche Abstand bei Teil- und Kapitelpräambeln ausgeglichen.

\vskip\dimexpr\topskip+.9\baselineskip\relax%
\@afterindentfalse\afterheading%

\tud@chapterpreamble

Um die KOMA-Script-Befehle für die Kapitelpräambel auch für das farbige Layout nutzen zu können, müssen diese leicht angepasst werden. Für die Präambeln der Kapitel gilt das gleiche wie für die der Teileseiten. Auch hier erfolgt die Ausführung des Befehls durch die gepatchte Version von \set@@@@preamble.

\newcommand*{\tud@chapterpreamble}{%
  \ifnum\tud@abstract@level@num>1\relax% sec/chap
    \renewcommand*{\tud@abstract@level@num}{1}\%
  \fi%
  \KOMAoptions{titlepage=false}%
  \addtokomafont{disposition}{\tud@color{\tud@chapter@fontcolor}}%
  \addtokomafont{dictum}{\tud@color{\tud@chapter@fontcolor}}%
  \tud@color{\tud@chapter@fontcolor}%
}

Die Farb- und Schriftzeinstellungen sind nur für reine Kapitelseiten notwendig.

\if@tud@chapterpage%
  \ifcase\tud@cdfont@num\relax\or% true
    \TUDoptions{cdfont=true}\%
  \or% heavy
    \TUDoptions{cdfont=heavy}\%
  \fi%
  \addtokomafont{disposition}{\tud@color{\tud@chapter@fontcolor}}%
  \addtokomafont{dictum}{\tud@color{\tud@chapter@fontcolor}}%
  \tud@color{\tud@chapter@fontcolor}%
  \@afterindentfalse\afterheading%
\fi%
Die beiden Befehle für Präambeln aus KOMA-Script ignoriieren in ihrer ursprünglichen Form die Einstellungen für \parskip. Dies wird mit diesem Patch beseitigt. Außerdem wird das Makro für die Layouteinstellungen für Teile (\tud@partpreamble) respektive Kapitel (\tud@chapterpreamble) aufgerufen. Für die Verwendung der abstract-Umgebung in den Präambeln wird außerdem der Befehl \phantomsection auf \relax gesetzt, um mit \autoref automatisch generierte Querverweise aus dem Paket hyperref nicht zu verändern. Alternativ zu \let\@parboxrestore\relax müsste innerhalb des Argumentes von \parbox die KOMA-Script-Option parskip neu gesetzt werden.

\CheckCommand\set@@@@preamble[\set@@@@preamble]{%\hbox to\hsize{#4\parbox[\set@@@@preamble]{#3}{#6\par}#5\par}}%
\patchcmd{\set@@@@preamble}{%\hbox to\hsize{#4\parbox[\set@@@@preamble]{#3}{#6\par}#5\par}}{%\def\@tempa####1@####2{%\@nameuse{tud@####1preamble}\KOMAoptionOf\KOMAoption{parskip}{\KOMAclassFileName}{parskip}2\hbox to\hsize{%\let\phantomsection\relax#4\parbox[\set@@@@preamble]{#3}{\@tempa#1#6\par}#5\par}}}{\tud@patch@wrn{set@@@@preamble}}%

11.3.2 Layoutumsetzung speziell für Teile

\CheckCommand\set@@@@preamble[\set@@@@preamble]{%\hbox to\hsize{#4\parbox[\set@@@@preamble]{#3}{#6\par}#5\par}}%
\patchcmd{\set@@@@preamble}{%\hbox to\hsize{#4\parbox[\set@@@@preamble]{#3}{#6\par}#5\par}}{%\def\@tempa####1@####2{%\@nameuse{tud@####1preamble}\KOMAoptionOf\KOMAoption{parskip}{\KOMAclassFileName}{parskip}2\hbox to\hsize{%\let\phantomsection\relax#4\parbox[\set@@@@preamble]{#3}{\@tempa#1#6\par}#5\par}}}{\tud@patch@wrn{set@@@@preamble}}%

Für \tudscrbook- und \tudscrreprt-Klasse ist etwas mehr Arbeit nötig. Für separate Teileseiten muss dafür Sorge getragen werden, dass die Überschrift unabhängig von der Tiefe der
Bei den Teile-Befehlen ohne Gliederungsnummerierung muss zusätzlich der Freiraum vorgehalten werden, damit alle Überschriften auf der gleichen vertikalen Position liegen.
Und nun das Gleiche für KOMA-Script v3.17:

\newcommand*{\tud@@part}{%}
\let{\tud@@part}{\@part}%
\patchcmd{\tud@@part}{\size@partnumber{\partformat}}{\size@partnumber{\tud@makeuppercase{\partformat}}}%
\{\}{\tud@patch@wrn{scr@@startspart}}%
\langle /article | poster \rangle
\newcommand*{\tud@@spart}{%}
\let{\tud@@spart}{\@spart}%
\patchcmd{\tud@@spart}{\normalfont\sectfont
nobreak\size@part}{\normalfont\sectfont
nobreak%}
\if@tud@parttitle{\else%}
\size@partnumber{\vphantom{\partname}}%
\tud@headmidvskip@normal%\nobreak%\fi%
\{\}{\tud@patch@wrn{scr@@startspart}}%
\langle /book | report \rangle
\{\}{\tud@patch@wrn{scr@@startspart}}%
11.3.3 Layoutumsetzung speziell für Kapitel

\chapterheadstartvskip
\chapterheadstartvskip wird für die Position der Kapitelüberschriften im Corporate Design angepasst. Mit der Option headingsvskip kann diese vertikal durch den Benutzer verschoben werden. Außerdem werden die benötigten Farben für die jeweils gewählte Option ausgewählt.

\newcommand*{\chapterheadstartvskip}{% \renewcommand*{\chapter@fontcolor}{}% \ifcase{\chapter@num}\relax\or\else% *color \renewcommand*{\chapter@fontcolor}{HKS41}% \fi% \vspace*{\dimexpr{\len@areavskip}+{\len@areaheadvskip}\relax} \if@chapterpage% \TUD@deprecated@lengthcs{pageheadingsvskip}% \vspace*{\dim@pageheadingsvskip}% \else% \TUD@deprecated@lengthcs{headingsvskip}% \vspace*{\dim@headingsvskip}% \fi%}

\chapterheadmidvskip
Für das Präfix des Kapitels bei der gegebenenfalls aktivierten Option chapterprefix ist einiges an Anpassungen notwendig, um die eigentliche Überschrift auf der gewünschten Position erscheinen zu lassen.

Zunächst muss die Überschrift um die Höhe des Präfixes und ggf. der Länge \parskip nach oben verschoben werden. Der Befehl \chapterformat enthält das Präfix selbst. Damit dieses in Majuskeln erscheint, wird \chapterformat im Argument des Makros \tud@makeuppercase verwendet. Zuletzt wird der Abstand zwischen dem Präfix und dem Kapiteltitel auf \tud@headmidvskip gesetzt. Somit liegen die Kapitelüberschriften für alle Einstellmöglichkeiten der parskip-Option immer auf exakt der selben Höhe.

Zuerst die Patches für KOMA-Script v3.18:

\newcommand*{\tud@makechapterhead}{% \ifcsdef{scr@v3.18}{\tempswatatrue}{\tempswatalse}\let{\tud@makechapterhead}{\scr@makechapterhead}\patchcmd{\tud@makechapterhead}{\csname scr@#1@format\endcsname}{\setlength{\@tempskipa}{\csname scr@#1@innerskip\endcsname}\chapterheadmidvskip}{\tempskipareverse}{\tempskipareverse}{{\tempskiparenormal}{\tempskiparenormal}}%}

\tud@makechapterhead
\scr@makechapterhead
\@makechapterhead
File h: tudscr-layout.dtx
Und nun das Gleiche für KOMA-Script v3.17:

Diese Makros dienen zur Behandlung der Besonderheiten von Kapiteln. Dies gilt sowohl für normale Kapitelüberschriften als auch für separate Kapitelseiten (Option chapterpage). Eingefügt werden sie für die Gliederungsbefehle der Kapitel (\chapter und \addchap) durch \sectioning@case.

Für den zweispaltigen Satz und aktiverer chapterpage-Option muss für die Kapitelseiten temporär auf einseitigen Satz umgestellt werden, damit alle Befehle und Einstellungen beim Erzeugen des jeweiligen Kapitels zum Tragen kommen. Soll der Fehler behoben werden, dass bei Kapitelüberschriften im zweispaltigen Satz der Abstand zwischen Überschrift und Textkörper nicht stimmt, sei außerdem das Paket twocolfix empfohlen.

Unabhängig von der Nutzung einer Kapitelseite, wird die Einstellung der Option cdchapter für die Gestalt des TUD-Kopfes beachtet, falls dieser zum Einsatz kommt.

Nach dem Setzen der Kapitelüberschrift selbst werden die Farbe für den Hintergrund, die Seitenzahl und den ggf. verwendeten TUD-Kopf angewendet. Danach wird die Seite beendet. Sollte zweiseitiger Satz, immer rechts öffnende Kapitel und die Option chapterpage gewählt sein, so wird für ein konsistentes Layout für Überschriften auf Einzelseiten die Rückseite abhängig von der Option clearcolor gegebenenfalls farbig gesetzt und die Option cleardoublespecialpage bezüglich des Seitenstils der Rückseite beachtet. Umgesetzt wird dies mit \tud@cleardoublespecialpage.
Außerdem wird für den Zweispaltensatz auf normales Verhalten zurückgestellt.

Sollten keine reinen Kapitelseiten verwendet werden, wird die Einstellung für die Gestalt des Seitenkopfes im restlichen Dokument \tud@head@bar@store gesichert und \tud@head@bar@restore definiert, womit der gesicherte Wert beim Ausführen der Ebene tudheadings.last für die darauffolgenden Seite wiederhergestellt wird.

Im Kompatibilitätsmodus muss genügend zusätzlicher Freiraum für ein ggf. auszugebendes DRESDEN-concept-Logo gelassen werden.

Unabhängig von der Option chapterpage wird dein zweispaltigen Layout der Einzug des ersten Absatzes verhindert.
Die Warnung für den Fall, dass die Option clearcolor ohne open=right und twoside verwendet wird.

\newcommand*{tud@clearcolor@wrn}{%\ClassWarning{\TUD@Class@Name}{Option 'clearcolor' is only available together with options 'twoside' and 'open=right'\MessageBreak}%\global\let\tud@clearcolor@wrn\relax%}

Damit ist das Intermezzo für die beiden Klassen \texttt{tudscrbook} und \texttt{tudscrrreport} beendet.

11.3.4 Erzwungene Majuskeln für Überschriften

Das Corporate Design verlangt, dass sämtliche Auszeichnungen, die in \textit{DIN BOLD} gesetzt werden, groß zu schreiben sind. Das bedeutet, dass alle Überschriften automatisch in Majuskeln und in der gewünschten Schriftart erscheinen müssen. Dies erfordert eine Anpassung der Gliederungsbefehle, ohne deren normales Verhalten zu beeinflussen.

Dazu werden die benötigten Hilfsmakros initialisiert. Die Gliederungsbefehle selbst werden innerhalb des Befehles \texttt{tud@sectioning@set} durch \texttt{tud@cmd@store} gesichert und anschließend mit \texttt{tud@sectioning@case} neu definiert. Dabei wird mit \texttt{SecDef} unterschieden, ob die Sternversion des Gliederungsbefehls durch den Anwender verwendet wird und ggf. das Hilfsmakro \texttt{tud@s(Gliederungsbefehl)} aufgerufen. Andernfalls wird \texttt{tud@(Gliederungsbefehl)} verwendet.

Alle für \texttt{TUD-Script} angepassten Gliederungsbefehle von \texttt{\part} bis \texttt{\subsubsection} werden seit Version v2.02 ausnahmslos mit \texttt{SecDef} erzeugt.

\newcommand*{tud@part}{\tud@part\langle*book | report\rangle}\newcommand*{tud@spart}{\tud@spart\langle/\book | \report\rangle}\newcommand*{tud@chapter}{\tud@chapter}\newcommand*{tud@schapter}{\tud@schapter}\newcommand*{tud@section}{\tud@section}\newcommand*{tud@ssection}{\tud@ssection}\newcommand*{tud@subsection}{\tud@subsection}\newcommand*{tud@ssubsection}{\tud@ssubsection}\newcommand*{tud@subsubsection}{\tud@subsubsection}\newcommand*{tud@ssubsubsection}{\tud@ssubsubsection}\newcommand*{tud@sectioning@set}{%\tud@sectioning@set\tud@sectioning@case\tud@sectioning@reset\tud@sectioning@uppercase%}

Das Makro \texttt{tud@sectioning@set} sichert und überschreibt die gegebenen Befehle der einzelnen Gliederungsebenen und fügt diesen die zusätzlich angegebene Schriftdefinition mit dem Makro \texttt{tud@font@koma@set} zu.

\newcommand*{tud@sectioning@set}[2]{%\tud@cmd@store[#1]}%
Für die Überschriften, welche mit `\minisec` gesetzt werden, genügt jedoch ein vereinfachtes Vorgehen.

```
\ifstr{#1}{minisec}{%
\renewcommand\minisec[1]{\@@tud@minisec{\tud@makeuppercase{##1}}}%
}{}%
\tud@sectioning@case{#1}%
\tud@font@koma@set{#1}{#2}%
```

Hier erfolgt die Anpassung und Formatierung aller anderen Gliederungsbefehle.

```
Als erstes wird unterschieden, ob die normale oder aber die nicht-nummerierte sprich die Sternversion gefordert ist. Dies passiert mit `\SecDef`. Für die Teileebene werden beispielsweise entweder `\tud@part` oder `\tud@spart` aufgerufen. Dabei wird durch `\SecDef` für `\tud@part` das obligatorische in das optionale Argument kopiert, falls keines angegeben wurde.

```
\csedef{#1}{%
\noexpand\SecDef%
\expandafter\expandafter\expandafter\noexpand\expandafter\csname tud@#1\endcsname%
\expandafter\expandafter\expandafter\noexpand\expandafter\csname tud@s#1\endcsname%
}%
```

Da die Großschreibung der Überschriften nur beim Setzen der Überschrift selbst, nicht aber beispielsweise im Inhaltsverzeichnis oder aber in der Kopfzeile unerwünscht erfolgen soll, werden die Gliederungsbefehle angepasst. Dabei müssen alle Varianten, also auch die mit Stern und optionalen Argument beachtet werden.

```
\newcommand*{\tud@sectioning@case}[1]{%
\tud@font@koma@set{#1}{#2}%
```

Für die Version mit Stern gibt es kein optionales Argument. Dementsprechend wird hier lediglich das obligatorische Argument in Großbuchstaben gesetzt.

```
\csedef{tud@s#1}{%
\expandafter\expandafter\expandafter\noexpand\expandafter\csname @@tud@#1\endcsname*%
\noexpand	ud@sectioning@uppercase{##1}%
}%
```

File h: tudscr-layout.dtx
Die Änderungen durch \tud@chapter@pre und \tud@chapter@app sollen bei der Erzeugung eines Kapitels greifen. Damit die originalen Befehle nicht beeinflusst werden, erfolgt die Manipulation der bereits adaptierten Befehle für Kapitel.

\begin{verbatim}
1067 \ifstr{#1}{chapter}\{
1068 \def\@tempa{,s}\%
1069 \for\@tempb:=\@tempa\do{\%
1070 \expandafter\pretocmd\expandafter{\csname tud@\@tempb#1\endcsname}{{
1071 \tud@chapter@pre}\%
1072 }{\tud@patch@wrn{tud@sectioning@case}}%
1073 \expandafter\apptocmd\expandafter{\csname tud@\@tempb#1\endcsname}{{
1074 \tud@chapter@app}\%
1075 }{\tud@patch@wrn{tud@sectioning@case}}%
1076 }\%
1077\%
1078\%
1079\end{verbatim}

Mit dem Makro \tud@sectioning@reset können die manipulierten Gliederungsbefehle wieder auf ihr ursprüngliches Verhalten zurückgesetzt werden.

\begin{verbatim}
1081 \newcommand*{\tud@sectioning@reset}[2]{\%
1082 \tud@cmd@restore{2}\%
1083 \ifblank{1}{\tud@cmd@restore{1}\%}
1084 \tud@font@koma@unset{2}\%
1085 \%}
\end{verbatim}

Der Befehl \tud@sectioning@uppercase ist für das Setzen der Gliederungsüberschriften in Majuskeln verantwortlich. Außerdem wird ggf. die Option footnotes=symbolheadings umgesetzt.

\begin{verbatim}
1086 \robustify{\footnote}\%
1087 \newcommand*{\tud@sectioning@uppercase}[1]{\%
1088 \begingroup\%
1089 \if@tud@symbolheadings\%
1090 \let\c@footnote\c@symbolheadings\%
1091 \renewcommand{\thefootnote}{\fnsymbol{symbolheadings}}\%
1092 \fi\%
1093 \tud@makeuppercase{1}\%
1094 \endgroup\%
1095 \%
\end{verbatim}
12 Titelei für die TUD-Script-Klassen

Zusätzlich zu den KOMA-Script-Varianten werde für die TUD-Script-Klassen spezielle, im Corporate Design der Technischen Universität Dresden gehaltene Ausführungen von Titelseite und Titelkopf bereitgestellt. Zusätzlich gibt es auch eine Umschlagsseite (Cover), welche in beiden Gestaltungsvarianten genutzt werden kann.


12.1 Optionen und Schriftelemente für die Titelei

Die alte tudbook-Klasse hat den Typ der Arbeit in der Betreffzeile des Titels gesetzt. Um dieses Verhalten nachbilden zu können, wird diese Option zur Verfügung gestellt.

```latex
\if@tud@subjectthesis
\if@tud@subjectthesis@locked
\TUD@key@lock{subjectthesis}{true}{% 
  \TUD@set@ifkey@lock{subjectthesis}{@tud@subjectthesis}{#1}{% 
}
\fi
\fi
```

Für Poster gibt es keine separate Titelseite sondern nur einen Titelkopf. Falls die falsche Einstellung gewählt ist, wird diese deaktiviert und eine Warnung ausgegeben.

```latex
\DefineFamilyMember{KOMA}
\DefineFamilyKey{KOMA}{titlepage}{true}{% 
  \TUD@set@numkey{titlepage}{@tempa}{\TUD@bool@numkey,{firstiscover}{2}}{#1}{% 
  \ifx\FamilyKeyState\FamilyKeyProcessed{% 
    \AtEndOfFamilyOptions{\tud@titlepage@deactivate}{% 
  \fi}% 
}
\newcommand*{\tud@titlepage@deactivate}{% 
  \if@titlepage\KOMAoptions{titlepage=false}{% 
    \ClassWarning{\TUD@Class@Name}{The class '{\TUD@Class@Name}' can't be used with activated titlepage option. It will be enforced to 'titlepage=false'.}{% 
  \fi}% 
\fi
\fi
\fi}% 
\renewcommand*{\tud@titlepage@deactivate}{% 
  \if@titlepage\KOMAoptions{titlepage=false}{% 
  \fi}% 
\fi
\AtBeginDocument{\tud@titlepage@deactivate}{% 
```

File: tudscr-title.dtx 170
Diese Font-Elemente werden auf der Titelseite verwendet. \thesis, um den Typ der Arbeit und ggf. den zu erlangenden akademischen Grad zu setzen. Allerdings kommt Letzteres nur zum Tragen, wenn die Option \subjectthesis deaktiviert ist. Es wird die gleiche Schrift wie für den Titel verwendet, allerdings etwas kleiner. \titlepage für alle Zusatzinformationen wie Autor, Datum etc.

\begin{verbatim}
26 \{book | report | article
27 \newkomafont{thesis}{%}
28 \usekomafont{title}{%}
29 \tud@thesis@fontsize%
30 \}
31 \{/book | report | article
32 \newkomafont{titlepage}{%}
33 \usekomafont{subject}{%}
34 \mdseries%
35 \large%
36 \tud@raggedright%
37 \}
\end{verbatim}

12.2 Titelseite und Titelkopf

12.2.1 Parameter für Titelseite und Titelkopf

\thanks \footnotemark
Die Befehle \thanks und \footnotemark werden mit \robustify behandelt, um die Makros vor dem Expandieren zu schützen. Dies ist z. B. für \tud@multiple@split aber auch für weitere Felder auf dem Titel nötig.

\begin{verbatim}
38 \robustify{\thanks}
39 \robustify{\footnotemark}
\end{verbatim}

\maketitle \cdtitle (Param.) \pagenumber (Param.)
\begin{verbatim}
\maketitle
\cdtitle \pagenumber
\end{verbatim}

Bei der entsprechend gewählten Layoutoption, wird eine neue Titelseite im Corporate Design gesetzt. Damit auf das Standardverhalten von KOMA-Script zurückgestellt werden kann, wird der ursprüngliche \maketitle-Befehl bei der Abarbeitung der Layoutoptionen gesichert. (siehe \tud@layout@process) und kann ggf. wiederhergestellt bzw. mit \tud@cmd@use verwendet werden. Ist das Layout des Corporate Designs gefordert wird entweder mit \tud@make@titlepage eine Titelseite gesetzt – wobei sich diese in ihrer Struktur am originalen \maketitle orientiert – oder das Titelkopflayout wird angepasst.

\begin{verbatim}
40 \newcommand*{\tud@title@pagenumber{}}
41 \let\tud@title@pagenumber\relax
42 \newcommand*{\tud@make@titlepage[1][]{%}
43 \begingroup%
\end{verbatim}

Das optionale Argument wird für das Setzen der Parameter verwendet.

\begin{verbatim}
44 \TUD@parameter@set{title}{#1}{%}
\end{verbatim}

Ist das Corporate Design für den Titel deaktiviert, wird mit \tud@cmd@use der originale KOMA-Script-Befehl \maketitle aufgerufen.

\begin{verbatim}
45 \ifcase\tud@layout@title@num\relax\false
46 \ifx\tud@title@pagenumber\relax\gdef\tud@title@pagenumber{1}{\fi%
47 \tud@cmd@use[\maketitle]{\tud@title@pagenumber}%
48 \tud@if@tudheadings{\currentpagestyle}{%
49 \ifstr{\tud@pagestyle}{empty}{\thispagestyle{empty.tudheadings}}{%}
50 }{%}
\end{verbatim}
Für die TUD-Script-Klassen gibt es für die Titelseite sowie den Titelkopf separate Befehle. Sollte keine explizite Auswahl der Schriften erfolgt sein, so werden die Corporate Design-Schriften aktiviert.

Abschließend wird dafür Sorge getragen, dass bei einem aktivierten Seitenstil aus der tudheadings-Reihe ebenfalls der Titel in diesem gesetzt wird, auch wenn das Layout des Corporate Designs selber nicht aktiv ist.

Die einzelnen Parameter dienen zur Einstellung der Titelseite und entsprechen in ihrem Verhalten den gleichnamigen Klassenoptionen.

Mit \TUD@parameter@ps@def werden alle verfügbaren Parameter für die Kopf- und Fußzeilen definiert.

Für den Fall, dass ein Wert nicht in der Schlüssel-Wert-Notation angegeben wurde, wird versucht, das gegebene Argument als Seitenzahl zu verwenden.
12.2.2 Die Definition der Titelseite

Bei der entsprechend gewählten Layoutoption, wird eine neue Titelseite im Corporate Design gesetzt. Damit auf das Standardverhalten von KOMA-Script zurückgestellt werden kann, wird der standardmäßige \maketitle-Befehl nur wenn notwendig überschrieben (siehe \tud@layout@process) und kann ggf. wiederhergestellt werden. Der Befehl selbst orientiert sich dabei in seiner Struktur an dem originalen \maketitle. Für die ggf. farbige Ausgabe der Titelrückseite wird das Hilfsmakro \tud@titleback definiert. Eine Seitenzahl für den Titel kann mit \tud@title@pagenumber angegeben werden.

Es wird durch die KOMA-Script-Option titlepage unterschieden, ob eine Titelseite oder aber ein Titelkopf gesetzt werden soll. In Abhängigkeit davon werden Farben für die Schriften und den Seitenhintergrund festgelegt. Hierfür wird die entsprechende Layoutoption ausgewertet und entsprechend dieser die Werte belegt.

Das Paket hyperref definiert im Normalfall die Fußnoten so um, dass es anklickbare Links erzeugt. Damit dies auf der Titelseite zu ungewollten Effekten führt, werden die Hyperlinks hier deaktiviert. Leider macht dies hyperref direkt beim Laden. Da die TUD-Script-Klassen aber nun mal zwei unterschiedliche Titel bereitstellen und die Möglichkeit bestehen soll, dass
Layout per später Optionswahl noch zu ändern, muss der interne Code aus \hyperref hier noch einmal für den TUD-Titel aufgerufen werden.

\tud@x@hyperref@realfootnotes%

Die Option für eine Umschlagseite titlepage=firstiscover wird im Gegensatz zu den \KOMA-Script-Klassen nur für einen optionalen Schmutztitel \extratitle und nicht für die eigentliche Titelseite unterstützt. Für dieses Unterfangen ist bei TUD-Script der Befehl \makecover mit den dazugehörigen Parametern zu nutzen.

\iffalse@extratitle\empty\fi\if@titlepageiscoverpage\\ClassWarning{\TUD@Class@Name}{Using option 'titlepage=firstiscover' is only possible with an extra title. Alternatively you can create a cover with '\string\makecover'}\fi\else

Der Teil ist aus den \KOMA-Script-Klassen kopiert...

\iffalse@titlepageiscoverpage\edeftitlepage@restore\noexpand\endgroup\noexpand\global\noexpand@colht\the@colht\noexpand\global\noexpand@colroom\the@colroom\noexpand\global\noexpand\vsize\the\vsize\noexpand\global\noexpand@titlepageiscoverpagefalse\noexpand\let\noexpand\titlepage@restore\relax\}\begingroup\topmargin=\dimexpr\coverpagetopmargin-1\textwidth\relax\oddsidemargin=\dimexpr\coverpageleftmargin-1\textwidth\relax\evensidemargin=\dimexpr\coverpageleftmargin-1\textwidth\relax\textwidth=\dimexpr\tud@len@layoutwidth-\coverpageleftmargin\relax\textheight=\dimexpr\tud@len@layoutheight-\coverpagetopmargin\relax\headheight=0pt\headsep=0pt\footskip=\baselineskip\@colht=\textheight\@colroom=\textheight\vsize=\textheight\columnwidth=\textwidth\hsize=\columnwidth\linewidth=\hsize\else\let\titlepage@restore\relax\fi\fi\fi

Das ganze Layout wird vertikal so verschoben, dass der Titel auf der gleichen Höhe steht wie die Überschriften auf Teil- und Kapitelseiten.

\vspace*{\%

File i: tudscr-title.dtx
Damit die Farbeinstellungen der Schriften lokal begrenzt bleiben, wird dieser Teil des Titels in einer Gruppe \begin{group} gesetzt.

Da die Überschrift auf einer festen Position liegen soll, muss von allem, was über der Überschrift positioniert werden soll, die Höhe bestimmt werden, um einen entsprechenden Offset einfügen zu können. Dafür werden zwei temporäre Standardboxen verwendet. In diesen wird der gewünschte Inhalt für die beiden Felder \@titlehead bzw. \@subject abgelegt und die sich daraus kumulierende Höhe in \@tempdima gespeichert. Vor dem Setzen der Felder kann nun der insgesamt notwendige vertikale Freiraum vor dem eigentlichen Titel für \@titlehead und \@subject abgezogen werden. Anschließend werden die beiden Boxen ausgegeben und der Aufbau der Titelseite kann beginnen.

Erkennen, ob in der Betreffzeile eine Abschlussarbeit ausgegeben werden soll und kann.

Nachdem bereits \@titlehead und \@subject gesetzt worden, folgen nun die weiteren Felder wie vom Corporate Design – mehr oder weniger – vorgegeben auf der Titelseite in unterschiedlichen Größen und Schriftstärken.

Der Untertitel wird ggf. in der gleichen Schrift gesetzt wie der Titel, allerdings etwas kleiner.
Ausgabe eines Autors oder mehrerer Autoren ggf. mit Zusatzinformationen.

Die Titelseite unterscheidet sich ein wenig, abhängig davon ob eine eine Abschlussarbeit gesetzt wird oder nicht. Falls dies nicht der Fall ist, dann wird das angegebene bzw. das heutige Datum und ggf. die Ergänzung ausgegeben.

Ist eine Abschlussarbeit angegeben, so wird zusätzlich unterschieden, ob diese bereits – gesteuert durch die Option `subjectthesis` – in der Betreffzeile angegeben wurde oder aber separat und in deutlicherer Form auf dem Titelblatt ausgegeben werden soll. Sowohl Arbeitstyp als auf akademischer Grad werden dabei im Verhältnis 1:2 zwischen allen weiteren Angaben auf dem Titel vertikal ausgerichtet.

Ebenfalls wird in Abhängigkeit von der Option `subjectthesis` der angestrebte akademische Grad ausgegeben.

Wenn Betreuer angegeben wurden, dann werden diese auf dem Titel ausgegeben. Um diese Titelseite auch für Dissertationen verwenden zu können, ist ein Feld für Gutachter vorgesehen. Außerdem wird für Promotionsschriften ein Feld für Fachreferenten bereitgehalten. Das
Format der Ausgabe ist dabei identisch. Darauf folgend wird der betreuende Hochschullehrer ausgegeben.

Bei einer Abschlussarbeit gehört normalerweise auch ein Abgabedatum auf den Titel. Sollte keines angegeben sein, so erscheint eine Warnung. Es wird an Stelle dessen das angegebene Enddatum oder aber das heutige Datum verwendet. Durch das Paket `tudscrsupervisor` wird zusätzlich das Feld `@duedate` definiert, welches hier ggf. für auf ein nutzbares Datum geprüft wird.

Zusätzlich kann noch ein Datum der mündlichen Verteidigung angegeben werden.

Zu guter Letzt kann das Standard-Verlagsfeld befüllt werden.

Die Rückseite des Titels wird – wie bei KOMA-Script – bloß bei zweiseitigem Satz ausgegeben. Für diese Klassen wird aufgrund der möglicherweise farbigen Rückseite das Hilfsmakro `\tud@titleback` benötigt.
Es werden Schriftart, die Schriftstärke, sowie die Farbe der Seite und des Kopfes und die Linienstärke des Querbalkens und die DRESDEN-concept-Logos aus den Einstellungen übernommen.

Im Kompatibilitätsmodus für die Version v2.02 wird im Zweifelsfall die Seite für den Fuß mit DRESDEN-concept-Logo vergrößert.

Bei der Ausgabe des Covers werden lediglich die Farb- und Schriftdefinitionen benötigt, die Seite kann danach beendet werden.

Beim Titel selbst muss noch ausgewertet werden, wie die Rückseite zu gestalten ist. Sollte als Klassenoption das strikte Beginnen aller Teile oder Kapitel auf einer rechten Seite aktiviert sein (open=right) und zusätzlich bei farbigem Corporate Design auch die Rückseiten farbig geleert werden, so erfolgt dies auch für die Titelseite. Die Klasse \texttt{tudscartcl} setzt lediglich die Titelseite farbig, die Rückseite niemals.
Die Danksagung ist identisch zu KOMA-Script.

Das sind die internen Makros für einen Titelkopf (titlepage=no), welche sich stark an den originalen KOMA-Script-Befehlen orientieren.

Dieser Teil ist mehr oder weniger funktional identisch zu KOMA-Script.
Hier wird überprüft, ob das Dokument zweispaltig gesetzt wird und zusätzlich das Paket `multicol` geladen wurde. Ist letzteres nicht der Fall, wird der Titel einspaltig ausgegeben und anschließend der im Zweispaltensatz begonnen.

Hier erfolgt die Ausgabe des Titelkopfes.

Ähnlich wie bei der Titelseite wird auch beim Titelkopf die Inhalte von `@titlehead` und `@subject` so nach oben verschoben, dass der Titel selbst auf der gewünschten Höhe landet.
Erkennen, ob in der Betreffzeile eine Abschlussarbeit ausgegeben werden soll und kann. Bei einem Titelkopf ist nur diese Variante möglich.

\begin{verbatim}
387 (*book | report | article)
388 \TUDoptions{subjectthesis=true}\
389 /book | report | article
390 \ifs\@subject\@empty\else
391 \sbox\@ne{\vbox{{\usekomafont{subject}\{\@subject\par\null}}}%}
392 \addtolength\@tempdima{\dimexpr\ht\@ne+\dp\@ne+p\@relax}%
393 \fi%
394 (*book | report | article)
396 \vspace*{-\@tempdima}%
397 (*book | report | article)
398 \ifs\@titlehead\@empty\else\usebox\z@\par\fi%
399 \ifs\@subject\@empty\else\usebox\@ne\par\fi%
400 \{\usekomafont{title}{\tud@title@uppercase{\@title}\par}%%
401 \ifs\@subtitle\@empty\else
402 \vskip 1ex\@plus.5ex\@minus.5ex%
403 \{\usekomafont{subtitle}{\tud@title@uppercase{\@subtitle}\par}%%
404 \fi%
405 \vskip 2ex\@plus1ex\@minus1ex%
\end{verbatim}

Ausgabe eines Autors oder mehrerer Autoren ggf. mit Zusatzinformationen.

\begin{verbatim}
406 (*book | report | article)
407 \vskip 2\bigskipamount%
408 \let\tud@split@author@do\tud@split@author@title%
409 \tud@multiple@split{@author}\par%
\end{verbatim}

Ausgabe von Datum, Verlag und Danksagung wie im Original.

\begin{verbatim}
410 \ifs\@date\@empty\else%
411 \vskip 2ex\@plus1ex\@minus1ex%
412 \{\usekomafont{date}{\tud@date@print}\par}%%
413 \fi%
414 \ifs\@publishers\@empty\else%
415 \vskip 2ex\@plus1ex\@minus1ex%
416 \{\usekomafont{publishers}{\@publishers}\par}%%
417 \fi%
418 \ifs\@dedication\@empty\else%
419 \vskip 2ex\@plus1ex\@minus1ex%
420 \{\centering\usekomafont{dedication}{\@dedication}\par}%%
421 \fi%
\end{verbatim}

Falls Felder genutzt wurden, die nur auf der Titelseite, nicht aber im Titelkopf ausgegeben werden, dann wird eine Warnung ausgegeben.

\begin{verbatim}
422 \ifs\@graduation\@empty\else\tud@title@wrn{graduation}\fi%
423 \ifs\@referee\@empty\else\tud@title@wrn{referee}\fi%
424 \ifs\@advisor\@empty\else\tud@title@wrn{advisor}\fi%
425 \ifs\@supervisor\@empty\else\tud@title@wrn{supervisor}\fi%
426 \ifs\@professor\@empty\else\tud@title@wrn{professor}\fi%
\end{verbatim}

Am Ende wird noch etwas vertikaler Abstand eingefügt.

\begin{verbatim}
427 \vskip 2\bigskipamount%
428 /book | report | article
429 \edefvoid{\multicolsep}{}
\end{verbatim}
Wird das DRESDEN-concept-Logo im Kompatibilitätsmodus für die Version v2.02 gesetzt, muss dafür am unteren Rand entsprechend Platz gelassen werden.

\[\text{12.2.4 Hilfsmakros für die Ausgabe}\]

\texttt{\textbackslash tud@title@uppercase} Für das Setzen einzelner Felder in Majuskeln werden diese beiden Befehle definiert. Bei der Verwendung des Befehls \texttt{\textbackslash thanks} auf der Titelseite soll sichergestellt werden, dass der Inhalt selbst nicht in Majuskeln gesetzt wird.

\[\text{\newcommand*{\textbackslash tud@title@uppercase[1]{%}\let\@tempa\relax%\begingroup%\let\\newline%Der Befehl \texttt{\textbackslash thanks} wird so umdefiniert, dass der Inhalt in ein temporäres Makro gespeichert wird. Danach wird durch das Setzen einer temporären Box der Befehl \texttt{\textbackslash thanks} ausgeführt und der Inhalt gesichert. Für die Ausführung von \texttt{\textbackslash tud@makeuppercase} wird das Argument von \texttt{\textbackslash thanks} ignoriert. Nachdem das gewünschte Feld in Majuskeln gesetzt wurde, wird die gespeicherte Fußnote ausgegeben.

\[\text{\def{\textbackslash thanks##1{\gdef{\@tempa{##1}}}}%\let\footnote\thanks%\sbox\z@{#1}%\let\thanks\@gobble%\let\footnote\@gobble%\tud@makeuppercase{#1}%\endgroup%\ifx\@tempa\relax\else\thanks\@tempa\fi%}

\texttt{\textbackslash tud@thesis@tosubject} Bei der Verwendung der Option subjectthesis wird zum einen geprüft, ob das Feld \texttt{\textbackslash thesis} gesetzt ist und zum anderen, ob das Feld \texttt{\textbackslash subject} auch leer ist.

\[\text{\newcommand*{\textbackslash tud@thesis@tosubject{%}\if@tud@subjectthesis\ifx\@thesis\@empty\else%\ifx\@subject\@empty\else%\ClassWarning{\TUD@Class@Name}{\text{The option ‘subjectthesis’ was activated, so the field ‘subject’ will be overwritten with the given thesis}}%\else%\fi%\let\@subject\@thesis%\fi%\fi%}

File i: tudscr-title.dtx 182
Dieser Befehl dient zur getrennten Ausgabe von mehreren, mit dem Makro \and separierten Einträgen in den Feldern \@supervisor, \@referee und \@advisor.

Dieses Makro wird für das Trennzeichen auf der Titelseite genutzt.

Das Makro \tud@split@author@title dient der formatierten Ausgabe aller Informationen für mehrere Autoren auf der Titelseite.

Zu Beginn werden alle Felder, welche innerhalb von \@author angegeben sind, gesichert.

Um die Abstände definiert zu setzen, wird \tud@newline@title verwendet. Sobald auch nur eine zusätzliche Information ausgegeben werden soll, wird ein etwas kleiner Abstand eingefügt. Anschließend wird jede Information einfach in einer neuen Zeile ausgegeben.

Die Felder \@author und ggf. \@authormore werden sowohl auf der Titelseite als auch im Titelkopf jedoch nicht auf der Umschlagseite ausgegeben.


File i: tudscr-title.dtx
Zunächst werden die zuvor bestehenden Feldinhalte wiederhergestellt. Danach wird durch den Aufruf von \tud@multiple@split wird das übergebene \langle Feld \rangle sukzessive abgearbeitet.

Bei der Nutzung eines Titelkopfes wird bloß eine reduzierte Menge an Informationen ausgegeben. Darauf wird der Anwender ggf. hingewiesen.


Der Befehl \maketitleonecolumn [(Parameter)] [(Inhalt)] kann bei zweispaltigem Satz dazu genutzt werden, einen einspaltigen Titel zu bekommen. Beim der Verwendung einer Titelseite wird lediglich der Titel sowie anschließend ggf. \langle Inhalt \rangle einspaltig ausgegeben. Beim Satz eines Titelkopfes wird der Befehl \tud@title@twocolumn zur Ausgabe genutzt.

\(^9\)http://www.komascript.de/node/1699
Hier erfolgt die Unterscheidung der Ausgabe, abhängig von der Option \texttt{titlepage}.

\begin{verbatim}
\if@twocolumn%
\if@titlepage%
tud@maketitle[#1]%
onecolumn{#2}\twocolumn%
\else%
⟨/book | report | article⟩
tud@title@twocolumn{\tud@maketitle[#1]{#2}}%
\fi
⟨/book | report | article⟩
\else%
\ClassError{\TUD@Class@Name}{\string\maketitleonecolumn' can only be used in twocolumn mode}{}
\fi
\end{verbatim}

Es folgt die Verarbeitung des Inhaltes des Titelkopfes, welcher einspaltig gesetzt werden soll. Damit sich dieser auch über mehrere Seiten erstrecken kann, wird mit einer Box gearbeitet.

\begin{verbatim}
\newsavebox{tud@title@box}
\newcommand{tud@title@twocolumn}[1]{%
\newsavebox{tud@title@box}
\begin{group}
\let\clearpage\relax
\let\newpage\relax
\end{group}
\setlength{\@tempdima}{\textheight}\
\tud@if@v@lower{2.03}{%
\tud@ddc@check%
\ifcase\@tempb\relax\else\tud@ddc@foot@num!=false
\setlength{\@tempdima}{\dimexpr\textheight-\tud@len@ddcdiff\relax}%
\if\tempswatrue\%
\fi\}
\fi%
\setlength{\@tempdima}{\dimexpr\textheight-\tud@len@ddcdiff\relax}%
\if\tempswatrue\%
\fi\}
\fi%
\end{verbatim}

Zunächst wird auf einseitigen Satz umgeschaltet. Der zu schreibende Inhalt wird in einer Box gespeichert. Während des Zusammenbaus der Box sind keine expliziten \texttt{newpage}- oder \texttt{clearpage}-Befehle erlaubt.

\begin{verbatim}
\onecolumn%
\begingroup
\let\clearpage\relax
\let\newpage\relax
\end{group}
\setlength{\@tempdima}{\textheight}\
\tud@if@v@lower{2.03}{%
\tud@ddc@check%
\ifcase\@tempb\relax\else\tud@ddc@foot@num!=false
\setlength{\@tempdima}{\dimexpr\textheight-\tud@len@ddcdiff\relax}%
\if\tempswatrue\%
\fi\}
\fi%
\end{verbatim}

Sollte der Inhalt der Box größer als die verfügbare Texthöhe des Satzspiegels sein, muss aufgrund der verwendeten Box ein zusätzlicher vertikaler Abstand von \texttt{.9\baselineskip} zur korrekten Positionierung der Überschrift eingefügt werden.

\begin{verbatim}
\setlength{\@tempdima}{\textheight}\
\tud@if@v@lower{2.03}{%
\tud@ddc@check%
\ifcase\@tempb\relax\else\tud@ddc@foot@num!=false
\setlength{\@tempdima}{\dimexpr\textheight-\tud@len@ddcdiff\relax}%
\if\tempswatrue\%
\fi\}
\fi%
\end{verbatim}

Als erstes muss geprüft werden, ob der Inhalt größer als der Textbereich ist. Vor der Version v2.03 gab es für das DRESDEN-concept-Logo einen abweichenden Satzspiegel der ggf. beachtet werden muss.
Jetzt werden der Inhalt stückweise so lange auf einzelnen Seiten ausgegeben, bis der immer
kleeiner werdende Rest in der Box auf eine Seite passt.

Passt der auszugebende einspaltige Teil auf eine Seite, ist eine vertikale Korrektur von
.9\baselineskip nicht notwendig. In diesem Fall wird die Box vor der Ausgabe einfach neu
und ohne diese Lange gesetzt.

Der Rest wird einfach als optionales Argument von \twocolumn ausgegeben.

In der Version v2.02 wird eine Warnung ausgegeben, damit der Anwender den Satzspiegel
manuell anpassen kann. Die Verwendung von afterpage liefert leider nicht in allen Fällen das
gewollte Ergebnis, weshalb darauf verzichtet wird.

12.4 Umschlagseite (Cover)

Zusätzlich zur eigentlichen Titelseite kann für Druckerzeugnisse eine separate Umschlagseite
erzeugt werden. Um die Gestaltung dieser anzupassen, wurden Einstellungsmöglichkeiten
vorgesehen, die insbesondere den verwendeten Satzspiegel betreffen.

Der Befehl \makecover ist für eine Umschlagseite bei einem gebundenen Werk zu verwen-

\makecover
ccdcover (Param.)
cd (Param.)
pagenumber (Param.)
cdgeometry (Param.)
@if@tud@cdgeometry@cover
ud@cdgeometry@cover@locked
@if@tud@cover

\newif\if@tud@cover
\newcommand*\makecover[1][]\{

File i: tudscr-title.dtx 186
Ist die Option \texttt{titlepage=firstiscover} aktiv, wird eine Warnung erzeugt.

\begin{verbatim}
\if@titlepageiscoverpage%
\ClassWarning{\TUD@Class@Name}{%
Using both option `titlepage=firstiscover' and `string\maketitle' together isn't supported. The `string\maketitle' option `titlepage=firstiscover' will be ignored. If you want to create an individual cover, you have to use `string\extratitle' instead of `string\maketitle'.
See also the documentation of KOMA-Script%
}%
\@titlepageiscoverpagefalse%
\fi%
\end{verbatim}


\begin{verbatim}
\clearpage%
\begingroup%
\let\thanks\@gobble%
\let\footnote\@gobble%
\TUD@parameter@set{cover}{#1}%
\let\tud@layout@title@num\tud@layout@cover@num%
\tud@layout@process%
\end{verbatim}

Wird der Parameter \texttt{cdgeometry} aktiviert, werden die Seitenränder wie durch das Corporate Design vorgegeben eingestellt. Andernfalls entsprechen diese den Einstellungen des Satzspiegels und können mit \texttt{coverpage...margin} durch den Anwender angepasst werden.

\begin{verbatim}
\if@tud@cdgeometry@cover%
\let\coverpagetopmargin\tud@len@slimmargin%
\let\coverpagebottommargin\tud@len@widemargin%
\let\coverpageleftmargin\tud@len@widemargin%
\let\coverpagerightmargin\tud@len@slimmargin%
\tud@if@v@lower{2.03}{%}
\def\coverpagetopmargin{.5\tud@len@both}%
\def\coverpagebottommargin{.5\tud@len@both}%
\else% !false
\fi%
\end{verbatim}

Für den Fall, dass eine Umschlagseite gesetzt werden soll, die nicht im Corporate Design erscheint, wird einfach das originale \texttt{maketitle} aufgerufen und dabei die Option \texttt{titlepage=firstiscover} verwendet, um die Seitenränder anpassen zu können. Außerdem werden ein mit \texttt{extratitle} gegebener Schmutztitel ebenso ignoriert, wie Danksagung oder Einträge für die Rückseite.

\begin{verbatim}
\ifcase\tud@layout@cover@num\relax\false% false
\KOMAoptions{titlepage=firstiscover}%
\let\extratitle@empty%
\let\upperstitelback@empty%
\let\lovertitleback@empty%
\let\dedication@empty%
\ifx\tud@title@pagenumber\relax\gdef\tud@title@pagenumber{1}\fi%
\tud@cmd@use{maketitle}[	ud@title@pagenumber]%
\else% !false
\end{verbatim}

File: tudscr-title.dtx
Der Befehl verwendet das Layout des Titels im Corporate Design. Allerdings werden nicht alle sondern nur für ein Cover relevante Angaben ausgegeben. Der Schalter if@tud@cover wird innerhalb des Befehles tud@maketitle genutzt, um die Unterscheidung zu machen, ob es sich bei der aktuellen Ausgabe um das Cover oder aber die Titelseite handelt.

626 \@tud@covertrue%
627 \@titlepagetrue%

Der Satzspiegel wird hierfür äquivalent zu den KOMA-Script-Klassen gesetzt, wenn die Option titlepage=firstiscover aktiviert ist.

628 \edef\titlepage@restore{%
629 \noexpand\endgroup%
630 \noexpand\global\noexpand@colht\noexpand@colht%
631 \noexpand\global\noexpand@colroom\noexpand@colroom%
632 \noexpand\global\noexpand@vsize\noexpand@vsize%
633 \noexpand\let\noexpand\titlepage@restore\relax%
634 %}

Linker und rechter Rand sowie die Textbreite werden identisch erzeugt. Für das Layout im Corporate Design wurden die Seitenräder (\coverpage...margin) zuvor angepasst. Die Höhe der Kopfzeile wird auf 0pt gesetzt, die Fußzeile entspricht dem normalen Satzspiegel im Corporate Design.

635 \begingroup%
636 \oddsidemargin=dimexpr\coverpageleftmargin-1in\relax%
637 \evensidemargin=dimexpr\coverpageleftmargin-1in\relax%
638 \textwidth=dimexpr\tud@len@layoutwidth-\coverpageleftmargin%
639 \-\coverpagerightmargin\relax%
640 \headheight=0pt%
641 \headsep=0pt%
642 \footskip=dimexpr\tud@len@footsep+\footheight\relax%
643 \@tempa=2\dimexpr\coverpagetopmargin\relax%
644 \ifx\@tempa\coverpagebottommargin\relax%
645 \let\coverpagebottommargin=tud@len@widemargin%
646 \else%

Bei der Höhe des oberen Seitenrandes muss unterschieden werden, ob der Satzspiegel im Corporate Design oder aber im Buchblock erscheinen soll.

647 \if@tud@cdgeometry@cover%
648 \def\coverpagetopmargin{%
649 \dimexpr\tud@len@topmargin+\tud@len@barheight%
650 \+\tud@len@headsep\relax%
651 %}
652 \else%

Bei der Verwendung eines eigenen Satzspiegels kann der obere Seitenrand über das Makro \coverpagetopmargin geändert werden. Das Vermeiden einer unerwünschten Überdeckung des Kopfes oder ein zu kleiner Fußbereich muss durch den Anwender korrigiert werden. Standardmäßig wird die Höhe jedoch für das Corporate Design angepasst. Auch die Fußhöhe wird im Zweifelsfall gesetzt.

653 \topmargin=dimexpr\tud@len@topmargin+\tud@len@barheight%
654 \+\tud@len@headsep-1in\relax%
655 \def\@tempa{2\dimexpr\coverpagetopmargin\relax}%
656 \ifx\@tempa\coverpagebottommargin\relax%
657 \let\coverpagebottommargin=tud@len@widemargin%
658 \tud@if@v@lower{2.03}%

File i: tudscr-title.dtx
Damit alle Einstellungen korrekt und wie gewünscht übernommen werden, muss die Höhe des Textbereiches \textheight vor der Größe des oberen Seitenrandes gesetzt werden. Dies hat mit der originären Definition von \coverpagetopmargin zu tun.

\textheight=\dimexpr\td@len@layoutheight-\coverpagetopmargin-
\coverpagebottommargin-\td@dim@extrabottommargin\relax
\topmargin=\dimexpr\coverpagetopmargin-1in\relax
\@colht=\textheight
\@colroom=\textwidth
\vsize=\textwidth
\hsize=\columnwidth
\linewidth=\hsize
\tud@len@areavskip=z@

Weil bereits die Titelseite normalerweise die Seitenzahl eins bekommt, kann es bei der Verwendung des Paketes hyperref zu einer Warnung aufgrund der doppelt verwendeten Seitenzahl kommen. Um dies zu verhindern, werden die Seitenanker für das Cover deaktiviert.

\ifundef{\hypersetup}{}{\hypersetup{pageanchor=false}}%

Die Ausgabe des Covers und ggf. das Wiederherstellen des Satzspiegels.

\tud@maketitle%
\clearpage%
\titlepage@restore%
\fi%
\endgroup%


\bool@lock{@tud@cdgeometry@cover}%
\TUD@parameter@family{cover}{%
\TUD@parameter@def{cdcover}{true}{\TUDoption{cdcover}{#1}}%
\TUD@parameter@let{cd}{cdcover}%
\TUD@parameter@def{cdgeometry}{true}{%
\ifstrbool{#1}{%
\bool@set@lock{@tud@cdgeometry@cover}{true}%
\}{}%
\iffalse{#1}{custom}{%
\bool@set@lock{@tud@cdgeometry@cover}{false}{%
\TUD@parameter@error{cdgeometry}{true, false}%
\}%
\}%
\iffalse{#1}{user}{%
\bool@set@lock{@tud@cdgeometry@cover}{false}{%
\TUD@parameter@error{cdgeometry}{true, false}%
\}%
\}%
\iffalse{#1}{custom}{%
\bool@set@lock{@tud@cdgeometry@cover}{false}{%
\TUD@parameter@error{cdgeometry}{true, false}%
\}%
\}%
\iffalse{#1}{user}{%
\bool@set@lock{@tud@cdgeometry@cover}{false}{%
\TUD@parameter@error{cdgeometry}{true, false}%
\}%
\}%
\iffalse{#1}{custom}{%
\bool@set@lock{@tud@cdgeometry@cover}{false}{%
\TUD@parameter@error{cdgeometry}{true, false}%
\}%
\}%
\iffalse{#1}{user}{%
\bool@set@lock{@tud@cdgeometry@cover}{false}{%
\TUD@parameter@error{cdgeometry}{true, false}%
\}%
\}%
\iffalse{#1}{custom}{%
\bool@set@lock{@tud@cdgeometry@cover}{false}{%
\TUD@parameter@error{cdgeometry}{true, false}%
\}%
\}%
\iffalse{#1}{user}{%
\bool@set@lock{@tud@cdgeometry@cover}{false}{%
\TUD@parameter@error{cdgeometry}{true, false}%
\}%
\}%
\iffalse{#1}{custom}{%
\bool@set@lock{@tud@cdgeometry@cover}{false}{%
\TUD@parameter@error{cdgeometry}{true, false}%
\}%
\}%
\iffalse{#1}{user}{%
\bool@set@lock{@tud@cdgeometry@cover}{false}{%
\TUD@parameter@error{cdgeometry}{true, false}%
\}%
\}%
Mit \texttt{TUD\_parameter\_ps\_def} werden alle verfügbaren Parameter für die Kopf- und Fußzeilen definiert.

698 \texttt{TUD\_parameter\_ps\_def\{#1\}\%}

Für den Fall, dass ein Wert nicht in der Schlüssel-Wert-Notation angegeben wurde, wird versucht, das gegebene Argument entweder als booleschen Ausdruck für das Seitenlayout oder als Seitenzahl zu interpretieren.

699 \texttt{TUD\_parameter\_handler\_value\{pagename\}\%}

700 \}
13 Befehle und Umgebungen für den Vorspann

Im Vorspann des Dokumentes kann der Benutzer eine Zusammenfassung oder auch Kurzfassung angeben. Des Weiteren kann eine Selbständigkeitserklärung sowie ein Sperrvermerk hinzugefügt werden. Damit sich für diese das Layout möglichst individuell wählen lässt, werden hier entsprechende Optionen und die dafür notwendige Ausgabe definiert. Im weiteren Verlauf der Dokumentation werden die Begriffe 'Zusammenfassung', 'Kurzfassung' sowie 'Erklärung' für die unterschiedlichen Elemente als gleichwertige Synonyme verwendet. Die Schalter und Befehle zum Setzen der Optionen heißen entweder \texttt{\texttt{tud}@abstract@...} oder aber \texttt{\texttt{tud}@declaration@...} je nach Element.

\texttt{tud@fm@check} Mit diesem Befehl wird für den Fall, dass eine oder mehrerer Erklärungen auf einer einzelnen Seite gesetzt werden sollen geprüft, ob dies überhaupt möglich ist. Sollte dies nicht der Fall sein, wird eine Warnung ausgegeben.

\begin{verbatim}
1 \newcommand*{\tud@fm@check}[2]{%
2 \ifnumless{\csuse{tud@#1@level@num}}{3}{% false/true/section
3 \ifboolexpr{bool {@titlepage} and bool {@tud@#1@fil} and bool {@tud@#1@multi}}{%
4 \sbox\z@{\vbox{#2}}%
5 \ifdim\textheight<\dimexpr\ht\z@+\dp\z@\relax%
6 \ClassWarning{\TUD@Class@Name}{%
7 The given content within the '#1' environment is too large, so it wasn't possible to center the body vertically. Setting the option '#1=nofill' would be a solution. You could also use multiple '#1' environments%
8 }%}
9 \boolfalse{@tud@#1@fil}%
10 \fi%}
11 \texttt{\texttt{tud}@#1@multi}%
12 \texttt{\texttt{tud}@#1@fil}%
13 }{%
14 }%}
\end{verbatim}

Sollte die Gliederungsebene der Erklärung einem Kapitel entsprechen, wird jede Erklärung immer auf eine neue Seite gesetzt. Die Optionen zum vertikalen Zentrieren wird ignoriert. Auf eine Warnung für den Anwender wird verzichtet.

\begin{verbatim}
20 }{% chapter
21 \ifnumless{\csuse{tud@#1@level@num}}{3}{%
22 \ifboolexpr{bool {@titlepage} and bool {@tud@#1@fil} and bool {@tud@#1@multi}}{%
23 \boolfalse{@tud@#1@fil}%
24 }{%
\end{verbatim}
Zum Schluss wird ggf. der gesammelte Eintrag im Inhaltsverzeichnis erstellt. Danach wird das übergebene Argument ausgeführt.

Das Makro \tud@fm@set wird – aufgrund der immanenten Zuweisung auf \tud@fm@next innerhalb von \tud@fm@first – abermals auf die Definition von \tud@fm@first gesetzt.

Ist die Einstellung tocaggregate aktiv, wird der gesammelte Eintrag ins Inhaltsverzeichnis generiert.

Der gesammelte Eintrag aller Überschriften ins Inhaltsverzeichnis.
Mit diesen Makros werden bei der Ausgabe der einzelnen Erklärungen alle relevanten Optionen ausgewertet und die notwendigen vertikalen Abstände entsprechend gesetzt. Dabei wird durch jede Zusammenfassung bzw. Erklärung \texttt{tud@fm@set} über \texttt{tud@fm@body@start} verwendet. Der Inhalt der beiden Umgebungen \texttt{abstract} und \texttt{descriptions} wird mit \texttt{tud@fm@check} ausgegeben.

\newcommand{\tud@fm@set}[1]{}
\newcommand{\tud@fm@first}[1]{%}

Im Zweispaltensatz wird der Inhalt der Umgebung einfach ausgegeben, eine vertikale Zentrierung findet nicht statt. Beim einspaltigen Satz wird die titlepage-Option beachtet.

\newcommand{\tud@fm@set}[1]{%}
\newcommand{\tud@fm@first}[1]{%}

Sollte diese aktiv sein, wird jede Erklärung oder Zusammenfassung abhängig von der Einstellung von \texttt{abstract/declaration=multi} entweder auf eine neue Seite gesetzt, oder aber auf der aktuellen ausgegeben. Dabei werden diese ggf. vertikal auf der Seite ausgerichtet.

\newcommand{\tud@fm@next}[1]{%}

Das Makro \texttt{tud@fm@check} setzt die Definition von \texttt{tud@fm@set} auf \texttt{tud@fm@first}. Bei der erstmaligen Verwendung von \texttt{tud@fm@first} wird diese auf \texttt{tud@fm@next} geändert.

\newcommand{\tud@fm@next}[1]{%}

Bei der abermaligen Verwendung des Makros \texttt{tud@fm@set} in der gleichen Umgebung wird \texttt{tud@fm@next} ausgeführt, um die vertikalen Abstände zwischen den einzelnen Teilen einzufügen.

\newcommand{\tud@fm@next}[1]{%}

Im Zweispaltensatz wird entweder ein Absatz eingefügt oder eine neue Spalte begonnen.

\newcommand{\tud@fm@next}[1]{%}

\textsuperscript{10}titlepage, twocolumn, abstract/declaration=multi, abstract/declaration=fill
Wird eine Titelseite verwendet, wird jede Erklärung oder Zusammenfassung abhängig von der Einstellung von `abstract/declaration=multi` entweder auf eine neue Seite gesetzt, oder aber auf der aktuellen ausgegeben. Dabei werden diese ggf. vertikal auf der Seite ausgerichtet.

```latex
\if@titlepage%
  \ifbool{@tud@#1@multi}{%  
    \ifbool{@tud@#1@fil}{\vfil}{}%
  }{%  
    \ifbool{@tud@#1@fil}{\vfil\null\clearpage\null\vfil}{\clearpage}%
  }
\else%  
  ⟨/book|report|article⟩
\fi%
\fi%  
⟨/book|report|article⟩
\fi%  
\tud@fm@body@start
\tud@fm@body@end
```

Das Makro `\tud@fm@body@start` setzt erst alle übergebenen Parameter für die aktuelle Umgebung. Anschließend werden alle notwendigen vertikalen Abstände mit `\tud@fm@set` eingefügt und ggf. notwendige Umgebungen geöffnet. Der Gegenpart dazu ist das Makro `\tud@fm@body@end`.

Bei deaktivierter `titlepage`-Option werden die Inhalte jeder erzeugten Erklärung oder Zusammenfassung direkt nacheinander ausgegeben. Sollte die Überschrift *nicht* als Gliederungsebene (`\section`, `\chapter`) gesetzt werden, wird der Inhalt entweder in der `quotation`-Umgebung oder – wenn entsprechend das Paket `quoting` geladen wurde – in der `quoting`-Umgebung gesetzt. Die entsprechende Umgebung wird hierfür im Makro `\tud@quoting` gespeichert.

```latex
\if@twocolumn\else\if@titlepage\else\relax%  
\ifnum\csuse{tud@#1@level@num}<\tw@relax%  
\begin{tud@quoting}%  
\fi%  
\fi%  
\fi
\ifnum\tud@x@multicol@num>1\relax%  
\vspace{-\multicolsep}  
\begin{multicols}{\tud@x@multicol@num}[%  
\tud@fm@head{#1}{#3}]
```

Sollte das Paket `multicol` Verwendung finden, wird die entsprechende Umgebung gestartet. Für das Setzen der Überschrift und der Kolumnentitel wird `\tud@fm@head` verwendet, welches abhängig von den gewählten Optionen die Gliederungsebene der Überschrift und die dazugehörigen Abstände setzt. Daher wird die Länge `\multicolsep` auf Null gesetzt.

```latex
\ifnum\tud@x@multicol@num>1\relax%  
\vspace{-\multicolsep}  
\begin{multicols}{\tud@x@multicol@num}{{\tud@fm@head{#1}{#3}}}%
```

File j: tudscr-frontmatter.dtx
Ansonsten wird lediglich die Überschrift der Zusammenfassung oder Erklärung im dritten Argument entsprechend der gewählten Option erzeugt.

Nach der Ausgabe des Inhaltes werden mit \tud@fm@body@end alle ggf. geöffneten Umgebungen wieder geschlossen.

Dieses Makro dient zur Ausgabe der Erklärungsüberschrift auf der gewünschten Gliederungsebene. Die gewählte Ebene ist in \tud@⟨Typ⟩@level@num gespeichert. Verwendet wird das Makro folgendermaßen: \tud@fm@head{(Erklärungstyp)⟨Überschrift⟩}. Damit die ggf. aktive Option abstract/declaration=toc funktionieren kann, wird für die Level, in denen keine Standardüberschrift verwendet wird, mit dem Befehl \phantomsection aus dem Paket hyperref – wenn es tatsächlich geladen wurde – ein Anker für einen Hyperlink erzeugt.

Ist \tud@⟨Typ⟩@toc@num ungerade, so entspricht die Ebene des erzeugten Eintrags ins Inhaltsverzeichnis für \texttt{tudscrbartcl} normalerweise dem eines Abschnitts, der für \texttt{tudscrbk} und \texttt{tudscrrep}t dem eines Kapitels.\footnote{sonst sieht das Inhaltsverzeichnis recht bescheiden aus} Mit der Eintellung \texttt{tocleveldown} kann der Eintrag um eine Ebene reduziert werden.
Dieser Befehl sorgt für die Auswahl des Seitenstiles über die Parameter von \texttt{abstract} und \texttt{declarations}.

\begin{verbatim}
\newcommand*{\tud@fm@pagestyle}[2]{%  
  \ifs@titlepage%  
    \ifcsundef{ps@#2}{%  
      \ClassError{\TUD@Class@Name}{'#2' is no valid pagestyle}{%  
        You tried to use '#2' as a pagestyle for the\MessageBreak%  
        '1' environment, but it was never defined.%  
    }%  
    }{%  
      \def{\tud@envir@ps}{#2}%  
    }%  
  \else%  
    \ClassWarning{\TUD@Class@Name}{%  
      The key 'pagestyle' can only be used with\MessageBreak%  
      activated option 'titlepage'\MessageBreak%  
    }%  
  \fi%  
\end{verbatim}

\section*{\texttt{\textbackslash tud@fm@pagestyle}}
Das empfehlenswerte Paket *quoting* unterstützt werden. Sollte dieses geladen werden, wird für das Setzen der Zusammenfassung bei einem Titelkopf die *quoting*-Umggebung genutzt.

```latex
\newcommand*{\tud@quoting}{quotation}
\AfterPackage{quoting}{\renewcommand*{\tud@quoting}{quoting}}
```

Das aus KOMA-Script bekannte Verhalten für die Zusammenfassung wird für die beiden Klassen *tudscrartcl* und *tudscrbreport* adaptiert. Dazu werden boolesche Schlüssel bei der Definiton der beiden Optionen *abstract* und *declaration* eingeführt. Außerdem werden zusätzliche Werte zur Auswahl einer Gliederungsebene für die gesetzten Überschriften wie *section* oder *chapter* oder bereitgestellt, welche auch für die Buchklasse *tudscrbook* zur Verfügung stehen. Das Setzen von Einträgen ins Inhaltsverzeichnis lässt sich ebenfalls anpassen.


```latex
\newcommand*{\tud@fm@switch}{%
{section}{2},{sect}{2},{sec}{2},{addsec}{2},%
{article}{poster}
{new}{2},{standard}{2},{heading}{2},%
{/article}{poster}
{book}{report}
{chapter}{3},{chap}{3},{addchap}{3},%
{new}{3},{standard}{3},{heading}{3},%
{/book}{report}
{book}{report}{article}
{single}{4},{one}{4},{simple}{4},%
{multiple}{5},{multi}{5},{all}{5},{aggregate}{5},%
{two}{5},{both}{5},{double}{5},%
{nofill}{6},{nofill}{6},{novfill}{6},{novfill}{6},%
{fill}{7},{fill}{7},{vfill}{7},{vfill}{7},%
{markboth}{8},{mark}{8},%
{nomarkboth}{9},{nomark}{9},%
{nototoc}{10},{notoc}{10},%
{totoc}{11},{toc}{11},{totoclevelup}{11},{toclevelup}{11},{levelup}{11},%
{totocleveldown}{12},{tocleveldown}{12},{leveldown}{12},%
{tocmultiple}{13},{tocmulti}{13},{tocall}{13},{tocaggregate}{13},%
{totocmultiple}{13},{totocmulti}{13},{totocall}{13},{totocaggregate}{13}%
{/book}{report}{article}
}
```

^12 nicht für *tudscrartcl*
13.1 Erweiterung der Umgebung für eine Zusammenfassung

Die abstract-Umgebung wird um mehrere Optionen erweitert. So kann in den neuen TUD-Script-Klassen die Sprache der Zusammenfassung leicht eingestellt und auch zwei Zusammenfassungen auf eine Seite gesetzt werden.

\tud@abstract@level@num \tud@abstract@toc@num \tud@abstract@toc@num@locked

Der Befehl \tud@abstract@level@num beschreibt die Gliederungsebene der Überschrift der Zusammenfassung numerisch und wird über die Schlüssel der Option abstract gesetzt. Über abstract=toc/notoc wird festgelegt, ob die Zusammenfassung einen Eintrag ins Inhaltsverzeichnis bekommt. Da unterschiedliche Gliederungsebenen der Überschriften möglich sind, wird das Standardverhalten in Abhängigkeit dieser gewählt. Initial erscheint für tudscrreport und tudscrapartcI – wie in KOMA-Script auch – keine Überschrift und kein Eintrag im Inhaltsverzeichnis. In tudscrbook wird standardmäßig eine Überschrift in Form eines Kapitels mit Eintrag ins Inhaltsverzeichnis gesetzt.

Mit \tud@abstract@toc@num wird die Gestalt des Inhaltsverzeichniseintrag festgelegt. Folgende Werte sind möglich:

<table>
<thead>
<tr>
<th>Wert</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>kein Eintrag ins Inhaltsverzeichnis</td>
</tr>
<tr>
<td>1</td>
<td>Individuelle Einträge auf der obersten Gliederungsebene</td>
</tr>
<tr>
<td>2</td>
<td>Individuelle Einträge auf der zweithöchsten Gliederungsebene</td>
</tr>
<tr>
<td>3</td>
<td>Ein gesammelter Eintrag auf der obersten Gliederungsebene</td>
</tr>
<tr>
<td>4</td>
<td>Ein gesammelter Eintrag auf der zweithöchsten Gliederungsebene</td>
</tr>
</tbody>
</table>

Dabei ist die oberste Gliederungsebene \chapter für tudscrbook und tudscrreport sowie \section für tudscrapartcI.

\newcommand*{\tud@abstract@level@num}{0}
\cs@lock{tud@abstract@toc@num}{0}
\newcommand*{\tud@abstract@level@num}{3}
\cs@lock{tud@abstract@toc@num}{1}
\if@tud@abstract@multi
\if@tud@abstract@fil
\if@tud@abstract@markboth

Is dieser Schalter aktiviert, wird eine durch \nextabstract angegebene zweite – wahrscheinlich in einer Fremdsprache verfasste – Zusammenfassung auf die gleiche Seite wie die erste zu setzen. Gesetzt wird der Schalter über die Option abstract=single/multi.

\newif\if@tud@abstract@multi
\@tud@abstract@multitrue
\newif\if@tud@abstract@fil
\@tud@abstract@filtrue
\newif\if@tud@abstract@markboth

Mit diesem Schalter wird bestimmt, ob eine Zusammenfassung auf einer Seite vertikal zentriert wird. Er wird mit abstract=fill/nofill gesetzt und ist normalerweise aktiviert.

\newif\if@tud@abstract@fil
\@tud@abstract@filtrue

Dieser Schalter setzt rechten und linken Kolumnentitel automatisch auf die verwendete Überschrift.
Alle Klassen unterstützen die Werte, welche in `\tud@fm@switch` definiert wurden. Die beiden Klassen `tudscrarticle` und `tudscrreport` adaptieren außerdem das aus KOMA-Script bekannte Verhalten über die booleschen Werte aus `\TUD@bool@numkey`.

Bei der Einstellungen der Überschriftgliederungsebene wird außerdem das Standardverhalten für einen Eintrag ins Inhaltsverzeichnis festgelegt, was allerdings vom Anwender jederzeit überschrieben werden kann.

Neben den Einstellungen für die Art der Gliederungsebene der Überschrift für die Zusammenfassung können außerdem noch die Optionen gesetzt werden, ob versucht werden soll, eine mögliche Zusammenfassung in einer anderen Sprache auf die gleiche Seite wie die erste zu setzen (`abstract=multi`).

File j: `tudscr-frontmatter.dtx`
Bei den Einstellungen für das Inhaltsverzeichnis werden die unterschiedlichen Möglichkeiten im Makro \tud@abstract@toc@num zusammengefasst.

\or\% nottotoc
\cs@set@lock{tud@abstract@toc@num}{0}\%
\or\% totoc
\ifnum\tud@abstract@toc@num<\thr@@\relax%
\cs@set@lock{tud@abstract@toc@num}{1}\%
\else%
\cs@set@lock{tud@abstract@toc@num}{3}\%
\fi\
\or\% totoleveldown
\ifnum\tud@abstract@toc@num<\thr@@\relax%
\cs@set@lock{tud@abstract@toc@num}{2}\%
\else%
\cs@set@lock{tud@abstract@toc@num}{4}\%
\fi\
\or\% totocmultiple
\ifnum\tud@abstract@toc@num=\z@\relax%
\cs@set@lock{tud@abstract@toc@num}{1}\%
\edef\@tempa{\noexpand\cs@set@lock{tud@abstract@toc@num}{\the\numexpr\tud@abstract@toc@num+2}\relax%}
\@tempa\%
\else%
\csundef{abstract}
\csundef{endabstract}
\NewEnviron{abstract}[1][]{% Sollten Überschriften in Form von Kapiteln gewählt worden sein, so wird temporär die titlepage-Option aktiviert, da alle weiteren notwendigen Einstellungen dieser entsprechen. Über den Parameter pagestyle kann bei der Verwendung einer Titelseite des Seitenstil

\titlepage, twocolumn, abstract=multi, abstract=fill
definiert werden. Dafür wird das dafür verantwortliche Makro \texttt{tud@envir@ps} auf \texttt{relax}
gesetzt.

Als nächstes werden die Optionen verarbeitet. Sollte nach Abarbeitung der Optionen für \texttt{abstract} festgestellt werden, dass eine spezielle Spaltenanzahl gewünscht ist, so wird – für den Fall, dass das Paket \texttt{multicol} geladen ist – diese gesetzt. Sonst wird die angegebene Anzahl der Spalten ignoriert und eine Warnung ausgegeben.

Bei aktiver \texttt{titlepage}-Option wird anschließend der aktuelle Seitenstil gesichert, um diesen nach der Umgebung wiederherstellen zu können. Wurde über die möglichen Parameter \texttt{pagestyle} oder indirekt mit \texttt{markboth} kein Seitenstil definiert, so wird dieser auf \texttt{empty} gesetzt.

Mit dem Befehl \texttt{\nextabstract} können einzelne Abschnitte innerhalb einer Zusammenfassung getrennt werden.

Nach der Umgebung wird bei aktiver \texttt{titlepage}-Option der ursprüngliche Seitenstil zurückgesetzt. Zuvor wird ggf. der notwendige vertikale Leerraum eingefügt.
Als Schlüssel für die abstract-Umgebung können Sprache, Anzahl der Spalten oder auch die zur Umgebung gehörigen Klassenoptionen angegeben werden.

Für das Setzen von Kolumnentiteln wird der Seitenstil auf headings gesetzt, falls kein anderer mit pagestyle angegeben wurde.

Für die Optionsangabe ohne Schlüssel und Wert kann eine Anzahl an Spalten oder eine alternative Sprache angegeben werden. Für die Auswertung des Argumentes verantwortlich ist der Befehl \TUD@parameter@handler@default, welcher auch von der Umgebung tudpage verwendet wird.

Der Befehl \textabstract lässt sich nur in der Umgebung abstract verwenden.
13.2 Selbstständigkeitserklärung und Sperrvermerk

Die Befehle für die Selbstständigkeitserklärung und den Sperrvermerk werden äquivalent zur abstract-Umgebung mit den gleichen Optionen ausgestattet.

Siehe die Option abstract.

declaration (Opt.)

File j: tudscr-frontmatter.dtx
203
\if@tud@declarations

Die \textit{declarations-}Umgebung wird ähnlich zur \textit{abstract-}Umgebung definiert. Prinzipiell funktioniert diese genauso, inklusive der Parameter. Der Schalter \texttt{\if@tud@declarations} wird verwendet, um die weiteren Befehle \texttt{\declaration}, \texttt{\confirmation} und \texttt{\blocking} innerhalb und außerhalb dieser Umgebung verwenden zu können.

\newif\if@tud@declarations
\NewEnviron{declarations}[1][{}]{%
  \@tud@declarationstrue%
  \ifnum\tud@declaration@level@num>\tw@elax\@titlepagetrue\fi% chapter
  \let\tud@envir@ps\relax%
  \TUD@parameter@set{declaration}{#1}%
  \tud@x@multicol@check%
  \if@titlepage%
    \tud@currentpagestyle@set%
    \ifx\tud@currentpagestyle@set%
      \tud@currentpagestyle@set%
    \else\tud@envir@ps\relax%
      \tud@if@tudheadings{\currentpagestyle}{%\def\tud@envir@ps{empty.tudheadings}%
        \%\def\tud@envir@ps{empty}%
        \%\def\tud@envir@ps{empty}%
      \}
    \fi
    \clearpage%
    \tud@envir@selectps%
  \fi
\if@tud@declarations

Falls innerhalb der Erklärung durch den Anwender bestimmte Felder verwendet werden, werden vorsichtshalber Befehle für die Titelei „entschärft“:

\def\and{, }%
\let\thanks\@gobble%
\fi
\fi
\fi
Im Gegensatz zur abstract-Umgebung werden in der Umgebung declarations alle Erklärungen als geschlossene Variante genutzt, um diese einzeln und nacheinander anzugeben. Jedes hierfür nutzbare Makro \declaration, \nextdeclaration, \blocking und \confirmation kann so innerhalb der Umgebung verwendet werden und führt intern die beiden Makros \tud@fm@body@start und \tud@fm@body@end selbstständig aus. Alternativ dazu müsste relativ umständlich geprüft werden, ob der Aufruf von \tud@fm@body@start bereits erfolgte und danach auch schon Text ausgegeben wurde. Einen Mehrwert bei der Nutzung hätte dies jedoch nicht gehabt außer der äquivalenten Verwendung von \nextdeclaration und \nextabstract, worauf der Einfachheit halber gerne verzichtet wurde, insbesondere da die erwartete Nutzung des Befehls \nextdeclaration als sehr gering einzustufen ist.

Dies sind die möglichen Schlüssel für die Befehle \declaration, \confirmation und \blocking. Die Schlüssel declaration bzw. option dienen zum Setzen der Werte, welche auch als Klassenoptionen gesetzt werden können.

\tud@fm@supporter Dies ist ein Hilfsmakro, um die Sternversionen der beiden Befehle \declaration und \confirmation zu realisieren.

\def\tud@fm@declaration{
  \ifstar{
    \def\tud@fm@supporter{supporter=\@supervisor}
    \tud@fm@confirmation
  }{
    \def\tud@fm@supporter{supporter=\@supporter}
    \tud@fm@confirmation
  }\tud@fm@declaration%
}
\def\tud@fm@declaration[1][2]{\ifbool{\tud@declarations}{\tud@fm@body@start{declaration}{#1}{#2}#3\tud@fm@body@end{declaration}}{\ClassError{\TUD@Class@Name}{Incorrect usage of \stringnextdeclaration}{The command \stringnextdeclaration can only be used within the environment declarations'\MessageBreak}}}
\def\tud@fm@confirmation{
  \ifstar{
    \def\tud@fm@supporter{supporter=\@supervisor}
    \tud@fm@confirmation
  }{
    \def\tud@fm@supporter{supporter=\@supporter}
    \tud@fm@confirmation
  }\tud@fm@confirmation%
}

Mit \nextdeclaration kann eine freie Erklärung abgegeben werden. Dabei sind – im Gegensatz zu \nextabstract – sowohl Überschrift als auch der Inhalt als obligatorisches Argument anzuzeigen, da innerhalb der Umgebung declarations die geschlossene Angabe des Inhalts innerhalb von \tud@fm@body@start und \tud@fm@body@end notwendig ist.

Mit diesem Befehl kann die Selbstständigkeitserklärung ausgegeben werden. Das optionale Argument bestimmt, wer als Unterstützer angegeben wird. Außerdem sind Schlüssel-Wert-Paare als Option nutzbar. Mit der Sternversion werden die mit \supervisor angegebenen Betreuer als Unterstützer genannt.
Damit das optionale Argument ohne Schlüssel für die Unterstützer verwendet werden kann, wird der Handler lokal umdefiniert. Außerdem werden irrelevant Parameter undefiniert gesetzt.

\begin{group}
\TUD@parameter@handler@default[declaration]{supporter}%
\undef\KV@declaration@company%
\ifbool{@tud@declarations}{%
\tud@fm@confirmation[\tud@fm@supporter,#1]%
}%
\begin{declarations}[\tud@fm@supporter,#1]%
\tud@fm@@confirmation%
\end{declarations}%
\endgroup%

Dies ist der eigentliche Inhalt des Befehls \confirmation. Für den Sperrvermerk wird äquivalent zu \confirmation verfahren. Das optionale Argument ohne Schlüssel setzt hier die Firma für den Sperrvermerk.

\blocking
\tud@fm@blocking
\begin{group}
\TUD@parameter@handler@default[declaration]{company}%
\undef\KV@declaration@supporter%
\undef\KV@declaration@closing%
\undef\KV@declaration@confirmationclosing%
\ifbool{@tud@declarations}{%
\tud@fm@blocking[#1]%
}%
\begin{declarations}[#1]%
\tud@fm@blocking%
\end{declarations}%
\endgroup%

Dies ist der eigentliche Inhalt des Befehls \blocking.
14 Kompatibilität zu früheren TUD-Script-Versionen

Mit der Version v2.02 wurde eine Menge – teilweise sehr tiefgreifend – an der Benutzerschnittstelle in TUD-Script geändert. Dabei wird versucht, die Kompatibilität zu früheren Versionen so weit es geht aufrecht zu erhalten und veraltete Optionen und Befehle zumindest weiter bereitzustellen, wobei bei der Verwendung dieser der Anwender darüber informiert wird. Die Vorkehrungen betreffen sowohl die Klassen selber als auch die zusätzlich bereitgestellten Pakete.

Um alte Optionen und Befehle dem Anwender bei der Verwendung kenntlich zu machen, werden entsprechende Warnungen definiert. Für Optionen wird dabei lediglich die empfohlene Option ausgegeben. Das Ausführen dieser wird später definiert.

Bei alten Befehlen wird nach der Warnung die Definition des neuen Befehls auf den alten überschrieben.
Using the length ‘\backslashchar#2’ is deprecated.
You should use option ‘#2’ instead.
\csxdef{tud@dim@#2}{\the\csuse{#2}}
\expandafter\setlength\csname #2\endcsname{\maxdimen}
\global\csuse{#2}=\csuse{#2}
\fi
}

⟨
/class & body
⟩
tudscrver\ (Opt.)
\tud@v@comp

In einigen Fällen sind Änderungen mit früheren Versionen nicht kompatibel oder unerwünscht, weil diese beispielsweise das Ausgabeergebnis verändern. Standardmäßig werden die Klassen in der aktuellen Version geladen. Mit \tudscrver=⟨Version⟩ kann auf das Verhalten einer früheren Version geschaltet werden. Die eingestellte Kompatibilität wird als Zahl in \tud@v@comp gespeichert. In den Makros \tud@v⟨Version⟩ werden die zugehörigen Nummern gespeichert.

\newcommand*{\tud@v@comp}{\tud@v@last}
\TUD@key{tudscrver}[last]{%
\tud@v@get\@tempa{#1}{%
\ifcsundef{tud@v\@tempa}{%
⟨
/*class*/
\ClassWarningNoLine{\TUD@Class@Name}%
⟨
/*package & fonts*/
\PackageWarningNoLine{tudscrfonts}%
⟨
/*package & fonts*/
{%
You have set option ‘tudscrver’ to ‘\@tempa’, but\MessageBreak
this value is not supported. Because of this, the\MessageBreak
value for ‘tudscrver’ was set to ‘first’%}
\}
\def\tud@v@comp{0}%
{
⟨
/*class*/
\ClassInfoNoLine{\TUD@Class@Name}%
⟨
/*package & fonts*/
\PackageInfoNoLine{tudscrfonts}%
⟨
/*package & fonts*/
{%
Switching compatibility level to ‘\@tempa’%}
\}
\edef\tud@v@comp{\csuse{tud@v\@tempa}}%
}%
{FamilyKeyStateProcessed%
}%

Eine zusätzliche Bedingung gibt es noch: Die Kompatibilität kann nur beim Laden der Klasse bzw. des Paketes gesetzt werden, danach nicht mehr.

⟨
/class & body
⟩

\AtEndOfClass%
⟨
/*package & fonts*/
\PackageInfoNoLine{tudscrfonts}%
⟨
/*package & fonts*/
{%
\edef\tud@v@comp{\csuse{tud@v\@tempa}}%
}%
{FamilyKeyStateProcessed%
}%

File k: tudscr-comp.dtx 209
Außerdem wird darauf geachtet, dass die mindestens benötigte Version von KOMA-Script installiert ist. (Technische Universität Dresden @Version@KOMA: Technische Universität Dresden @Version@KOMA). Sollte dies nicht der Fall sein, wird ein Fehler erzeugt.

\begin{tabular}{|c|c|c|}
\hline
\texttt{\tud@v@first} & \texttt{\tud@v@2.00} & \texttt{\tud@v@2.01} \\
\texttt{\tud@v@2.01a} & \texttt{\tud@v@2.02} & \texttt{\tud@v@2.02a} \\
\texttt{\tud@v@2.02b} & \texttt{\tud@v@2.03} & \texttt{\tud@v@2.03a} \\
\texttt{\tud@v@2.03b} & \texttt{\tud@v@2.04} & \texttt{\tud@v@2.04a} \\
\texttt{\tud@v@2.04b} & \texttt{\tud@v@2.05} & \texttt{\tud@v@2.05a} \\
\texttt{\tud@v@2.05b} & \texttt{\tud@v@2.06} & \texttt{\tud@v@2.05c} \\
\texttt{\tud@v@2.05d} & \texttt{\tud@v@2.06} & \texttt{\tud@v@2.06a} \\
\hline
\end{tabular}

Die numerischen Werte zu den einzelnen Versionen.
\tud@v@get Mit \tud@v@get wird die angegebene Versionsnummer in eine Normalform gebracht und an das Makro im ersten Argument übergeben. Damit spielt es keine Rolle, ob die im zweiten Argument übergebene Versionsnummer mit oder ohne führendes ',v' angegeben wird.

\newcommand*{\tud@v@get}[2]{\begingroup\def\@tempa{\kernel@ifnextchar v{\@tempb}{\@tempb v}}\def\@tempb v##1\@nil{\def\@tempc{##1}}\@tempa #2\@nil\edef\tud@reserved{\endgroup\ifnum\tud@v@comp<\csuse{tud@v@\@tempa}\relax\def\tud@reserved{\endgroup\@firstoftwo}\else\def\tud@reserved{\endgroup\@secondoftwo}\fi}\tud@reserved\endgroup}

\tud@if@v@lower Mit diesem Befehl können abhängig von der gegebenen Kompatibilitätsversion in den Klassen unterschiedliche Einstellungen vorgenommen werden.

\newcommand*{\tud@if@v@lower}[1]{\begingroup\tud@v@get\@tempa{#1}\ifcsdef{tud@v@\@tempa}{\ifnum\tud@v@comp<\csuse{tud@v@\@tempa}\relax\def\tud@reserved{\endgroup\@firstoftwo}\else\def\tud@reserved{\endgroup\@secondoftwo}\fi\def\tud@reserved{\endgroup\@secondoftwo}\%\PackageWarningNoLine{tudscrfonts}\%\@firstoftwo}{\ClassWarningNoLine{\TUD@Class@Name}\%\@secondoftwo}{%\MessageBreak\There's no defined macro '\string\tud@v@\@tempa'\%\PackageWarningNoLine{tudscrfonts}\%\@secondoftwo}{%\MessageBreak\There's no defined macro '\string\tud@v@\@tempa'\%\PackageWarningNoLine{tudscrfonts}\%\@secondoftwo}}\tud@reserved\endgroup}
14.1 Veraltete Optionen

*fontspec* (Opt.)  Früher musste die Unterstützung für die Schriftauswahl via *fontspec* manuell über die folgende Option aktiviert werden.

\begin{Verbatim}
\TUD\key{fontspec}[true]{%
⟨*class⟩
\ClassWarningNoLine{\TUD@Class@Name}%
⟨/class⟩
⟨*package & fonts⟩
\PackageWarningNoLine{tudscrfonts}%
⟨/package & fonts⟩
{%
The option key ‘fontspec’ is deprecated\MessageBreak
You should load the package ‘fontspec’ manually%
}%
\TUD\set@ifkey{fontspec}{@tud@x@fontspec@enabled}{#1}{%
⟨*class⟩
\ClassError{\TUD@Class@Name}%
⟨/class⟩
⟨*package & fonts⟩
\PackageError{tudscrfonts}%
⟨/package & fonts⟩
{Option ‘fontspec’ too late}{%
Option ‘fontspec’ can only be set while loading the\MessageBreak
\if@tud@x@fontspec@enabled%
\end{Verbatim}
Diese Optionen werden nur aus Gründen der Kompatibilität zu v1.0 definiert.

Hiermit lässt sich die Schriftstärke im Dokument beeinflussen.

Diese Optionen dienten zur Auswahl, ob die Schrift \texttt{DIN BOLD} für die Überschriften verwendet werden soll.
Mit dieser Option kann die genutzte Standardschrift für den Mathematiksatz für das gesamte Dokument umgestellt werden. Die \texttt{tudbook}-Klasse hat neben der Option \texttt{sansmath} außer-
dem den zusätzlichen Schlüssel \texttt{serifmath} definiert, welcher aus Gründen der Kompatibilität hier ebenfalls vorgehalten wird.

\begin{verbatim}
291 \TUD@key{sansmath}[true]{%
292 \TUD@set@ifkey{sansmath}{@tempswa}{#1}%
293 \ifx\FamilyKeyState\FamilyKeyStateProcessed%
294 \if@tempswa%
295 \TUD@deprecated@key{sansmath}{cdmath=true}%
296 \else%
297 \TUD@deprecated@key{sansmath=#1}{cdmath=false}%
298 \fi%
299 %}
300 }
\end{verbatim}

\begin{verbatim}
barfont (Opt.) Mit dieser Option kann die Schrift des Corporate Designs und deren Schriftstärke in der TUD-Kopfzeile unabhängig von der gewählten Schriftart für den Fließtext aktiviert werden.

312 \TUD@key{barfont}[true]{%
313 \TUD@set@numkey{barfont}{@tempa}{%
314 \TUD@bool@numkey,%
315 {nocc}{0},{noccfonts}{0},{nocdfont}{0},{notudfonts}{0},%
316 {cd}{1},{cdfont}{1},{cdfonts}{1},{tudfonts}{1},%
317 {light}{1},{lightfont}{1},{lite}{1},{litefont}{1},{noheavyfont}{1},%
318 {heavy}{2},{heavyfont}{2},{bold}{2},{boldfont}{2}%
319 }{#1}%
\end{verbatim}
widehead (Opt.) Der Schalter dient zur Steuerung der Breite des Querbalkens im Kopf der tudpage-Seite. Entweder der Querbalken liegt im Satzspiegel oder aber über die komplette Papierbreite.

```
330 \TUD@key{widehead}[true]{%
331 \TUD@set@ifkey{widehead}{@tempswa}{#1}%
332 \ifx\FamilyKeyState\FamilyKeyStateProcessed%
333 \if@tempswa%
334 \TUD@deprecated@key{widehead}{cdhead=paperwidth}%
335 \else%
336 \TUD@deprecated@key{widehead=#1}{cdhead=textwidth}%
337 \fi%
338 \fi%
339 }
```

tudfoot (Opt.) Diese Option wird nur aus Gründen der Kompatibilität zu v1.0 definiert.

```
340 \TUD@key{tudfoot}[true]{%
341 \TUD@set@ifkey{tudfoot}{@tempswa}{#1}%
342 \ifx\FamilyKeyState\FamilyKeyStateProcessed%
343 \if@tempswa%
344 \TUD@deprecated@key{tudfoot}{cdfoot=true}%
345 \else%
346 \TUD@deprecated@key{tudfoot=#1}{cdfoot=false}%
347 \fi%
348 \fi%
349 }
```


```
350 \TUD@key{color}[true]{%
351 \TUD@set@numkey{color}{@tempa}{%
352 \TUD@bool@numkey,%
353 {nocolor}{0},{nocolour}{0},{monochrome}{0},{monochromatic}{0},%,
354 {color}{1},{colour}{1},%
355 {lite}{2},{light}{2},{pale}{2},%
356 {colorlite}{2},{lightcolor}{2},{palecolor}{2},%
357 {colorlight}{2},{lightcolour}{2},{palecolour}{2},%
358 {colormapale}{2},{pale colormap}{2},{palecolour}{2},%
359 {bicolor}{3},{biccolour}{3},{biccolour}{3},%
360 {bichrome}{3},{bicromatic}{3},{dichrome}{3},{dichromatic}{3},%
361 {full}{4},{colorfull}{4},{fullcolor}{4},{colourfull}{4},{fullcolour}{4}},%
362 }(#1)%
363 \ifx\FamilyKeyState\FamilyKeyStateProcessed%
364 \ifcase\@tempa\relax% false
365 \TUD@deprecated@key{color=#1}{cd=true}%
```

14.2 Veraltete Befehle und Längen

\footlogoheight (Länge) Um die Höhe von Logos im Fußbereich der tudheadings-Seitenstile einheitlich festzulegen, gab es bis einschließlich der Version v2.04 die Länge \footlogoheight anstelle der Option footlogoheight. Sollte der Anwender diese anpassen, wird \tud@dim@footlogoheight auf diesen Wert gesetzt und eine Warnung erzeugt.

\pageheadingsvskip (Länge) Mit diesen Längen konnte der Anwender vor der Version v2.05 die Überschriften von Titel, Teilen und Kapiteln vertikal verschieben. Diese Funktionalität wird seitdem mit den Optionen headingsvskip und pageheadingsvskip abgedeckt.

\chapterheadingvskip (Länge) Die Länge \chapterheadingvskip wird aus Gründen der Kompatibilität zu älteren Versionen definiert.

\professorship Für die Angabe des Lehrstuhls bzw. der Professur mit kann anstelle von \chair als Aliasbefehl auch \professorship genutzt werden.

\studentid \matriculationid Zur Angabe von Matrikelnummer kann auch \studentid oder \matriculationnumber genutzt werden.

\enrolmentyear Das Immatrikulationsjahr kann auch mit \enrolmentyear angegeben werden.
\birthplace Zur Angabe des Geburtsortes kann auch \birthplace verwendet werden.

394 \TUD@deprecated@cs{birthplace}{placeofbirth}

\submissiondate Der Befehl \submissiondate kann als Aliasbefehl für den Standardbefehl \date zur Datums-
angabe genutzt werden.

395 \TUD@deprecated@cs{submissiondate}{date}

\oralexaminationdate Für \defensedate kann als Aliasbefehl auch \oralexaminationdate verwendet werden.

396 \TUD@deprecated@cs{oralexaminationdate}{defensedate}

\birthday Der Geburtstag kann auch mit \birthday angegeben werden.

397 \TUD@deprecated@cs{birthday}{dateofbirth}

\location Für die Angabe des Ortes kann auch \location genutzt werden.

398 \TUD@deprecated@cs{location}{place}

\logofile \logofilename Diese beiden Befehle können anstelle von \headlogo eingesetzt werden.

399 \TUD@deprecated@cs{logofile}{headlogo}
400 \TUD@deprecated@cs{logofilename}{headlogo}

\startdate \finaldate \maturitydate Für das Paket tudscrsupervisor sind auch einige Befehle als veraltet anzusehen.

401 \AfterPackage{tudscrsupervisor}{%
402 \TUD@deprecated@cs{startdate}{issuedate}%
403 \TUD@deprecated@cs{finaldate}{duedate}%
404 \TUD@deprecated@cs{maturitydate}{duedate}%
405 }

Ab der TUD-Script-Version v2.04 werden einige vertikale Abstände abhängig von der Schrift-
größe definiert. Sollte eine frühere Kompatibilitätseinstellung angegeben worden sein, wird
dieses verhalten deaktiviert.

406{+class}
407 \AtEndOfClass%
408{/class}
409{+package & fonts}
410 \AtEndOfPackage%
411{/package & fonts}
412{%
413 \tud@if@v@lower{2.04}{\TUDoptions{relspacing=false}}{}}%
15 Das Paket tudscrcomp – Kompatibilität veralteter Klassen

Neben den TUD-Script-Klassen gibt es zahlreiche andere \LaTeX-Umsetzungen des Corporate Designs der Technischen Universität Dresden. Um eine Migration von diesen „veralteten“ Klassen zu erleichtern, wird das Paket \texttt{tudscrcomp} bereitgestellt. Mit diesem können bei der Verwendung von TUD-Script die meisten Optionen und Befehle genutzt werden, welche durch die Klasse \texttt{tudbook} und teilweise vormals durch die Klassen die TUD-Script-Klassen in der Version v1.0 sowie die Posterklassen \texttt{tudmathposter} bzw. \texttt{tudposter} bereitgestellt wurden.

15.1 Erkennen der geladenen Klasse

Damit je nach geladener Klasse die passenden Kompatibilitätseinstellungen und -befehle bereitgestellt werden können, wird auf die geladene Klasse geprüft. Danach wird das dazu passende Paket geladen.

```
\PassOptionsToPackage{oldcolors}{tudscrcolor}
\ifclassloaded{tudscrposter}{
  \RequirePackageWithOptions{tudscrcomp-poster}{\TUD@Version}\}
\else{
  \RequirePackageWithOptions{tudscrcomp-book}{\TUD@Version}\}
```

15.2 Gemeinsame Optionen und Befehle für alle Klassen

Zunächst werden alle Optionen und Befehle definiert, die unabhängig von der geladenen Klasse generell bereitgestellt werden.

```
\newcommand*{\einrichtung}{\faculty}
\newcommand*{\fachrichtung}{\department}
\newcommand*{\institut}{\institute}
\newcommand*{\professur}{\chair}
```

\textbf{\texttt{\dinBold}}\textbf{\texttt{\dinfamily}}

Für die explizite Schriftauswahl gibt es ebenfalls zwei ältere Befehle.

```
\newcommand*{\dinBold}{\dinbn}
\newcommand*{\dinfamily}{\dinbn}
```

15.3 Kompatibilität zu \texttt{tudbook}

Die folgenden Optionen und Befehle werden durch die Klasse \texttt{tudbook} und teilweise durch TUD-Script v1.0 definiert.

```
\TUD@key{colortitle}[true]{%
\TUD@set@ifkey{colortitle}{@tempswa}{#1}{%}
```

\textbf{\texttt{\texttt{\texttt{colortitle}} (Opt.)}}\textbf{\texttt{\texttt{nocolortitle}} (Opt.)}

Für farbige Einstellungen wird von \texttt{tudbook} die Option \texttt{color} definiert. Soll die Titelseite konträr dazu gesetzt werden, muss sich mit den Schlüsseln \texttt{colortitle} und \texttt{nocolortitle} behelfen werden.

```
\TUD@key{colortitle}[true]{%
\TUD@set@ifkey{colortitle}{@tempswa}{#1}{%}
```
Außer der Option ddc gibt es bei der alten tudbook-Klasse noch den Schlüssel ddcfooter. Dieser wird auf die Option ddcfoot gelegt.

Es werden weitere Aliasbefehle für die Eingabefelder der tudbook-Klasse definiert.

Bei der Definition des Typs der Abschlussarbeit mit \dissertation wird die Lokalisierungsvariable \dissertationname verwendet und die Feldnamen angepasst.

Eine rudimentäre Umgebung für ein Glossar.

\moreauthor \submitdate \supervisorII \supervisedIIby \submittedon
\dissertation
\theglossary (Umg.) \glossaryname \glossitem

File k: tudscr-comp.dtx
Die alte \texttt{tudbook}-Klasse stellt den Befehl \texttt{chapterpage} bereit. Mit diesem ist es möglich, das Verhalten der Kapitelseiten – welches durch die Option \texttt{chapterpage} gesteuert wird – temporär umzuschalten, also statt Kapitelseiten lediglich Überschriften zu setzen und umgekehrt. Dies ist typographisch durchaus zu hinterfragen, allerdings sollen die neuen Klassen möglichst kompatibel zu der alten sein, weshalb diese Funktionalität trotzdem implementiert wird. Der Befehl \texttt{chapterpage} setzt den globalen Schalter \texttt{if@tud@comp@chapterpage}. Der Befehl \texttt{tud@comp@chapterpage@set} setzt für Kapitel das komplementäre Verhalten zur eigentlich gewählten \texttt{chapterpage}-Option. Nach dem Setzen der entsprechenden Überschrift wird \texttt{tud@comp@chapterpage@set} nochmals aufgerufen, das Verhalten auf den ursprünglichen Zustand geschaltet und der globale Schalter \texttt{if@tud@comp@chapterpage} zurückgesetzt.

Da wie bereits beschrieben das Vorgehen äußerst fragwürdig ist, wird bei der Verwendung von \texttt{chapterpage} zumindest einmalig eine Warnung ausgegeben.
Hier erfolgt die notwendige Anpassung der internen Gliederungsbefehle.

15.4 Kompatibilität zu tudmathposter

Die hier bereitgestellten Optionen und Befehle werden durch die Klasse \texttt{tudmathposter} bzw. \texttt{tudposter} definiert.

\texttt{bluebg} (Opt.) Mit der Option \texttt{bluebg} kann der Hintergrund des Posters in HKS41 gesetzt werden.

\texttt{tudposter} (Seitenstil) Der Seitenstil \texttt{tudposter} wird als Alias für \texttt{empty.tudheadings} definiert.

Hier werden alle alten Feldbefehle auf die Pendants von TUD-Script gelegt und das Makro \texttt{\tud@split@author@list} um diese Befehle erweitert.
Für die Angabe einer Website wird \ webpage* verwendet, um etwaige Makros zur Formatierung nicht beachten zu müssen.

Die Makros zur Angabe von Logo-Dateien (\... logfile) werden auf die passenden TUD-Script-Befehle gelegt. Die Änderung der Makros, welche für die Einbindung der Logos verantwortlich sind (\... logo), wird allerdings nicht unterstützt.

Die Klasse \ tudmathposter definier zusätzliche Gliederungsbefehle, welche allerdings völlig willkürliche vertikale Abstände davor und danach verwendet. Es besteht keinerlei Ambition, diese genau abzubilden. Falls hier tatsächlich ein Anpassungsbedarf besteht, kann dies
mit den entsprechenden Mitteln von KOMA-Script (`\RedeclareSectionCommand`) nach dem Laden von `tudscrcomp` erfolgen.

Die neuen Gliederungsbefehle erschließen sich mir ohnehin nicht, wirken eher so, als ob diese entstanden sind, weil auf Anwendungsebene etwas nichts so funktioniert hat, wie gewollt und statt einer vernünftigen Ursachenforschung einfach ein eigenes Konstrukt erschaffen wurde, um unzulängliche Fähigkeiten zu umgehen. Sei's drum. Für die Umsetzung werden die Basisgliederungsbefehle geklont und anschließend der gewünschte vertikale Abstand vor der Überschrift entfernt.

596 \newcounter{topsection}
597 \let\c@topsection\c@section
598 \newcommand*{topsectionnumdepth}{\sectionnumdepth}
599 \newcommand*{\scr@topsection@sectionindent}{\section@sectionindent}
600 \newcommand*{\scr@topsection@beforeskip}{\section@beforeskip}
601 \newcommand*{\scr@topsection@afterskip}{\section@afterskip}
602 \newcommand*{\sectiontocdepth}{\sectiontocdepth}
603 \newcommand*{\scr@topsection@tocindent}{\section@tocindent}
604 \newcommand*{\scr@topsection@tocnumwidth}{\section@tocnumwidth}
605 \DeclareNewSectionCommand[%
606 style=section,%
607 font={\usekomafont{section}},
608 beforeskip=\z@,%
609 ]{topsection}
610 \newcounter{topsubsection}
611 \let\c@topsubsection\c@subsection
612 \newcommand*{topsubsectionnumdepth}{\subsectionnumdepth}
613 \newcommand*{\scr@topsubsection@sectionindent}{\subsection@sectionindent}
614 \newcommand*{\scr@topsubsection@beforeskip}{\subsection@beforeskip}
615 \newcommand*{\scr@topsubsection@afterskip}{\subsection@afterskip}
616 \newcommand*{\subsectiontocdepth}{\subsectiontocdepth}
617 \newcommand*{\scr@topsubsection@tocindent}{\subsection@tocindent}
618 \newcommand*{\scr@topsubsection@tocnumwidth}{\subsection@tocnumwidth}
619 \DeclareNewSectionCommand[%
620 style=section,%
621 font={\usekomafont{subsection}},
622 beforeskip=\z@,%
623 ]{topsubsection}

\centersection
\centersubsection
\topcentersection
\topcentersubsection
\tud@comp@centersection


626 \newcommand*{\centersection[2][\@nil]}{%
627 \tud@comp@centersection\section{#1}{#2}%
628 }
629 \newcommand*{\centersubsection[2][\@nil]}{%
630 \tud@comp@centersection\subsection{#1}{#2}%
631 }
632 \newcommand*{\topcentersection[2][\@nil]}{%
633 \tud@comp@centersection\topsection{#1}{#2}%
634 }
635 \newcommand*{\topcentersubsection[2][\@nil]}{%
636 \tud@comp@centersection\topsubsection{#1}{#2}%
637 }
Alle Gliederungsbefehle nutzen die passenden Ebenen mit der standardmaßig Ausrichtung und passen kurzfristig \raggedsection an. Dabei auch darauf geachtet, ob das optionale Argument durch den Anwender verwendet wird.

\sectiontopskip \subsectiontopskip

Die Klasse \texttt{tudmathposter} stellt zu den neuen Gliederungsbefehlen noch zusätzlich die beiden Makros \texttt{\sectiontopskip} und \texttt{\subsectiontopskip} bereit. Diese werden in darin bei der Definition dieser Gliederungsebenen verwendet. Da die TUD-Script-Klassen den Ansatz verfolgen, dem Benutzer weitestgehende Freiheiten bei der Gestaltung des Layouts einzuräumen, werden diese an die Einstellungen der zentralen KOMA-Script-Benutzerschnittstelle gekoppelt.

\sectiontopskip \subsectiontopskip

\fusszeile

Mit diesen beiden Befehlen lässt sich der Inhalt des Fußbereiches angepasst. Wird in einem und/oder beiden Argumente von \texttt{\footcolumn} ein Stern * verwendet, so bleibt der bisherige Inhalt erhalten.

\fusszeile

Mit der Option \texttt{tudmathposterfoot} kann die Darstellung des Fußes im Poster angepasst werden. Die Klasse \texttt{tudmathposter} setzt den Fußbereich in zwei asymmetrischen Spalten, wohingegen die TUD-Script-Klassen diesen zentriert und symmetrisch zum Satzspiegel platzieren.
Die Option cdfoot wird um Werte für die Einstellung des Posterfußes erweitert.

\newcommand*{\tud@comp@mathposterwidth}{4.2\tud@len@widemargin}
\patchcmd{\tud@footcontent@use}{\hsize=\dimexpr#2-\@tempdimc\relax%}{\if@tud@mathposterfoot\hsize=\dimexpr\tud@comp@mathposterwidth\relax%\else\hsize=\dimexpr(#2-\columnsep)/2\relax%\fi%}{\tud@patch@wrn{tud@footcontent@use}}{}{	ud@patch@wrn{tud@footcontent@use}}

Um die Option tudmathposterfoot umzusetzen, muss für die Ausgabe des Fußbereichs eine Anpassung des Makros \tud@footcontent@use erfolgen. Normalerweise wird der Fußbereich in zwei gleichbreite Spalten über die komplette Textbreite aufgeteilt. Dahingegen werden durch die Klasse tudmathposter beide Fußspalten nicht über den kompletten Textbereich verteilt, sondern es verbleibt etwas ungenutzter Platz rechts davon, welcher für ein etwaiges Logo verwendet wird. Dieses Verhalten kann über die Option tudmathposterfoot aktiviert werden und wird hier nachgebildet.
Weiterhin stellt die Klasse `tudmathposter` zwei Umgebungen bereit, mit denen sich Tabellen und Bilder an einer bestimmten Stelle exakt platzieren lassen. Prinzipiell könnte hierfür auch das Paket `float` zum Einsatz kommen.

Außerdem wird eine Umgebung für farbige Tabellen sowie zwei Umschaltbefehle für die farbliche Ausprägung der Tabellenzeilen definiert.

\ifundef{\schnittrand}{}{\ifxlen{\schnittrand}{\edef\@tempa{\paperwidth=\the\dimexpr\paperwidth+\schnittrand*2\relax,\paperheight=\the\dimexpr\paperheight+\schnittrand*2\relax,\layout=\the\paperwidth:\the\paperheight,\layoutoffset=\the\dimexpr\schnittrand\relax,\showcrop}}\PackageErrorNoLine{tudscrcomp}{Wrong definition for \string\schnittrand}{The definition of \string\schnittrand does not expand to a valid dimension expression.}}%


\AtEndPreamble{%\RequirePackage{amsmath}[2016/06/28]%\RequirePackage{amsfonts}[2013/01/14]%\RequirePackage{calc}[2014/10/28]%\RequirePackage{textcomp}[2016/06/19]%\RequirePackage{tabularx}[2016/02/03]}

16 Das Paket fix-tudscrfonts – Schriften für alte TUD-Klassen

Das Paket \texttt{fix-tudscrfonts} bietet die Schriftfamilien des Corporate Designs der Technischen Universität Dresden im TUD-Script-Stil für die „veralteten“ Klassen von Klaus Bergmann sowie die Posterklassen an.

Sämtliche Optionen werden an das Paket \texttt{tudscrfonts} durchgereicht, welches automatisch am Ende der Präambel geladen wird, falls dies zuvor durch den Anwender geschieht. Für die Klassen von Klaus Bergmann, TUD-Script in der Version v1.0 sowie die alternativen
Posterklassen bleiben die vertikalen Abstände schriftgrößenunabhängig. Außerdem wird für die fette Schriftstärke ein stärkerer Schriftschnitt genutzt.

\PassOptionsToPackage{relspacing=false,cdfont=ultrabold}{tudscrfonts}
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{tudscrfonts}}
\ProcessOptions\relax
\AtEndPreamble{%
\tud@fontsfix@maybe@err%
\RequirePackage{tudscrfonts}\[\TUD@Version]%
}

16.1 Verwendbarkeit und Ladezeitpunkt von fix-tudscrfonts

Da das Paket eine Vielzahl an Anpassungen der Schriftbefehle vornimmt sowie die Definition von \LaTeX-S-Standardbefehlen bereits vor dem Laden der eigentlichen Dokumentklasse sichern muss, kann dieses Paket ausschließlich vor \documentclass mit \RequirePackage geladen werden.

\tud@fontsfix@maybe@err

Dieser Befehl wird nach dem Laden einer unterstützten Klasse zu \relax gesetzt. Sollte das Paket jedoch erst nach einer Dokumentklasse geladen werden, so wird ein Fehler ausgegeben.

\ifx\usepackage\RequirePackage
\newcommand*\tud@fontsfix@maybe@err{%
\PackageError{fix-tudscrfonts}{Package too late}{% You must load `fix-tudscrfonts' with\MessageBreak\string\RequirePackage' before any document class.%}
}%
\else
\newcommand*\tud@fontsfix@maybe@err{%
\PackageError{fix-tudscrfonts}{Unsupported class found}{% You have to use `fix-tudscrfonts' only\MessageBreak with supported classes. These are: `tudbook',\MessageBreak `tudbeamer', `tudfax', `tudform', `tudhaus',\MessageBreak and `tudletter' as well as `tudposter'\MessageBreak and `tudmathposter'. Otherwise it's adequate\MessageBreak to use package `tudscrfonts' as usual.%}
}%
\fi

Wird das Paket mit einer nicht unterstützten Dokumentklasse verwendet, wird ebenfalls ein Fehler ausgegeben.

16.2 Patches für die unterstützten Klassen

Für alle unterstützten Klassen sind vor und nach dem Laden einige Anpassungen an den bereitgestellten Schriftbefehlen notwendig.

File k: tudscr-comp.dtx
Mit diesen beiden Befehlen werden vor dem Laden der alten Klassen einige Befehle gesichert und die benötigten Optionen gesetzt. Nach dem Laden der jeweiligen Klasse werden die gesicherten Makrodefinitionen wiederhergestellt und einige Schriftdefinitionen angepasst. Bei den alten TUD-Klassen betrifft dies in erster Linie die Mathematiksschriften sowie die in den Klassen definierten, fixen Schriften.

\newcommand{\tud@fontsfix@AdaptClass}[2]{% 
  BeforeClass{#1}{% 
    \let\tud@fontsfix@maybe@err\relax% 
    \tud@cmd@store{DeclareFixedFont}% 
    \renewcommand{\DeclareFixedFont}[6]{}% 
    \tud@cmd@store{rmdefault}% 
    \tud@cmd@store{sfdefault}% 
    \tud@cmd@store{ttdefault}% 
    \tud@cmd@store{bfdefault}% 
    \tud@cmd@store{mddefault}% 
    \tud@cmd@store{itdefault}% 
    \tud@cmd@store{sldefault}% 
    \tud@cmd@store{scdefault}% 
    \tud@cmd@store{updefault}% 
    \tud@cmd@store{rmfamily}% 
    \tud@cmd@store{sffamily}% 
    \tud@cmd@store{ttfamily}% 
    \tud@cmd@store{familydefault}% 
    \tud@cmd@store{seriesdefault}% 
    \tud@cmd@store{shapedefault}% 
    \tud@cmd@store{normalfont}% 
    \let\normalfont\relax% 
    \PassOptionsToClass{serifmath}{#1}% 
  }% 
  AfterClass{#1}{% 
    \RequirePackage{tudscrbase}{\TUD@Version}% 
    \tud@cmdrestore{DeclareFixedFont}% 
    \tud@cmdrestore{rmdefault}% 
    \tud@cmdrestore{sfdefault}% 
    \tud@cmdrestore{ttdefault}% 
    \tud@cmdrestore{bfdefault}% 
    \tud@cmdrestore{mddefault}% 
    \tud@cmdrestore{itdefault}% 
    \tud@cmdrestore{sldefault}% 
    \tud@cmdrestore{scdefault}% 
    \tud@cmdrestore{updefault}% 
    \tud@cmdrestore{rmfamily}% 
    \tud@cmdrestore{sffamily}% 
    \tud@cmdrestore{ttfamily}% 
    \tud@cmdrestore{familydefault}% 
    \tud@cmdrestore{seriesdefault}% 
    \tud@cmdrestore{shapedefault}% 
    \tud@cmdrestore{normalfont}% 
    #2%,% 
  }% 
  AfterPackage{tudscrfonts}{% 
    \def{\dinBold}{\dinnb}% 
    \def{\dinfamily}{\dinnb}% 
    \AtBeginDocument{% 
    \newcommand{\univLightVII}% 
    \newcommand{\univLightObliqueVII}% 
    \newcommand{\univBoldVII}% 
    \newcommand{\univLightIX}% 
    \newcommand{\univLightObliqueIX}% 
    \newcommand{\univBoldIX}% 
    \newcommand{\univLightXI}% 
    \newcommand{\univLightObliqueXI}% 
    \newcommand{\univBoldXI}% 
    \newcommand{\univLightXV}% 
    \newcommand{\univLightObliqueXV}% 
    \newcommand{\univBoldXV}% 
    %}
\newcommand*{\univBoldIX}{\univBoldVII}\% 
\newcommand*{\univLightXI}{\univLightVII}\% 
\newcommand*{\univLightObliqueXI}{\univLightObliqueVII}\% 
\newcommand*{\univBoldXI}{\univBoldVII}\% 
\newcommand*{\univLightXV}{\univLightVII}\% 
\newcommand*{\univLightObliqueXV}{\univLightObliqueVII}\% 
\newcommand*{\univBoldXV}{\univBoldVII}\% 
\newcommand*{\univLightHead}{\univLightVII}\% 
\newcommand*{\univBoldHead}{\univBoldVII}\% 
\providecommand*{\tud@head@fntsz}{9}\% 
\if@tud@univers\%
  \if@tud@x@fontspec@enabled\%
    \DeclareFixedFont*{\univLightVII}{\encodingdefault}{UniversLight}{m}{n}{7}\%
    \DeclareFixedFont*{\univLightObliqueVII}{\encodingdefault}{UniversLightOblique}{m}{n}{7}\%
    \DeclareFixedFont*{\univBoldVII}{\encodingdefault}{UniversBold}{m}{n}{7}\%
    \DeclareFixedFont*{\univLightIX}{\encodingdefault}{UniversLight}{m}{n}{9}\%
    \DeclareFixedFont*{\univLightObliqueIX}{\encodingdefault}{UniversLightOblique}{m}{n}{9}\%
    \DeclareFixedFont*{\univBoldIX}{\encodingdefault}{UniversBold}{m}{n}{9}\%
    \DeclareFixedFont*{\univLightXV}{\encodingdefault}{UniversLight}{m}{n}{15}\%
    \DeclareFixedFont*{\univLightObliqueXV}{\encodingdefault}{UniversLightOblique}{m}{n}{15}\%
    \DeclareFixedFont*{\univBoldXV}{\encodingdefault}{UniversBold}{m}{n}{15}\%
    \DeclareFixedFont*{\univLightHead}{\encodingdefault}{UniversLight}{m}{n}{9}\%
  \else\%
    \DeclareFixedFont*{\univLightVII}{\encodingdefault}{\tud@univers}{l}{n}{7}\%
    \DeclareFixedFont*{\univLightObliqueVII}{\encodingdefault}{\tud@univers}{l}{sl}{7}\%
    \DeclareFixedFont*{\univBoldVII}{\encodingdefault}{\tud@univers}{b}{n}{7}\%
    \DeclareFixedFont*{\univLightIX}{\encodingdefault}{\tud@univers}{l}{n}{9}\%
    \DeclareFixedFont*{\univLightObliqueIX}{\encodingdefault}{\tud@univers}{l}{sl}{9}\%
    \DeclareFixedFont*{\univBoldIX}{\encodingdefault}{\tud@univers}{b}{n}{9}\%
    \DeclareFixedFont*{\univLightXV}{\encodingdefault}{\tud@univers}{l}{n}{15}\%
    \DeclareFixedFont*{\univLightObliqueXV}{\encodingdefault}{\tud@univers}{l}{sl}{15}\%
    \DeclareFixedFont*{\univBoldXV}{\encodingdefault}{\tud@univers}{b}{n}{15}\%
    \DeclareFixedFont*{\univLightHead}{\encodingdefault}{\tud@univers}{l}{n}{9}\%
  \fi\%
\fi\%
\else\%
  \DeclareFixedFont*{\univLightVII}{\encodingdefault}{\tud@univers}{l}{n}{7}\%
  \DeclareFixedFont*{\univLightObliqueVII}{\encodingdefault}{\tud@univers}{l}{sl}{7}\%
  \DeclareFixedFont*{\univBoldVII}{\encodingdefault}{\tud@univers}{b}{n}{7}\%
  \DeclareFixedFont*{\univLightIX}{\encodingdefault}{\tud@univers}{l}{n}{9}\%
  \DeclareFixedFont*{\univLightObliqueIX}{\encodingdefault}{\tud@univers}{l}{sl}{9}\%
  \DeclareFixedFont*{\univBoldIX}{\encodingdefault}{\tud@univers}{b}{n}{9}\%
  \DeclareFixedFont*{\univLightXV}{\encodingdefault}{\tud@univers}{l}{n}{15}\%
  \DeclareFixedFont*{\univLightObliqueXV}{\encodingdefault}{\tud@univers}{l}{sl}{15}\%
  \DeclareFixedFont*{\univBoldXV}{\encodingdefault}{\tud@univers}{b}{n}{15}\%
  \DeclareFixedFont*{\univLightHead}{\encodingdefault}{\tud@univers}{l}{n}{9}\%
\fi\%
\fi\%
\fi\%

File k: tudscr-comp.dtx 230
Nach der Definition der notwendigen Befehle erfolgt nun die Umsetzung für alle unterstützten Klassen.

\tud@fontsfix@AdaptClass{tudletter}{}
\tud@fontsfix@AdaptClass{tudfax}{}
\tud@fontsfix@AdaptClass{tudform}{}
\tud@fontsfix@AdaptClass{tudhaus}{}
\tud@fontsfix@AdaptClass{tudposter}{}
\tud@fontsfix@AdaptClass{tudbeamer}{}

Wird das Paket fix-tudscrfonts mit den alten TUD-Klassen tudbook, tudmathposter etc. oder TUD-Script in der Version v1.0 verwendet, müssen einige Kompatibilitätseinstellungen vorgenommen werden. Hierfür sind wenige Patches notwendig.
16.3 Spezielle Patches für die Klasse tudbook

Die notwendigen Anpassungen der Klasse \texttt{tudbook} betreffen die Schriftauswahl bei Überschriften.

Die Überschriften sollen abhängig von der Option \texttt{cdfont} und nicht immer zwingend in \texttt{DIN BOLD} gesetzt werden.

\begin{verbatim}
\@makechapterhead
\@makeschapterhead
\section
\showtitle
979 \tud@fontsfix@AdaptClass{tudbook}{%
980 \patchcmd{\@makechapterhead}{\dinBold\Huge\bfseries}{%
981 \tud@sec@fontface[\bfseries]%
982 \if@color\color{HKS41-100}\fi%
983 \Huge%
984 }{%\{\tud@patch@wrn{\@makechapterhead}\}%
985 \patchcmd{\@makechapterhead}{\MakeUppercase}{%
986 \tud@makeuppercase%
987 }{%\{\tud@patch@wrn{\@makechapterhead}\}%
988 \patchcmd{\@makechapterhead}{\dinBold\Huge\bfseries}{%
989 \tud@sec@fontface[\bfseries]%
990 \if@color\color{HKS41-100}\fi%
991 \Huge%
992 }{%\{\tud@patch@wrn{\@makechapterhead}\}%
993 \patchcmd{\@makechapterhead}{\MakeUppercase}{%
994 \tud@makeuppercase%
995 }{%\{\tud@patch@wrn{\@makechapterhead}\}%
996 \patchcmd{\section}{\dinBold\Large\bfseries\MakeUppercase}{%
997 \tud@sec@fontface[\bfseries]%
998 \if@color\color{HKS41-100}\fi%
999 \Large%
1000 \tud@makeuppercase%
1001 }{%\{\tud@patch@wrn{\section}\}%
1002 \patchcmd{\showtitle}{\dinBold\Huge\bfseries\MakeUppercase}{%
1003 \tud@sec@fontface[\bfseries]%
1004 \if@colortitle\color{HKS41-30}\fi%
1005 \Huge%
1006 \tud@makeuppercase%
1007 }{%\{\tud@patch@wrn{\showtitle}\}%
1008 }
\end{verbatim}

16.4 Spezielle Patches für die Klasse tudmathposter

Auch für die Posterklasse \texttt{tudmathposter} sowie das Paket \texttt{tudfonts} sind kleine Anpassungen notwendig.

\begin{verbatim}
title (Schriftel.)
\maketitle
\subtitlefont
\preprocesstitle
\sectionfont
\subsectionfont
\ps@tudposter
\if@tudfonts@nodin
\end{verbatim}

Es werden einige Einstellungen für die Überschriften angepasst.

\begin{verbatim}
1009 \tud@fontsfix@AdaptClass{tudmathposter}{%
1010 \def\raggedtitle{\tud@raggedright}%
1011 \renewcommand*\raggedpart{\tud@raggedright}%
1012 \renewcommand*\raggedsection{\tud@raggedright}%
1013 \setkomafont{title}{\tud@sec@fontface[\bfseries]\Huge}%
1014 \CheckCommand\maketitle{%
1015 \if@matheanull
1016 \setlength\@tempskipa{31.194586mm-\topsep}%
1017 \else
1018 \setlength\@tempskipa{33.02mm-\topskip}%
\end{verbatim}

File k: tudscr-comp.dtx 232
Für die Schriftstärke in der Kopfzeile muss der Seitenstil angepasst und anschließend nochmals akiviert werden.

Die Klasse \tudmathposter lädt für die Einstellungen der Schriften das Paket \tudfonts. Wenn \tudscrfonts zum Einsatz kommen soll, ist das unerwünscht, weshalb das Laden hiermit unterbunden wird.
Für die Verwendung des Paketes fix-tudscrfonts mit TUD-Script in der Version v1.0 sind ebenfalls einige Anpassungen notwendig. Es muss Sorge dafür getragen werden, dass einige veraltete Befehle aktualisiert werden. Hierfür müssen diese „unschädlich“ gemacht werden, noch bevor das Paket selbst in Erscheinung tritt.

\BeforePackage{tudscrfonts}{%
\undef\univln%
\undef\univrn%
\undef\univbn%
\undef\univxn%
\undef\univls%
\undef\univrs%
\undef\univbs%
\undef\univxs%
\undef\dinbn%
\undef\textuln%
\undef\texturn%
\undef\textubn%
\undef\textuxn%
\undef\textuls%
\undef\texturn%
\undef\textubs%
\undef\textuxs%
\undef\textdbn%
%}
Teil I

tudscr-misc.dtx

17 Verschiedenes für das TUD-Script-Bundle

Alles, wofür sich eine separate Datei nicht lohnt, landet hier.

17.1 Sukzessives Abarbeiten von Layoutoptionen im Dokument

Werden mit \TUDoptions nach \begin{document} mehrere Layoutoptionen gleichzeitig angegeben, muss dafür Sorge getragen werden, dass diese in der korrekten Reihenfolge abgearbeitet werden.

Das Makro \TUD@SpecialOptionAtDocument kann innerhalb von \TUD@key verwendet werden, um einen im obligatorischen Argument angegebenen Schalter zu aktivieren. Dabei wird über \tud@atdocument@hook garantiert, dass der Befehl \tud@atdocument@process, welcher für die Abarbeitung der einzelnen Makros in der richtigen Reihenfolge verantwortlich ist, lediglich einmal über \AtEndOfFamilyOptions ausgeführt wird.

\begin{alltt}
\texttt{%execute & (class&!inherit | package & fonts)}
\newcommand*\TUD@SpecialOptionAtDocument[1]{% \texttt{%execute & (class&!inherit | package & fonts)}
\let\tud@atdocument@hook\relax
\AtBeginDocument{%}
\renewcommand*\TUD@SpecialOptionAtDocument[1]{% \texttt{%execute & (class&!inherit | package & fonts)}
\ifx\tud@atdocument@hook\relax% \texttt{%execute & (class&!inherit | package & fonts)}
\gdef\tud@atdocument@hook\{\global\let\tud@atdocument@hook\relax\}% \texttt{%execute & (class&!inherit | package & fonts)}
\AtEndOfFamilyOptions{% \texttt{%execute & (class&!inherit | package & fonts)}
\tud@atdocument@hook%
\tud@atdocument@process%
}\texttt{%execute & (class&!inherit | package & fonts)}
\fi%
\gappto\tud@atdocument@hook\{\global\booltrue{@#1}%
\}
\}
\end{alltt}

\begin{alltt}
\texttt{\tud@atdocument@process %}
\texttt{\if\tud@font@set %}
\texttt{\if\tud@font@math@set %}
\texttt{\if\tud@font@skip@set %}
\texttt{\if\tud@x@scr@headings@reset %}
\texttt{\if\tud@layout@process %}
\texttt{\if\tud@cdgeometry@process %}
\texttt{\if\tud@cdgeometry@process %}
\texttt{\newif\if\tud@font@set %}
\texttt{\newif\if\tud@font@math@set %}
\texttt{\newif\if\tud@font@skip@set %}
\texttt{\newif\if\tud@x@scr@headings@reset %}
\texttt{\newif\if\tud@layout@process %}
\texttt{\newif\if\tud@cdgeometry@process %}
\texttt{\newif\if\tud@cdgeometry@process %}
\texttt{\newif\if\tud@font@set %}
\texttt{\newif\if\tud@font@math@set %}
\texttt{\newif\if\tud@font@skip@set %}
\texttt{\newif\if\tud@x@scr@headings@reset %}
\texttt{\newif\if\tud@layout@process %}
\texttt{\newif\if\tud@cdgeometry@process %}
\texttt{\newif\if\tud@cdgeometry@process %}
\end{alltt}

Das Makro \tud@atdocument@process wird im Dokument ausgeführt, wenn mindestens eine Option gesetzt wurde, welche \TUD@SpecialOptionAtDocument verwendet. Je nachdem, welche Schalter insgesamt aktiviert wurden, werden die dazugehörigen Befehle für die Einstellungen von Schriften, Layout und/oder Satzspiegel in der richtigen Reihenfolge ausgeführt.

File l: tudscr-misc.dtx 235
17.2 Papierformat und Schriftgröße

Insbesondere für Poster aber auch für alle anderen TUD-Script-Klassen wird überprüft, ob
nach einer Änderung des Standardpapierformates auch die Schriftgröße durch den Anwender
angepasst wurde. Diese wird jedoch nicht auf Plausibilität geprüft, da der Aufwand hierfür
relativ hoch wäre. Vielmehr wird davon ausgegangen, dass die explizite Angabe der Schrift-
größe bewusst und dem Papierformat sowie den verwendeten Textspalten entsprechend
erfolgt.

Die nachfolgend definierten Optionen und Befehle orientieren sich dabei stark an internen
KOMA-Script-Makros.

\begin{itemize}
\item \textit{paper} (Opt.)
\end{itemize}

Da die KOMA-Script-Option \textit{paper} eine Menge an Einstellmöglichkeiten bietet, müssen die
relevanten hier abgefangen werden.
Sowohl ISO/DIN-Formate...

...als auch frei gewählte Papierformate werden erkannt.

Die Standardpapiergröße spielt allerdings keine Rolle für eine Warnung.

Bei der Schriftgröße ist das bloße Erkennen und Weiterreichen der Option an die Klasse ausreichend.

Die Schlüssel sollen lediglich beim Laden der Dokumentklasse die angegebenen Optionen abfangen. Wurde das Papierformat jedoch nicht die Schriftgröße geändert, wird nun eine Warnung erzeugt.
Um die Änderung des Papierformats auch über die \LaTeX-X-Standardoptionen wie beispielsweise a5paper abfangen zu können, müssen auch die äquivalent zu KOMA-Script ausgewertet werden.

\newcommand*{\tud@x@scr@paper}{\expandafter{\tud@x@scr@@paper\CurrentOption paper\tud@x@scr@@paper}}
\newcommand*{\tud@x@scr@@paper}{}
\def{\tud@x@scr@@paper #1#2paper#3\tud@x@scr@@paper}{\@tempswafalse\ifstr{#3}{paper}{\ifnumber{#2}{\ifstr{#1}{a}{\@tempswatrue}{\ifstr{#1}{b}{\@tempswatrue}{\ifstr{#1}{c}{\@tempswatrue}{\ifstr{#1}{d}{\@tempswatrue}{}}}}}{}}}{}\if@tempswa{\TUDExecuteOptions{#3=#1#2}}\fi\PassOptionsToClass{\CurrentOption}{\TUD@Class@KOMA}}

17.3 Sprungmarken

\if@tud@bookmarks
Wird das Paket hyperref geladen, so kann die Option tudbookmarks genutzt werden, um zu steuern, ob für Titel und Inhaltsverzeichnis automatisch ein Eintrag für die Sprungmarken erzeugt werden soll.

\newif{\if@tud@bookmarks}
\TUD@ifkey{tudbookmarks}{@tud@bookmarks}{@tud@bookmarks}
\tudbookmark
\tud@x@hyperref@realfootnotes

Die Befehle zum Eintragen der Sprungmarken. Im Dokument kann \tudbookmark auch vom Anwender genutzt werden, um weitere Lesezeichen abhängig von der Option tudbookmarks manuell zu erzeugen.

Für die Titelseite werden – wie auch durch das Paket hyperref – die Fußnoten mit dem Befehl \tud@x@hyperref@realfootnotes auf den originalen Zustand zurückgesetzt.

\newcommand*{\tudbookmark[1][3]}{\@gobbletwo}
\newcommand*{\tud@x@hyperref@realfootnotes}{}
\AfterPackage{hyperref}{\newcommand*{\tudbookmark[3][3]}{\@gobblethree}}
\newcommand*{\tud@x@hyperref@realfootnotes}{\relax\if@tud@bookmarks\phantomsection\ifblank{#1}{\pdfbookmark{#2}{#3}}{\pdfbookmark[#1]{#2}{#3}}}
Wird das Paket \texttt{bookmark} geladen, können die erzeugten Outlines gezielt auf der obersten Ebene erzeugt werden.

\newcommand*{\tud@x@bookmark@startatroot}{\relax}
\AfterPackage{bookmark}{\renewcommand*{\tud@x@bookmark@startatroot}{\if@tud@bookmarks\bookmarksetup{startatroot}\fi}}
\pdfstringdefDisableCommands{\let\NoCaseChange\@firstofone}
\tud@x@bookmark@startatroot

\texttt{titlepage (Umg.)}
\texttt{\@maketitle}
\texttt{\tud@make@titlehead}

Ist die Option \texttt{tudbookmarks} aktiviert, werden für Umschlag- und Titelseite PDF-Lesezeichen bzw. Outline-Einträge erzeugt.

\pretocmd{\tud@make@titlehead}{\tud@x@bookmark@startatroot}{\tudbookmark[⟨book | report⟩]{\titlename}{title}}{\tud@patch@wrn{tud@make@titlehead}}
\pretocmd{\@maketitle}{\tud@x@bookmark@startatroot}{\tudbookmark[⟨book | report⟩]{\titlename}{title}}{\tud@patch@wrn{@maketitle}}

Dies geschieht außerdem auch für die Titelköpfe.

\pretocmd{\tud@make@titlehead}{\tud@x@bookmark@startatroot}{\tudbookmark[⟨book | report⟩]{\titlename}{title}}{\tud@patch@wrn{tud@make@titlehead}}
\pretocmd{\@maketitle}{\tud@x@bookmark@startatroot}{\tudbookmark[⟨book | report⟩]{\titlename}{title}}{\tud@patch@wrn{@maketitle}}
Für die Verzeichnisse geschieht dies auch. Für das Inhaltsverzeichnis ist hierfür etwas mehr Aufwand notwendig. Für dieses wird im Normalfall keine PDF-Outline erzeugt. Deshalb ist hier ein wenig Trickerei notwendig, um zwar einen Outline-Eintrag für das Inhaltsverzeichnis zu erzeugen, dieses dabei aber nicht in sich selbst einzutragen.

17.4 Warnung bei der Nutzung des Paketes tocloft

Das Paket tocloft verursacht allerlei Probleme. Wird es geladen, so erscheint eine entsprechende Warnung.

17.5 Fußnoten in Überschriften

Die KOMA-Script-Option footnotes wird um den Schlüssel symbolheadings erweitert, mit welchem eingestellt werden kann, ob in den Überschriften Symbole für die Fußnoten anstelle von Zahlen verwendet werden sollen.
Bei aktivierter Option `footnotes=symbolheadings` wird dieser Zähler für die Symboleauswahl von Fußnoten in Überschirften inkrementiert.

\AtEndOfClass{\chapter}{\newcounter{symbolheadings}}

Damit Fußnoten nicht automatisch in Majuskeln gesetzt werden, wird der interne Befehl `\@uclcnotmath` aus dem Paket `textcase` angepasst.

\CheckCommand*{\@uclcnotmath}[4]{%  
\begin{group}
  \#1\%
  \def\{%{\let\%{\def\NoCaseChange##1{\noexpand\NoCaseChange{\noexpand##1}}}\%  
  \@nonchangecase\label
  \@nonchangecase\ref
  \@nonchangecase\ensuremath
  \def\cite##1##{\toks@{\noexpand\cite##1}\@citex}{%  
  \def\@citex##1{\NoCaseChange{\the\toks@{##1}}}%
  \def\reserved@a##1##2{\let\##2\reserved@a}{%  
  \expandafter\reserved@a\@uclclist\reserved@b{\reserved@b\@gobble}{%  
  \protected@edef\reserved@a{\endgroup\noexpand\@skipmath#3#4$\valign$}{%  
  \reserved@a}{%  
\newcommand*{\tud@x@textcase@uclcnotmath}{%  
\def\footnote##1##{\toks@{\noexpand\footnote##1}\@footnote}{%  
\def\@footnote##1{\NoCaseChange{\the\toks@{##1}}}%
\@mkboth##1##2{\NoCaseChange{\@mkboth{##1}{##2}}}%
\patchcmd{\@uclcnotmath}{\@nonchangecase\ensuremath}{\@nonchangecase\ensuremath\tud@x@textcase@uclcnotmath}{%  
17.6 Durchreichen von Optionen und Standardoptionen

Durchreichen aller Klassenoptionen an die KOMA-Script-Klasse bzw. an die genutzte TUD-Script-Elternklasse. Für Klassen wird vor dem Durchreichen noch geprüft, ob ein spezielles Papierformat angegeben wurde.

\DeclareOption*{\tud@x@scr@paper}{%  
\newcommand*{\tud@x@textcase@uclcnotmath}{%  
\newcommand*{\@uclcnotmath}{\@nonchangecase\ensuremath\tud@x@textcase@uclcnotmath}{%  
\newcommand*{\@uclcnotmath}{\@nonchangecase\ensuremath}{%  
\newcommand*{\@uclcnotmath}{\@nonchangecase\ensuremath\tud@x@textcase@uclcnotmath}{%  
257 \patchcmd{\@uclcnotmath}{\@nonchangecase\ensuremath\tud@x@textcase@uclcnotmath}{\@nonchangecase\ensuremath}{%  
258 \@nonchangecase\ensuremath\tud@x@textcase@uclcnotmath}{%  
259}%}{%\tud@patch@wrn{\@uclcnotmath}}

File l: tudscr-misc.dtx
Es werden die Standardoptionen ausgeführt. Für die Klasse \texttt{tudscrposter} werden die Farben aktiviert.

\begin{verbatim}
266 (*class&!inherit)
267 \TUDExecuteOptions{%
268 (book | report | article)  cd=true,relspacing=true,tudbookmarks=true%
269 (poster)  cd=bicolor,relspacing=true,tudbookmarks=false,cdfont=ultrabold%
270 }
271 (/class&!inherit)
272 (*package & tutorial)
273 \KOMAoptions{headings=small}
274 \TUDoptions{cdfoot=true}
275 \TUDExecuteOptions{ToDo=true}
276 (/package & tutorial)
277 \TUDProcessOptions\relax
\end{verbatim}

Die korrespondierende KOMA-Script-Klasse bzw. TUD-Script-Elternklasse wird geladen.

\begin{verbatim}
278 (*class&!inherit)
279 \LoadClass{\TUD@Class@KOMA}[2015/04/23]
280 (/class&!inherit)
281 (*class & inherit)
282 \LoadClass{\TUD@Class@Parent}
283 (/class & inherit)
\end{verbatim}
18 Das Paket tudscrcolor – Die Farben des Corporate Designs

Das Corporate Design der Technischen Universität Dresden legt nicht nur die zu nutzenden Schriften und das Layout sondern auch die zu verwendenden Farben fest. Diese werden nachfolgend für das CMYK- und RGB-Farbmodel definiert. Sie können im Dokument mit sämtlichen Befehlen zur Farbauswahl wie \color{Farbe} oder \textcolor{Farbe} verwendet werden.

18.1 Optionen und Befehle

Das Paket kann entweder mit einfachen, für TUD-Script ausreichenden oder erweiterten Farbdefinitionen geladen werden. Letztere definieren zusätzliche Farbbezeichnungen für die Kompatibilität zu alten TUD-Klassen.

\newcommand*{\setcdcolors}[1]{%  
\selectcolormodel{#1}  
\tud@setcolors@do%}

Der Befehl \setcdcolors definiert die Farben des Corporate Designs. Das Argument dient zur Auswahl des gewünschten Farbmodells. Dies kann dazu genutzt werden, innerhalb des Dokumentes die Definition der Farben für ein neues Farbmodell zu ändern.

\newcommand*{\tud@color}[1]{%  \ifdefvoid{#1}{}{\color{#1}}%}

18.2 Farbdefinitionen

18.2.1 Notwendige Farben für TUD-Script

Die eigentliche Definitionen der Farben erfolgt erst durch die Ausführung des Befehls \tud@setcolors@do. Damit wird es möglich, auf die angegebenen Optionen in Form von \tud@setcolors@add oder den durch das Paket xcolor angegebenen Farbraum zu reagieren.

File m: tudscr-color.dtx
Die primäre Hausfarbe (dunkles Blau)
\definecolor{HKS41}{cmyk/RGB/rgb}{%
1.00, 0.70, 0.10, 0.50/011, 042, 081/0.0431372549, 0.31764705882%
}\%

Die sekundäre Hausfarbe (grau), allein und ausschließlich für die Verwendung in der Geschäftsausstattung und nicht für Fließtext, Grafiken etc.
\definecolor{HKS92}{cmyk/RGB/rgb}{%
0.10, 0.00, 0.05, 0.65/080, 089, 085/0.31372549019, 0.34901960784, 0.33333333333%
}\%

Auszeichnungen 1. Kategorie (helles Blau)
\definecolor{HKS44}{cmyk/RGB/rgb}{%
1.00, 0.50, 0.00, 0.00/000, 089, 163/0.63921568627%
}\%

Auszeichnungen 2. Kategorie (Indigo, Purpur, dunkles Grün, helles Grün)
\definecolor{HKS36}{cmyk/RGB/rgb}{%
0.80, 0.90, 0.00, 0.00/081, 041, 127/0.49803921568%
}\%
\definecolor{HKS33}{cmyk/RGB/rgb}{%
0.50, 1.00, 0.00, 0.00/129, 026, 120/0.63921568627%
}\%
\definecolor{HKS57}{cmyk/RGB/rgb}{%
1.00, 0.00, 0.90, 0.20/000, 122, 071/0.28235294117%
}\%
\definecolor{HKS65}{cmyk/RGB/rgb}{%
0.65, 0.00, 1.00, 0.00/034, 173, 054/0.21176470588%
}\%

Ausnahmefarbe (Orange)
\definecolor{HKS07}{cmyk/RGB/rgb}{%
0.00, 0.60, 1.00, 0.00/232, 123, 020/0.48235294117, 0.07843137254%
}\%

cddarkblue (Farbe)
cdgray (Farbe)
cdgrey (Farbe)
cdblue (Farbe)
cdindigo (Farbe)
cdpurple (Farbe)
cddarkgreen (Farbe)
cdgreen (Farbe)
cdorange (Farbe)

Die definierten Grundfarben werden zur einfacheren Verwendung im Dokument noch einmal speziell benannt.
\colorlet{cddarkblue}{HKS41}%
\colorlet{cdgray}{HKS92}%
\colorlet{cdgrey}{HKS92}%
\colorlet{cdblue}{HKS44}%
\colorlet{cdindigo}{HKS36}%
\colorlet{cdpurple}{HKS33}%
\colorlet{cddarkgreen}{HKS57}%
\colorlet{cdgreen}{HKS65}%
\colorlet{cdorange}{HKS07}%

Damit enden die notwendigen Farbdefinitionen für das \texttt{tudscr-color}-Paket. Abhängig von den gewählten Optionen werden mit \texttt{tud@setcolors@add} ggf. zusätzliche Farbnamen definiert.
\tud@setcolors@add%
\}

File m: tudscr-color.dtx
18.2.2 Zusätzliche Farben für alte TUD-Klassen

Die erweiterten Farbbefehle werden durch `tudscrcolor` definiert, wenn das Paket explizit mit der Option `oldcolors` geladen wird. Damit werden alle gängigen Farbdefinitionen der vielen Insellösungen des \LaTeX-Universums an der Technischen Universität Dresden unterstützt.

\begin{verbatim}
46 \newcommand*{\tud@setcolors@add}{}
47 \newcommand*{\tud@setcolors@old}{%
48 \colorlet{HKS41K10}{HKS41!10}%
49 \colorlet{HKS41K20}{HKS41!20}%
50 \colorlet{HKS41K30}{HKS41!30}%
51 \colorlet{HKS41K40}{HKS41!40}%
52 \colorlet{HKS41K50}{HKS41!50}%
53 \colorlet{HKS41K60}{HKS41!60}%
54 \colorlet{HKS41K70}{HKS41!70}%
55 \colorlet{HKS41K80}{HKS41!80}%
56 \colorlet{HKS41K90}{HKS41!90}%
57 \colorlet{HKS41K100}{HKS41!100}%
58 \colorlet{HKS92K10}{HKS92!10}%
59 \colorlet{HKS92K20}{HKS92!20}%
60 \colorlet{HKS92K30}{HKS92!30}%
61 \colorlet{HKS92K40}{HKS92!40}%
62 \colorlet{HKS92K50}{HKS92!50}%
63 \colorlet{HKS92K60}{HKS92!60}%
64 \colorlet{HKS92K70}{HKS92!70}%
65 \colorlet{HKS92K80}{HKS92!80}%
66 \colorlet{HKS92K90}{HKS92!90}%
67 \colorlet{HKS92K100}{HKS92!100}%
68 \colorlet{HKS44K10}{HKS44!10}%
69 \colorlet{HKS44K20}{HKS44!20}%
70 \colorlet{HKS44K30}{HKS44!30}%
71 \colorlet{HKS44K40}{HKS44!40}%
72 \colorlet{HKS44K50}{HKS44!50}%
73 \colorlet{HKS44K60}{HKS44!60}%
74 \colorlet{HKS44K70}{HKS44!70}%
75 \colorlet{HKS44K80}{HKS44!80}%
76 \colorlet{HKS44K90}{HKS44!90}%
77 \colorlet{HKS44K100}{HKS44!100}%
78 \colorlet{HKS36K10}{HKS36!10}%
79 \colorlet{HKS36K20}{HKS36!20}%
80 \colorlet{HKS36K30}{HKS36!30}%
81 \colorlet{HKS36K40}{HKS36!40}%
82 \colorlet{HKS36K50}{HKS36!50}%
83 \colorlet{HKS36K60}{HKS36!60}%
84 \colorlet{HKS36K70}{HKS36!70}%
85 \colorlet{HKS36K80}{HKS36!80}%
86 \colorlet{HKS36K90}{HKS36!90}%
87 \colorlet{HKS36K100}{HKS36!100}%
88 \colorlet{HKS33K10}{HKS33!10}%
89 \colorlet{HKS33K20}{HKS33!20}%
90 \colorlet{HKS33K30}{HKS33!30}%
91 \colorlet{HKS33K40}{HKS33!40}%
92 \colorlet{HKS33K50}{HKS33!50}%
93 \colorlet{HKS33K60}{HKS33!60}%
94 \colorlet{HKS33K70}{HKS33!70}%
95 \colorlet{HKS33K80}{HKS33!80}%
96 \colorlet{HKS33K90}{HKS33!90}%
97 \colorlet{HKS33K100}{HKS33!100}%
98 \colorlet{HKS57K10}{HKS57!10}%
\end{verbatim}

File m: tudscrcolor.dtx
18.3 Ausführung der Optionen

Zum Schluss werden die Optionen ausgeführt und ggf. an \texttt{xcolor} weitergereicht. Anschließend werden die Farben für das Dokument definiert. Ohne die Angabe eines optionalen
Argumentes an das Paket \texttt{xcolor} erfolgt die Definition für den gewählten bzw. standardmäßig eingestellten Farbraum.

209 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{xcolor}}
210 \ExecuteOptions{reduced}
211 \ProcessOptions\relax
212 \RequirePackage{xcolor}[2007/01/21]
213 \tud@setcolors@do\%

File m: \texttt{tudscr-color.dtx}
Das Paket `tudscrsupervisor` – Betreuung wissenschaftlicher Arbeiten

Diese Paket stellt für die TUD-Script-Klassen mehrere Umgebungen und Befehle zur Erstellung der Aufgabenstellung einer Abschlussarbeit sowie eines Gutachtens und eines Aushangs bereit.

Diesen beiden Makros dienen dazu, unterschiedliche Varianten für die Ausgabe innerhalb der nachfolgenden Umgebungen generieren zu können. Momentan werden diese nur innerhalb der `task`-Umbgebung verwendet.

Der Befehl \texttt{student} kann als Alias für \texttt{author} genutzt werden.

Der Befehl \texttt{tud@authortable@set} dient bei Aufgabenstellung und Gutachten zur Ausgabe einer Tabelle mit Informationen zum Autor beziehungsweise zu den Autoren.\footnote{Matrikelnummer, Jahrgang, Studiengang etc.}

Zu Beginn wird eine Tabelle mit den Bezeichnern aller genutzten Feldern ausgegeben. Danach folgen alle Autoren. Damit ein einheitliches Layout entsteht und auch die Tabellen am Ende der Umgebung in der ersten Spalte die gleiche Breite haben wie im oberen Teil, ist die Bestimmung einer festen Spaltenbreite notwendig, die so breit wie der längste Bezeichner ist. Dafür muss festgestellt werden, welche optionalen Felder denn nun überhaupt genutzt werden. Dafür wird \texttt{tud@multiple@fields@preset} mit \texttt{null} als Argument aufgerufen, um alle potenziellen Felder erkennen zu können.

Anschließend werden die Bezeichner sowohl der obligatorischen als auch der genutzten, optionalen Felder in \texttt{@tempa} gespeichert. Mit der Liste wird der längste Bezeichner bestimmt und dessen Länge in \texttt{tud@len@authortable} gespeichert.

Der Befehl \tud@split@author@do wird innerhalb der TUD-Script-Klassen zur formatierten Ausgabe mehrerer Autoren auf der Titelseite verwendet, welche durch \author{\langle Autor(en)\rangle} angegeben und mit \and getrennt wurden. Er wird hier auf die Ausgabe der Autoren mit den jeweils zusätzlich gegebenen Informationen in einer Tabelle angepasst.

Weil alle Autoren in einer Tabelle gesetzt werden wird geprüft, welche Felder individuell via \author angegeben wurden. Damit die Tabellen die gleiche Höhe haben, auch wenn für einen Autor ein Feld ausgelassen wurde, werden alle insgesamt angegebenen Felder mit via \tud@multiple@fields@preset mit \null initialisiert. Anschließend werden die für den aktuellen Autor angegebenen Felder gesetzt.
Nach viel Geplänkel kommt nun die eigentliche Tabelle mit ggf. zusätzlichen Informationen zum Autor.

\begin{tabular}{l}
\textsf{\textbf{##1}}
\end{tabular}

Sollte ein weiterer Autor folgen, wird \texttt{\tabcolsep} zusätzlich eingefügt, um den Standardabstand bei Tabellen zu sichern, da die Tabelle vorher ohne rechten „Rand“ gesetzt wurde, um die letzte Tabelle ggf. genau bis zum rechten Rand setzen zu können.

Hier erfolgt die eigentliche Ausgabe.

Nach den Autoren wird der Titel über die komplette Textbreite ausgegeben. Danach wird der Inhalt der Aufgabenstellung gesetzt.

\begin{tabular}{@{}p{\tud@len@authortable}@{}}
\textsf{\textbf{##2}}
\end{tabular}


\newenvironment{task}[1][1][]{%
Zu Beginn wird als erstes die Überschrift und – die entsprechende Option vorausgesetzt – im PDF einen Lesezeichen- oder auch Outline-Eintrag gesetzt.

Da auch Gutachter und Betreuer durch den Befehl `\and` getrennt werden, wird dieser für die korrekte Ausgabe umdefiniert. Anschließend folgt die Ausgabe in einer Tabelle, die Spalte der Bezeichner entspricht der aus dem oberen Teil.

\taskform  Dies soll die Standardform einer Aufgabenstellung sein. Im ersten Argument werden kurz die Ziele motiviert und erläutert, im zweiten Argument werden im besten Fall die Schwerpunkte in einer \itemize-Umgebung aufgelistet.

19.2 Gutachten

evaluation (Umg.)
headline (Param.)
heading (Param.)
line (Param.)
grade (Param.)

Die Umgebung für das Gutachten nutzt ebenfalls die \tudpage-Umgebung. Sie wird auf einer neuen (rechten) Seite gesetzt. Es wird zu Beginn eine Tabelle mit Informationen zum Autor gesetzt. Zum Abschluss werden Ort, Datum und Gutachter ausgegeben.

Die gegebenen Note sowie Ort und Datum werden am Ende ggf. ausgegeben.

Der Befehl \texttt{\and} wird für einen möglichen Zweitgutachter angepasst. Das Hilfsmakro \texttt{\@tempa} dient zur Übernahme des richtigen Bezeichners für Erst- bzw. Zweitgutachter. Sollten mit \texttt{\referee} keine Gutachter angegeben sein, so werden die angegeben Betreuer verwendet.

\begin{tabular}{@{}l@{}}
\global\let\@referee\refereeothername
\end{tabular}

\texttt{\evaluationform} Dies soll die Standardform eines Gutachtens sein. Im ersten Argument wird kurz die Aufgabenstellung zusammengefasst, im zweiten Argument wird der Inhalt und die Struktur der
Arbeit kurz beschrieben. Im dritten Argument erfolgt die Bewertung, das letzte Argument beinhaltet die Note.

217 \newcommand \evaluationform [5] []
218 \begin {evaluation} [#1]
219 \removelastskip%
220 \ifblank {#2} {} \{ \minisec \{ \taskname \} #2 \%
221 \ifblank {#3} {} \{ \minisec \{ \contentname \} #3 \%
222 \ifblank {#4} {} \{ \minisec \{ \assessmentname \} #4 \%
223 \ifblank {#5} {} \{ \def \@grade { #5 } \%
224 \end {evaluation} 
225 

19.3 Aushang

notice (Umg.)
headline (Param.)
heading (Param.)
line (Param.)

Die Umgebung für Aushänge nutzt ebenfalls die \tudpage-Umgebung. Sie wird auf einer neuen (rechten) Seite gesetzt. Die Überschrift wird in der Voreinstellung auf den sprachabhängigen Bezeichner \noticename gesetzt, welcher allerdings mit dem Parameter \headline überschrieben werden kann.

226 \newenvironment \notice [1] []
227 \cleardoubleoddpage
228 \def \@headline { \noticename }
229 \TUD @parameter @family { \tudpage } {
230 \TUD @parameter @def { \headline } { \def \@headline { ##1 } }
231 \TUD @parameter @let { \heading } { \headline }
232 \TUD @parameter @let { \line } { \headline }
233 \TUD @parameter @handler @default { \headline }
234 }

Es wird zu Beginn das angegebene Datum oben auf der rechten Seite ausgegeben. Anschließend wird die Überschrift und der gegebene Titel gesetzt.

235 \tudpage [ pagemode = empty , cdhead = date , #1 ]
236 \tudbookmark \{ \noticename \} \{ notice \%
237 \ifx \@headline \@empty \else
238 \section * { \@headline }
239 \fi
240 }

Wenn keine Kontaktperson direkt angegeben wurden, werden die Informationen der angegeben Betreuer verwendet. Wenn eine Personenangabe gefunden wurde, werden die Kontaktdaten ausgegeben.

241 \ifx \@contactperson \@empty \let \@contactperson \@supervisor \fi
242 \ifx \@contactperson \@empty \else
243 \removelastskip%
244 \ifdim \parskip > \z@ \vskip \parskip \else \vskip \topsep \fi
245 \renewcommand \* \tud @ split @ contactperson @ do [2] { \%
246 \tud @ multiple @ fields @ store { @contactperson } { #1 } \%
247 \tud @ multiple @ fields @ preset { @contactperson } { #1 } \%
248 \begin { tabular } [ t ] { @{} l @{} } \%
249 \ignorespaces \#1 \tabularnewline
250 \ifx \@office \@empty \else \@office \tabularnewline
251 \ifx \@telephone \@empty \else \@telephone \tabularnewline
252 \ifx \@telefax \@empty \else \@telefax \tabularnewline
253 \ifx \@emailaddress \@empty \else \@emailaddress \tabularnewline

File n: tudscr-supervisor.dtx 255
Dies soll die Standardform eines Aushangs für eine Abschlussarbeit sein. Im ersten Argument wird kurz der Inhalt zusammengefasst, im zweiten Argument werden die Arbeitsschwerpunkte beschrieben.

```latex
\newcommand{\noticeform}[3]{% 
  \begin{notice}[#1] 
  \removelastskip
  \ifblank{#2}{}{\ifx@title@empty\else\minisec{\expandonce{\@title}}\fi}\#2\fi
  \#2\%
  \ifblank{#3}{}{\minisec{\focusname} \begin{itemize} \td@RaggedRight \#3 \end{itemize}}
  \end{notice}%
}\end{notice}%
```

File n: tudscr-supervisor.dtx
20 Das Paket twocolfix – Bugfix für den zweispaltigen Satz

Der \LaTeX\$\varepsilon$-Kernel enthält einen Fehler, der Kapitelüberschriften im zweispaltigen Layout höher setzt, als im einspaltigen. Zwar ist der Fehler\footnote{http://latex-project.org/cgi-bin/ltxbugs2html?pr=latex/3126} schon länger bekannt, allerdings noch nicht in den \LaTeX\$\varepsilon$-Kernel übernommen worden. Das Paket \texttt{twocolfix} soll das Problem beheben. Eine Integration des Bugfixes in KOMA-Script wurde bereits bei Markus Kohm angefragt, jedoch von ihm nicht weiter verfolgt\footnote{http://www.komascript.de/node/1681}.

Es wird der fehlerhafte Befehl aus dem \LaTeX\$\varepsilon$-Kernel neu definiert. Allerdings definieren die KOMA-Script-Klassen das Makro \texttt{@topnewpage} um und sichern vorher das Original in \texttt{\scr@topnewpage}. Daher wird der neue Befehl erst temporär definiert.

\begin{verbatim}
\long\def \@tempa [#1]{% 
  @nodocument% 
  \@next\@currbox@freelist{}{}% 
  \global \setbox\@currbox% 
  \vbox {% 
    \break% 
    \prevdepth\z@% 
    \begingroup% 
    \normalcolor% 
    \hsize\textwidth% 
    Damit der Inhalt des optionale Argumentes mit den gleichen Absatzeinstellungen gesetzt wird, werden diese nach \texttt{@parboxrestore} für KOMA-Script-Klassen wiederhergestellt.
    \@parboxrestore% 
    \ifcsname KOMAClassName\endcsname% 
      \KOMAoptionOf{[\KOMAoption{parskip}]\{\KOMAClassFileName}{parskip}% 
    \fi% 
    \col@number \@ne% 
    \ignorespaces #1\par% 
    \ifdim\parskip>\z@\null\fi% 
    \vskip -\dbltextfloatsep% 
    \endgroup% 
    \ifdim\parskip>\z@\vskip\parskip\else\null\fi% 
    \vskip -\topskip% 
  }% 
  \begingroup% 
  \splitmaxdepth\maxdepth \splittopskip\topskip% 
  \setbox@tempboxa \vsplit@currbox to\z@% 
  \endgroup% 
  \ifdim \ht@currbox<\textheight% 
    \ht@currbox \textheight% 
  \fi% 
  \global \count@currbox \tw@% 
  \@tempdim a -\ht@currbox\% 
  \advance \@tempdim a -\dbltextfloatsep% 
  \global \advance \@colht \@tempdim a% 
\end{verbatim}

\footnote{http://latex-project.org/cgi-bin/ltxbugs2html?pr=latex/3126}
Um eine Anpassung am \LaTeX\-Kernel feststellen zu können, wird auf die zum Zeitpunkt des Paketes aktuelle Definition von \texttt{\@topnewpage} geprüft.

File: tudscr-twocolfix.dtx 258
Nachdem der neue Befehl sowie die Definition aus dem \LaTeX2ε-Kernel zuerst temporär definiert wurden, wird nun abhängig von der aktiven Klasse der notwendige Befehl mit der neuen Definition überschrieben.

\@topnewpage \scr@topnewpage \ifx\scr@topnewpage\@undefined
\@check@eq\@topnewpage\@tempb%
\let\@topnewpage\@tempa%
\else
\@check@eq\scr@topnewpage\@tempb%
\let\scr@topnewpage\@tempa%
\fi
21 Das Paket mathswap – Aktive Trennzeichen im Mathematikmodus


Ähnliche Funktionalitäten werden bereits durch die Pakete icomma und ziffer bereitgestellt. Bei icomma muss jedoch beim Verfassen des Dokumentes durch den Autor beachtet werden, ob das verwendete Komma einem Dezimaltrennzeichen entspricht ($t = 1,2$) oder aber einem normalen Komma im Mathematiktext ($z = f(x, y)$), wo ein gewisser Abstand nach dem Komma durchaus gewünscht ist. Das Paket ziffer liefert dafür die gewünschte Funktionalität,\(^\text{18}\) ist allerdings etwas unflexibel, was den Umgang mit den Trennzeichen anbelangt. Als Alternative zu diesem Paket kann außerdem ionumbers verwendet werden.

Das Paket mathswap sorgt dafür, dass Trennzeichen direkt vor einer Ziffer erkannt und nach bestimmten Vorgaben ersetzt werden. Sollte sich jedoch zwischen Trennzeichen und Ziffer Leerraum befinden, wird dieser als solcher auch gesetzt.

\begin{verbatim}
\commaswap
\dotswap
@@commaswap
@@dotswap
\end{verbatim}

Die beiden Befehle \commaswap und \dotswap sind die zentrale Benutzerschnittstelle des Paketes. Das Makro \commaswap definiert das Trennzeichen oder den Inhalt, wodurch ein Komma ersetzt werden soll, auf welches direkt danach eine Ziffer folgt. Normalerweise setzt \LaTeX nach einem Komma im mathematischen Satz zusätzlich einen horizontalen Abstand. Bei der Ersetzung durch \commaswap entfällt dieser. Die Voreinstellung für \commaswap ist deshalb auf ein Komma (,) gesetzt.

Mit dem Makro \dotswap kann definiert werden, wodurch der Punkt im mathematischen Satz ersetzt werden soll, wenn auf diesen direkt anschließend eine Ziffer folgt. Da der Punkt im deutschsprachigen Raum zur Gruppierung von Ziffern genutzt wird, ist hierfür standardmäßig ein halbes geschütztes Leerzeichen definiert (\,).

Die mit \commaswap{\langle Trennzeichen\rangle} und \dotswap{\langle Trennzeichen\rangle} definierten Trennzeichen werden in @@commaswap und @@dotswap gesichert.

\begin{verbatim}
1 \newcommand{\@commaswap}{,}
2 \newcommand{\commaswap[1]}{\gdef{\commaswap}{\#1}{\ignorespaces}}
3 \newcommand{\@dotswap}{,}
4 \newcommand{\dotswap[1]}{\gdef{\dotswap}{\#1}{\ignorespaces}}
\end{verbatim}

\begin{verbatim}
\ms@comma
\ms@commachar
\ms@dot
\ms@dotchar
\end{verbatim}

In den Befehlen \ms@commachar und \ms@dotchar wird die originale Definition von Komma und Punkt im Mathematikmodus gesichert, welche weiterhin verwendet wird, falls nach diesen nicht direkt eine Ziffer folgt. Mit den Makros \ms@comma und \ms@dot werden die Definitionen von Komma und Punkt als normale Textzeichen für den mathematischen Satz

\(^{\text{18}}\) kein Leerraum nach Komma, wenn direkt danach eine Ziffer folgt
bereitgehalten. Anschließend werden beide als aktive Zeichen für den Mathematikmodus gesetzt.

5 \newcommand*{\ms@commachar}{}
6 \newcommand*{\ms@dotchar}{}
7 \AtBeginDocument{%
8 \DeclareMathSymbol{\ms@comma}{\mathord}{letters}{"3B}%
9 \mathchardef\ms@commachar\mathcode\expandafter`\%
10 \mathcode\expandafter`\%="8000%
11 \DeclareMathSymbol{\ms@dot}{\mathord}{letters}{"3A}%
12 \mathchardef\ms@dotchar\mathcode\expandafter`.`%
13 \mathcode\expandafter`\%="8000%
14 \mathswapon%
15 }

\mathswapon\mathswapoff Mit den Befehlen \mathswapon und \mathswapoff kann die Funktionalität des Paketes innerhalb des Dokumentes aktiviert und deaktiviert werden.

\ms@commaswap\ms@dotswap Der Befehl \ms@ifnum reicht das erste Argument an \ms@@ifnum weiter, falls der Inhalt von \ms@token zu einer Ziffer expandiert. Andernfalls wird das zweite Argument ausgeführt.

26 \newcommand*{\ms@ifnum}[2]{%
27 \ifx\ms@token0\ms@@ifnum[1]\else%
28 \ifx\ms@token1\ms@@ifnum[1]\else%
29 \ifx\ms@token2\ms@@ifnum[1]\else%
30 \ifx\ms@token3\ms@@ifnum[1]\else%
31 \ifx\ms@token4\ms@@ifnum[1]\else%
32 \ifx\ms@token5\ms@@ifnum[1]\else%
33 \ifx\ms@token6\ms@@ifnum[1]\else%
34 \ifx\ms@token7\ms@@ifnum[1]\else%
35 \ifx\ms@token8\ms@@ifnum[1]\else%
36 \ifx\ms@token9\ms@@ifnum[1]\else%
37 #2%
38 \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
39 }

Wurde eine nachfolgende Ziffer erkannt, so wird die Definition von Punkt und Komma auf die eines normalen Textzeichens gesetzt und anschließend die mit \commaswap bzw. \dotswap angegebene Ersetzung ausgegeben.

40 \newcommand*{\ms@ifnum[1]}{%
41 \begingroup%
42 \mathcode\expandafter`\%="\the\ms@comma%
43 \mathcode\expandafter`\%="\the\ms@dot%
44 #1%
45 \endgroup%
46 }
Mit `\futurelet` wird dem Makro `\ms@token` das nachfolgende Zeichen von Komma bzw. Punkt zugewiesen. Anschließend wird mittels `\ms@smartcomma` bzw. `\ms@smartdot` geprüft, ob es sich beim Inhalt von `\ms@token` um eine Ziffer handelt. Dazu wird der Befehl `\ms@ifnum` verwendet. Sollte dies der Fall sein, so wird ein Komma innerhalb einer Zahl durch `@commaswap` und ein Punkt entsprechend mit `@dotswap` ersetzt. Sollte direkt nach dem Trennzeichen keine Ziffer folgen, wird für Punkt und Komma die originale Definition aus `\ms@commachar` bzw. `\ms@dotchar` verwendet.

\begin{verbatim}
47 \newcommand*{\ms@token}{}
48 \begingroup
49 \catcode',=\active
50 \gdef,{\futurelet{\ms@token}{\ms@smartcomma}}
51 \catcode'.=\active
52 \gdef.{\futurelet{\ms@token}{\ms@smartdot}}
53 \endgroup
54 \newcommand*{\ms@smartcomma}{\ms@ifnum{\ms@commaswap}{\ms@commachar}}
55 \newcommand*{\ms@smartdot}{\ms@ifnum{\ms@dotswap}{\ms@dotchar}}
\end{verbatim}
Anhang

Index

Kursive Zahlen entsprechen der Seite, auf welcher der korrespondierende Eintrag beschrieben wird. Unterstrichene Zahlen verweisen auf die Codezeile der Definition.

A

\abovecaptionskip .................................. c-497
abstract (Opt.) ......................................... c-264
abstract (Umg.) ......................................... j-340
abstract (Param.) ....................................... j-340
language (Param.) ....................................... j-340
markboth (Param.) ........................................ j-340
option (Param.) .......................................... j-340
\abstractname (Lok.) ..................................... e-15
\advisor .................................................... d-173
\advisorname (Lok.) ....................................... e-15
\advisornickname (Lok.) .................................. e-15
\assessmentname (Lok.) .................................... e-60
\author .................................................... d-53
\authorname ............................................... d-53
\authornickname ......................................... d-53
\authornickname (Field) .................................. d-53
\authorname (Lok.) ........................................ e-57

B

\bachelorthesisname (Loc.) ............................... e-40
backgroundcolor (Opt.) .................................. h-295
barfont (Opt.) ............................................ h-312
\BeforeSelectAnyPageStyle .............................. c-198
\belowcaptionskip .................................... c-497
\bfdefult ................................................ c-356
\bfseriesrm ............................................... c-356
\bfseriessf ............................................... c-356
\bfseriesstt .............................................. c-356
\bigskipamount .......................................... c-497
\birthday ............................................... k-397
\birthplace ............................................. k-394
\blautabelle ............................................ k-753
bleedmargin (Opt.) ....................................... f-266
\blocking ................................................. j-613
\blockingname (Opt.) ..................................... e-15
\blockingtext (Loc.) ..................................... e-15
bluebg (Opt.) ............................................ k-537
\bool@lock ............................................... b-231
\bool@set@lock .......................................... b-231
\bool@std@lock .......................................... b-231

cd (Opt.) ................................................ h-14
\cdblue (Farbe) ......................................... m-35

cddchapter (Opt.) ...................................... h-180
cdclock ................................................... h-115
cdddarkblue (Farbe) ..................................... m-35
cdddarkgreen (Farbe) ................................... m-35
\cdfont .................................................. c-651
\cdfont (Opt.) .......................................... c-202
\cdfonts (Opt.) ......................................... k-231
\cdfoot (Opt.) .......................................... g-755
cdgeometry (Opt.) ....................................... f-206
cdgray (Farbe) ........................................... m-35
cdgreen (Farbe) ......................................... m-35
cdgrey (Farbe) ........................................... m-35
cdhead (Opt.) .......................................... g-571
cdindigo (Farbe) ........................................ m-35
cdmath (Opt.) .......................................... c-721
cdorange (Farbe) ........................................ m-35
cdpart (Opt.) ............................................ h-159
cdpurple (Farbe) ........................................ m-35
cdsection (Opt.) ....................................... h-210
cdtitle (Opt.) .......................................... h-138
cdintersection .......................................... k-626
cdsubsection ............................................ k-626
\chair ..................................................... d-1
\chair (Field) ........................................... d-1
\chairname ............................................ d-284
\chairname (Field) ...................................... d-284
\chairman ................................................ d-284
\chairmanname (Lok.) .................................. e-60
\chapterheadingvskip (Länge) ......................... k-386
chapterpage ............................................. k-491
chapterpage (Opt.) ..................................... h-230
clearcolor (Opt.) ...................................... m-35
clearcolour (Opt.) ..................................... m-35
\cleardoubleevenpageusingstyle ....................... f-1188
\cleardoubleoddpageusingstyle ....................... f-1188
\cleardoublepageusingstyle ............................ f-1188
\cleardoublespecialpage (Opt.) ....................... h-234
\color (Opt.) .......................................... k-350
\colorname ............................................. k-427
\colour (Opt.) ......................................... k-350
\columnsep ............................................ c-497
\commaswap ............................................. p-1
\commaswap ............................................. p-1
\company ................................................ d-169
\company (Field) ....................................... d-169
\confirmation ........................................... f-586
\confirmationclosing ................................... f-185
\confirmationclosing (Field) ......................... f-185
\confirmationname (Lok.) ............................... e-15

File Key: a=tudscr-version.dtx, b=tudscr-base.dtx, c=tudscr-fonts.dtx, d=tudscr-fields.dtx,
e=tudscr-locale.dtx, f=tudscr-area.dtx, g=tudscr-pagestyle.dtx, h=tudscr-layout.dtx,
i=tudscr-title.dtx, j=tudascr-frontmatter.dtx, k=tudscr-compact.dtx, l=tudascr-misc.dtx,
m=tudascr-color.dtx, n=tudscr-supervisor.dtx, o=tudscr-twocolumnfix.dtx, p=tudscr-mathswap.dtx
\def\tudvartimes#1#2#3#4#5#6#7#8#9{}\def\tudvartimes#1#2#3#4#5#6#7#8#9{}
\def\tudvartimes#1#2#3#4#5#6#7#8#9{}\def\tudvartimes#1#2#3#4#5#6#7#8#9{}
Änderungsliste

v1.0
Allgemein
TUD-Script-Bundle erstmalig veröffentlicht ...................... 6

v2.00
Allgemein
TUD-Script-Bundle auf docstrip umgestellt und stark erweitert ...... 6

v2.01
Allgemein
Anpassungen in Dokumentation und Schrifteinstellung, Fehlerkorrekturen .. 6
Versionsanforderungen bei benötigten Paketen .................. 24
Implementierung
titlepage (Schriftel.:) Explizite Wahl der genutzten Schriftstärke ..... i-26

v2.01a
Implementierung
\maketitle: Fehler bei Nutzung von \subtitle behoben ............ i-40

v2.01b
Implementierung\tud@authortable@set: Probleme mit Paket calc behoben ... n-4

v2.02
Allgemein
Problem globaler Längenänderungen behoben ...................... 6
Umbenennung mehrerer Befehle zur Kompatibilität mit anderen Paketen .. 6
Paket titlepage nicht weiter unterstützt 9, 26, 51, 64, 74, 104, 170, 235
FamilyKeyState wird von Optionen genutzt ........ 26, 74, 104, 142, 191, 235
Warnung bei Verwendung von graphics 24
Unterschutzung der standalone-Klasse 74
mathswap: Schalter zum Aktivieren und Deaktivieren der Funktionalität ... 260
Unterschutzung für fontspec 26
Verwendung von mweights unnötig ... 26
tudscrfonts: Nutzung alter TUD-Klassen mit neuen Schriftfamilien ermöglicht 208
Implementierung
abstract (Umg.): markboth (Param.): neu ................... j-340
pagestyle (Param.): neu ................................ j-340
barfont (Opt.): neu ........................................ k-312
\blocking:
neu, \restrictiontext umbenannt ....................... j-613
an scrlayer-scrpage angepasst ......................... j-613
\blockingname (Lok.): neu, \restrictionname umbenannt ... e-15
\blockingtext (Lok.):
neu, \restrictioncontexttext umbenannt e-15
Korrektur bei der Verwendung von \@title .................. e-15
cdcover (Opt.): neue Option für \makecover ................. h-115
cdfont (Opt.): Schlüsselwerte für cdhead ................ e-202
Schlüsselwerte für fontspec ...................... e-202
cleardoublespecialpage (Opt.): Werte für Seitenstil aus den TUD-Script- bzw. KOMA-Script-Klassen möglich ...... h-234
\confirmation: Neuimplementierung für scrlayer-scrpage ........ j-586
\confirmationtext (Lok.): Korrektur bei der Verwendung von \@title ................ e-15
\contactperson: neu, Makro \contact umbenannt .............. d-297
\@contact: neu, Makro \contact umbenannt .............. d-297
ddc (Opt.): automatische Logowahl ................ g-1170
ddfont (Opt.): Logo von DRESDEN-concept entweder in Kopf oder Fuß ............ g-1170
\declaration: Neuimplementierung für scrlayer-scrpage ........ j-552
declarations (Umg.):
neu ...................................................... j-493
columns (Param.): neu ................................ j-493
markboth (Param.): neu ................................ j-493
pagestyle (Param.): neu ................................ j-493
\discipline: neu, von \branch umbenannt ................. d-90
\discipline (Field): neu, von \branch umbenannt ............. d-90
\discipline (Lok.): neu, Umbenennung von \branchname ...... e-40
\email: neu, Umbenennung des Makros \email ............... d-297
\@email (Field): neu, \email...................................... d-297
empty.tudheadings (Seitenstil): neu ................ g-7
fontspec (Opt.): neu .................................. k-187
footnotes (Opt.): neu ................................... k-216
geometry (Opt.): neu ..................................... k-311
\graduation: neu, von \degree umbenannt ................. d-146
\@graduation (Field): neu, von \degree umbenannt ......... d-146
\@graduation (Field): neu, \degreeumbabrennnt ......... d-146

File Key:
a=tudscr-version.dtx, b=tudscr-base.dtx, c=tudscr-locale.dtx, d=tudscr-fields.dtx,
e=tudscr-local.dtx, f=tudscr-area.dtx, g=tudscr-pagestyle.dtx, h=tudscr-layout.dtx,
i=tudscr-title.dtx, j=tudscr-frontmatter.dtx, k=tudscr-abstract.dtx, l=tudscr-misc.dtx,
m=tudscr-color.dtx, n=tudscr-supervisor.dtx, o=tudscr-twocolfix.dtx, p=tudscr-mathswap.dtx
\graduationtext (Lok.): neu, umbenannt von \degreetext \hspace{e-15}
\headingsvskip (Länge): neu \hspace{k-380}
\headlogo: Anpassung an automatische Wahl des DRESDEN-concept-Logos \hspace{g-670}
if@tud@abstract@markboth: neu \hspace{j-263}
if@tud@ddc@auto: neu \hspace{g-1180}
if@tud@ddc@internal: neu \hspace{g-1147}
if@tud@declaration@markboth: neu \hspace{j-417}
if@tud@dinbold: neu \hspace{c-47}
if@tud@layout@cover@num@locked: neu \hspace{h-14}
if@tud@symbolsheads: neu \hspace{l-216}
if@tud@fontsayscale: enabled: neu \hspace{c-27}
if@tud@standalone@crop: neu \hspace{j-15}
ifdef: Anpassungen für \fontspec \hspace{c-966}
\makecover:
Warnung für die Option titlepage mit \hspace{i-587}
dem Wert firstiscover \hspace{j-40}
ohne Corporate Design verwendbar \hspace{i-587}
Satzspiegel mit \coverpage \margin \hspace{i-587}
änderbar, falls CD-Layout deaktiviert \hspace{i-587}
cdgeometry \hspace{i-587}
\maketitle:
firstiscover wird als Wert für die \hspace{i-40}
Option titlepage unterstützt \hspace{i-40}
Im zweispaltigen Satz wird für den \hspace{i-40}
Titelkopf \tud@title@twocolumn \hspace{i-40}
verwendet \hspace{i-40}
Titelseite mit \tud@make@titlepage \hspace{i-40}
pagernumber \hspace{i-40}
\maketitleonecolumn: einspaltiger Text \hspace{i-526}
obligatorisch \hspace{i-526}
\mathswapoff: neu \hspace{p-16}
\mathswapon: neu \hspace{p-16}
\ms@commaswap: neu \hspace{p-16}
\ms@dotswap: neu \hspace{p-16}
\mv@dinbold: Symbole symbols werden \hspace{c-773}
aus Schriftart DIN BOLD verwendet \hspace{j-340}
\nextabstract: neu \hspace{j-340}
\nicename (Lok.): neu, umbenannt von \hspace{e-60}
\contactname \hspace{g-406}
\pageheadingsvskip (Länge): neu \hspace{k-380}
\parttitle (Schriftel.): neu \hspace{h-406}
\plain.tudheadings (Seitenstil): neu \hspace{g-7}
\professorothername (Lok.): neu \hspace{e-15}
\protected@expandyargs: neu \hspace{b-114}
\refereename (Lok.): Unterscheidung, ob \hspace{b-151}
ein oder mehrere Gutachter \hspace{g-42}
angegeben sind \hspace{e-15}
\supervisor: erzeugter Eintrag der \hspace{d-160}
Betreuer mit \supervisor für \hspace{d-160}
Selbstständigkeitserklärung \hspace{d-160}
verworfen \hspace{d-160}
\supporter: erzeugter Eintrag der \hspace{d-160}
Betreuer mit \supervisor für Selbstan-
\thispagestyle{Auswahl eines Aliasseitenstils unterdrückt} \g{303}
\TUD@AfterPackage@doc: neu \b{466}
\TUD@bool@numkey: neu \b{167}
\tud@cggeometry@process:
  intiale Festlegung der Länge \marginpar \f{734}
  Satzspiegel des Corporate Designs angepasst \f{734}
  Satzspiegel kompatibilitätsabhängig \f{734}
\tud@cggeometry@set:
  Längenberechnung für den Satzspiegel an Referenzlänge \f{34}
\tud@len@widemargin gekoppelt \f{34}
\tud@cmd@check: neu \b{257}
\tud@CurrentPagestyle@reset:
  Bugfix \g{1274}
\tud@ddc@foot@num: Seitenstilauswahl bei Deaktivierung verbessert \g{1216}
\tud@ddc@head@num: Seitenstilauswahl bei Deaktivierung verbessert \g{1202}
\TUD@deprecated@cs@ca: neu \k{1}
\TUD@deprecated@key: neu \k{1}
\tud@foo@logocolor: neu \g{339}
\tud@foo@logocmd@set:
  neu \g{930}
  optionale Parameter für unterschiedliche Dateien individuell nutzbar \g{930}
\tud@foo@logofilenames: \g{920}
\tud@foo@logofileoptions: \g{920}
\tud@head@bar@num: \g{571}
\tud@head@font@num: \g{571}
\tud@head@font@set: Bugfix für die Änderung von \textbackslash{spaceskip} durch das Paket \texttt{ragged2e} \g{412}
\tud@head@text@box: Zweite Zeile wird auch leer gesetzt \g{506}
\tud@headmid@font@normal: \g{392}
\tud@headmid@font@reverse: \g{392}
\tud@if@v@lower: \g{161}
\TUD@lengthkey: \g{167}
\TUD@parameter@ps@def:
  \texttt{cf@oot} (Param.): \g{1255}
  \texttt{cd@head} (Param.): \texttt{barfont} und \texttt{widehead}
    überlagert \g{1254}
  \texttt{foot@logo} (Param.): \g{1256}
  \texttt{head@logo} (Param.): \g{1256}
\tud@parthead@start@skip:
  Bei Verwendung der Option \texttt{part@title:}
  Abstand ist identisch zu Untertitel auf Titelseite \g{659}
\tud@part:
  Korrektur bei der Position der Überschrift \g{783}
\tud@ps@on@background: \g{339}
\tud@ps@on@select: \g{333}
\tud@title@twocolumn: Bugfix für falsche Position des Titels \j{526}
\tud@v@2.0: neu \k{121}
\tud@v@last: angepasst \k{121}
\tud@x@textcase@uc@nomath:
  \texttt{textcase}:
    Ignorieren des Befehls \texttt{\@both} im Argument des Befehls
  \texttt{MakeTextUppercase} \i{234}
\tudheadings@foot@back (Layer): \g{110}
\tudheadings@foot@logo (Layer):
  überarbeitet \g{110}
\tudheadings@head@back (Layer): \g{19}
\tudheadings@head@bar (Layer): \g{19}
\tudheadings@pagecolor (Layer): \g{279}
\tudpage (Umg.):
  \texttt{pagestyle} (Param.):
\tud@if@tudheadings verwendet \g{1274}
\v{2.03a}
Implementierung
\tud@v@2.0: neu \k{121}
\v{2.04}
Allgemein
  Interaktion mit Paketen insbesondere für den Mathematiksatz verbessert \ldots \texttt{26}
\tudscrfonts:
  Unterstützung veralteter Posterklassen \ldots \texttt{208}
Implementierung
\abovecaptionskip:
  Wert abhängig von der Schriftgröße \c{497}
\abstract
  \texttt{abstract} (Umg.):
  Reimplementierung \j{340}
\bfseries@lt:
  \texttt{bfseries@lt}:
    Unterstützung für das Paket \texttt{mweights} \c{356}
\bfseries@rm:
  \texttt{bfseries@rm}:
    Unterstützung für das Paket \texttt{mweights} \c{356}
\bfseries@sf:
  \texttt{bfseries@sf}:
    Unterstützung für das Paket \texttt{mweights} \c{356}
\bigskipamount:
  \texttt{bigskipamount}:
    Wert abhängig von der Schriftgröße \c{497}
\cd
  \texttt{cd} (Opt.):
    \texttt{Wert barcolor neu} \h{14}
\cdchapter
  \texttt{cdchapter} (Opt.):
    \texttt{Wert barcolor neu} \h{180}
\cdcover
  \texttt{cdcover} (Opt.):
    \texttt{Wert barcolor neu} \h{115}
\cdfont
  \texttt{cdfont} (Opt.):
   \texttt{bedingtes Setzen der Option automark} \c{755}
\cdfoot
  \texttt{cdfoot} (Opt.):
    \texttt{Wert barcolor neu} \c{755}
\cdpart
  \texttt{cdpart} (Opt.):
    \texttt{Wert barcolor neu} \c{571}
\cddate
  \texttt{cddate} (Opt.):
    \texttt{Wert barcolor neu} \c{159}
\cdday
  \texttt{cdday} (Opt.):
    \texttt{Wert barcolor neu} \c{138}
\columnsep:
  \texttt{columnsep}:
    Wert abhängig von der Schriftgröße \c{497}
\declarations
  \texttt{declarations} (Umg.):
    Reimplementierung \j{493}
\fontsize
  \texttt{fontsize} (Opt.):
    \texttt{Option aus KOMA-Script erweitert} \c{334}
\footnotesize
  \texttt{footnotesize} \g{984}
\if@tud@foo@logocmd@set: \g{930}
\if@tud@foo@logo@option@set: \g{1180}
\if@tud@head@logo@option@set: \g{1180}
\if@tud@main@logo@wrn: \g{630}

File Key: a=tudscr-version.dtx, b=tudscr-base.dtx, c=tudscr-fonts.dtx, d=tudscr-fields.dtx, 
e=tudscr-locale.dtx, f=tudscr-area.dtx, g=tudscr-pagestyle.dtx, h=tudscr-layout.dtx, 
i=tudscr-title.dtx, j=tudscr-frontmatter.dtx, k=tudscr-comp.dtx, l=tudscr-misc.dtx, 
m=tudscr-color.dtx, n=tudscr-supervisor.dtx, o=tudscr-twocolumnfix.dtx, p=tudscr-mathswap.dtx
\if@tud@fonts\@enabled:
  Erkennung des Paketes \texttt{mweights} \texttt{c-356}
\fi
\if@tudmath\@poster
  wird unterstützt \texttt{c-356}
\fi
\maketitle: \texttt{tudmath}\@poster wird
  unterstützt \texttt{c-356}
\m@default: Unterstützung für das Paket \texttt{mweights} \texttt{c-356}
\texttt{mseries@rm}: Unterstützung für das Paket \texttt{mweights} \texttt{c-356}
\texttt{mseries@sf}: Unterstützung für das Paket \texttt{mweights} \texttt{c-356}
\texttt{mseries@tt}: Unterstützung für das Paket \texttt{mweights} \texttt{c-356}
\md@skipamount: Wert abhängig von der Schriftgröße \texttt{c-497}
\filename: \texttt{neu}, Umbenennung von \texttt{authorname} \texttt{e-60}
\nextdeclaration: \texttt{neu} \texttt{j-552}
\n@DIN (Opt.): \texttt{neu} \texttt{k-261}
\p@gestyle: Aktivierung der Seitenstile \texttt{tudheadings} vor \texttt{scrayer-scrpage}
möglich \texttt{g-306}
\preprocesstitle: \texttt{tudmath}\@poster wird
  unterstützt \texttt{c-356}
\ps@tudposter: \texttt{tudmath}\@poster wird
  unterstützt \texttt{c-356}
\rm@default: Unterstützung für das Paket \texttt{mweights} \texttt{c-356}
\texttt{scr}@@startpart:
  KOMA-Script v3.18 \texttt{h-783}
\texttt{scr}@@startpart:
  KOMA-Script v3.18 \texttt{h-783}
\sectionfont: \texttt{tudmath}\@poster wird
  unterstützt \texttt{c-356}
\texttt{sf}@default: Unterstützung für das Paket \texttt{mweights} \texttt{c-356}
\texttt{slantedGreek} (Opt.): \texttt{neu} \texttt{c-745}
\texttt{smallskipamount}: Wert abhängig von der Schriftgröße \texttt{c-497}
\subsectionfont: \texttt{tudmath}\@poster wird
  unterstützt \texttt{c-356}
\texttt{sub@titlefont}: \texttt{tudmath}\@poster wird
  unterstützt \texttt{c-356}
\texttt{title} (Schriftel.): \texttt{tudmath}\@poster wird
  unterstützt \texttt{c-356}
\@default: Unterstützung für das Paket \texttt{mweights} \texttt{c-356}
\texttt{tudabstract@toc@num} \texttt{c-251}
\texttt{tudabstract@toc@num@locked} \texttt{j-251}
\texttt{tud@AfterChangingArea} \texttt{f-1086}
\texttt{TUD@AfterPackage@set} \texttt{b-466}
\texttt{tud@BeforeSelectAnyPageStyle}:
  Auswahl der Schriften für Fußbereich
ergänzt \texttt{f-1041}
\texttt{tud@cf@font@set} \texttt{c-651}
\texttt{tud@cf@font@trim} \texttt{c-651}
\texttt{tud@cf@font@trim} \texttt{c-651
v2.05a
Implementierung
\tud@v2.05a: neu  

v2.05b
Implementierung
\tud@v2.05b: neu  

v2.05c
Implementierung
\tud@v2.05c: neu  

v2.05d
Implementierung
\tud@font@greek@check@slanted: neu  
\tud@v2.05d: neu  

v2.05e
Implementierung
\tud@v2.05e: neu  

v2.05f
Implementierung
\tud@v2.05f: neu  

v2.05g
Implementierung
\tud@changes@name (Lok.): neu  
\tud@counter@name (Lok.): neu  
\tud@counters@name (Lok.): neu  
\tud@environment@name (Lok.): neu  
\tud@environments@name (Lok.): neu  
\tud@field@name (Lok.): neu  
\tud@fields@name (Lok.): neu  
\tud@general@name (Lok.): neu  
\tud@implementation@name (Lok.): neu  
\tud@index@text (Lok.): neu  
\tud@KOMAfont@name (Lok.): neu  
\tud@KOMAfonts@name (Lok.): neu  
\tud@layer@name (Lok.): neu  
\tud@layer@pagestyle@name (Lok.): neu  
\tud@layer@pagestyles@name (Lok.): neu  
\tud@option@name (Lok.): neu  
\tud@options@name (Lok.): neu  
\tud@parameter@name (Lok.): neu  
\tud@parameters@name (Lok.): neu  
\tud@todo@name (Lok.): neu  
\tud@TUDcolor@name (Lok.): neu  
\tud@TUDcolors@name (Lok.): neu  
\tud@v2.05g: neu  

v2.05h
Implementierung
\tud@v2.05h: neu  

v2.05i
Implementierung
\tud@v2.05i: neu  

v2.05j
Implementierung
\tud@sectioning@set: Sonderbehandlung von \add...
\tud@v2.05j: neu  

v2.05k
Implementierung
\tud@x@fontspec@fam@din: neu  
\tud@x@fontspec@fam@univ: neu  
\tud@x@fontspec@noerror: unnötig seit fontspec '2017/01/24'  
\tud@x@fontspec@set@din: neu  
\tud@x@fontspec@set@univ: neu  

v2.05m
Implementierung
\if@tud@parameter@ps@font@set: neu  
\tud@v2.05m: neu  
\tud@x@siunitx@enabledtrue: neu  