

The tablestyles package

Matthias Pospiech
matthias.pospiech@gmx.de

0.1 from 2014/06/27

Abstract

LaTeX unfortunately does not include a good separation of text and style in tables. This package tries to solve this problem by defining reusable table commands and an interface to define a style for table. Furthermore the package defines common used column styles and a bugfix command for lists in tables.

Contents

1	Introduction	2
2	Usage	2
2.1	Loading	2
2.2	Tutorial	2
2.3	Setup of styles	6
2.4	Column type definitions	11
2.5	Items in tables	14
3	Implementation	15
3.1	Setup and packages	15
3.2	Configuration commands	16
3.3	Style applying commands	18
3.4	Style definitions	21
3.5	Support for itemized lists in tables	23
3.6	New columntype definitions	24

1 Introduction

This package provides commands to be used inside a tabular environment, which allow the user to define the style of a table once but apply it to every table using the commands defined by this package. The styles are divided into the head, subhead and the body of the table. Further this package introduces new column types and enhancing the typesetting of lists inside tables.

2 Usage

The usage is first described using a brief review of the possibilities to change the layout of a table using the font size, font color, line appearance and the background colors of special and alternating cells. The latter is based on the xcolor/colortab package. Subsequent the commands provided by the package are explained in detail followed by a description of the creation of new styles.

2.1 Loading

The package is loaded with

```
\usepackage{tablestyles}
```

It should be loaded after the xcolor package and before the hyperref package.

2.2 Tutorial

The code listing 1 shows a tabular with all commands necessary to set up the style of the table displayed below (table 1). In this code the commands of this package are *not* used. The equivalent code with the commands of this package is provided in the following code listing 2.

```
\begin{table}  
%  
\small % fontsize  
\sffamily % font family = sans serif  
\centering % alignment of tabular  
\renewcommand{\arraystretch}{1.4} % larger cell height  
\rowcolors{1}{colorrow}{colorbody}%  
%  
\begin{tabular}{*{2}{p{0.45\textwidth}}}  
\hline  
\rowcolor{colorhead}  
 \bfseries header &  
 \bfseries header \\\end{tabular}
```

```

\hline
%
content & content \\
content & content \\
content & content \\
content & content \\
content & content \\
%
\rowcolor{colorsubhead}
subhead & subhead \\
%
content & content \\
content & content \\
\hline
\end{tabular}
\caption{table with bold header font}
\end{table}

```

Listing 1: example code for a table with manual changes to the appearance.

header	header
content	content
content	content
content	content
content	content
content	content
subhead	subhead
content	content
content	content

Table 1: table with bold header font

The aim of this package is to generalize the generation of such tabular styles. The code listing 2 demonstrates the usage, with the result shown in table 2, which is identical with the previous one (table 1).

```

\begin{table}
%
\tablestyle[sansbold]
%
\begin{tabular}{*{2}{p{0.45\textwidth}}}}
\theadstart
\thead header &
\thead header \\
\tbody

```

```

%
content & content \\
content & content \\
content & content \\
content & content \\
content & content \\
%
\subheadstart
\subhead subhead &
\subhead subhead \\
%
content & content \\
content & content \\
\tend
\end{tabular}
\caption{table with bold header font using the styles by this package}
\end{table}

```

Listing 2: example code for a table with style commands.

header	header
content	content
content	content
content	content
content	content
content	content
subhead	subhead
content	content
content	content

Table 2: table with bold header font using the styles by this package

The following sections show the possibilities of applying a style in general and how to style a heading or subheading

applying styles

`\tablestyle` The command `\tablestyle[<style>]` is used to choose the style of the table. It redefines all commands that apply changes to font, colors, lines, etc. It is required before the tabular environment to apply the changes to the complete table. If not inserted the last chosen layout, or if never applied the standard layout is used.

`\tbegin` To apply lines or any other elements at the top and bottom of a table the commands `\tbegin` at the beginning `\tend` at the end of the table are used.

`\tend`

```

\begin{table}
%
\tablestyle[stylename]
%
\begin{tabular}{*{2}{p{0.45\textwidth}}}}
\tbegin
%
content & content \\
content & content \\
\tend
\end{tabular}
\end{table}

```

Listing 3: code example with table style applied.

headings and subheadings

If a table starts with a header `\thead` `\tbody` is replaced by `\theadstart` and the following main table content starts with `\tbody` (as shown in the previous code listing 3). In general the commands `\theadstart` and `\tbodystart` set up the color of the heading row and print out a line by default. `\theadend` and `\tbodyend` can be used to print out a line afterwards (or apply any other useful properties at the end), but are empty by default. The properties of each cell in the heading are applied with `\thead` and `\tbody`. Unfortunately the cell property command must be applied to each cell. The code listing 4 demonstrates the implementation of heading and subheading.

```

\theadstart
  \thead header &
  \thead header \\
\theadend % not necessary in default styles
\tbody
%
content & content \\
...
content & content \\
%
\tbodystart
\tbody subhead &
\tbody subhead \\
\tbodyend % not necessary in default styles
%

```

Listing 4: code for heading and subheading.

If a heading or subheading shall span all columns of a table this can be achieved with the simplified commands `\theadrow{<number of cells>}{<text>}` and the

`\tsubheadrow` equivalent `\tsubheadrow{⟨number of cells⟩}{⟨text⟩}`. Note that these automatically insert a `\tabularnewline`. The first parameter therefore must match the number of columns in the table.

```
\theadstart
\theadrow{2}{full width header}
\tbody
%
content & content \\
%
\tsubheadrow{2}{full width subheader} %
%
content & content \\
```

Listing 5: example for multi column rows.

2.3 Setup of styles

Style are applied to the table with the following commands

- `\tbegin`
- `\tbody`
- `\tend`
- `\thead`
- `\theadstart`
- `\theadend`
- `\tsubheadstart`
- `\tsubheadend`
- `\tsubhead`
- `\theadrow`
- `\tsubheadrow`

which were described in the previous section. To define the style the following commands are used which define the font, the color, and other settings.

fontsize

`\setuptablefontsize` Font sizes can be set up using `\setuptablefontsize{⟨name⟩}{⟨size definition⟩}` and be retrieved with `\tablefontsize{⟨name⟩}`. Predefined are sizes for *body* and *head*

```
% definition
\setuptablefontsize{body}{\small}
...
% request size
\tablefontsize{body} % outputs \small
```

Listing 6: configuration and selection of font sizes .

colors

Colors are defined and requested similar to font sizes, here with the commands `\setuptablecolor` and `\tablecolor`. The color definition must be in the notation of xcolor. Predefined are colors for *body*, *head*, *subhead* and *row*.

```
% definition
\setuptablecolor{body}{white!100}
...
% request color
\tablecolor{body} % outputs white!100
```

Listing 7: configuration and selection of colors.

alternating row colors

Alternating rows are implemented using the `\rowcolors` command of the `colortbl` package (should be loaded by xcolor). Here the colors *body* and *row* are alternated. They are introduced using `\tablealtcolored` and can be disabled (until the style of tables is reset using `\tablestyle`) with `\disablealternatecolors`. The examples below show the effect.

```
\begin{table}[H]
\tablestyle % using \tablealtcolored
\begin{tabular}{*{2}{p{0.25\textwidth}}}}
\tbegin
content & content \\
content & content \\
content & content \\
content & content \\
\tend
\end{tabular}
\end{table}
```

content	content
content	content
content	content
content	content

Listing 8: automatic alternating row colors.

```

\begin{table}[H]
\tablestyle
\disablealternatescolors
\begin{tabular}{*{2}{p{0.25\textwidth}}}}
\tbegin
content & content \\
content & content \\
content & content \\
content & content \\
\tend
\end{tabular}
\end{table}

```

content	content
content	content
content	content
content	content

Listing 9: disabled alternating row colors.

colored lines in tables

Horizontal lines are inserted with `\hline` and vertical lines with `\vline`. These can be changed in color with `\arrayrulecolor{red}\hline` and `\color{red}\vline`. This package provides the commands `\coloredhline{<color>}` and `\coloredvline{<color>}` in order to generalise the setting of line colors. The code listing 10 shows an example of their usage. Note that this example uses `tabularx` and makes use of the new columnstyle ‘C’.

`\coloredhline`
`\coloredvline`

```

\colorlet{grayline}{gray!70}
\definecolor{blueline}{rgb}{0,0.27,0.55}

\begin{table}
\tablestyle
% Overwriting style, instead of defining a new one
\renewcommand{\tbegin}{\coloredhline{blueline}}
\renewcommand{\tbody}{\coloredhline{blueline}}
\renewcommand{\tend}{\coloredhline{blueline}}
\begin{tabularx}{0.8\textwidth}{%
l!\coloredvline{grayline}}
CC!\coloredvline{grayline}}
CC!\coloredvline{grayline}}
C
}
\theadstart
\thead header &
\multicolumn{2}{>{\columncolor{\tablecolor{head}}\thead}
c!\coloredvline{grayline}}}{header} &
\multicolumn{2}{>{\columncolor{\tablecolor{head}}\thead}
c!\coloredvline{grayline}}}{header} &

```



```

\multicolumn{1}{>{\columncolor{\tablecolor{head}}\thead}c}
{header}
%
\tabularnewline
\tbody
%
description & 0,3 & 0,35 & 0,5 & 0,65 & 0,80 \\
description & 0,3 & 0,35 & 0,5 & 0,65 & 0,80 \\
description & 0,3 & 0,35 & 0,5 & 0,65 & 0,80 \\
description & 0,3 & 0,35 & 0,5 & 0,65 & 0,80 \\
\tend
\end{tabularx}
\end{table}

```

Listing 10: disabled alternating row colors.

header	header	header	header	header
description	0,3	0,35	0,5	0,65
description	0,3	0,35	0,5	0,65
description	0,3	0,35	0,5	0,65
description	0,3	0,35	0,5	0,65

Table 3: table with colored vertical and horizontal lines

Table styles

Complete styles are applied using `\tablestyle[style name]` and defined with `\setuptablestyle {style name}{style definition}`. If no argument is given for the command `\tablestyle` the style with name *default* is used. Within the definition of styles the command `\resettablestyle` has a special meaning. Its purpose is to reset a style before the actual style definitions are applied. Any change to this command has thus an influence to all styles using this command.

The default style for example is defined by the code listing 11. It sets up font sizes for body and head (used in `\thead` and `\subhead`), defines the font style using the sans serif font family, redefines the `\arraystretch` and introduces alternating row colors with `\tablealtcolored`.

```

\setuptablestyle{default}{%
\resettablestyle
\renewcommand{\arraystretch}{1.4}
\centering
\sffamily
\upshape%
\tablefontsize{body}
\tablealtcolored%

```

```

\renewcommand{\thead}{%
\tablefontsize{head}%
\ignorespaces%
}
\renewcommand{\tsubhead}{%
\tablefontsize{head}%
\ignorespaces%
}
}

```

Listing 11: definition of the default style.

The package defines the styles

- default (sans serif fonts, alternating row colors, header with gray background color)
- roman (as default style, but with roman fonts)
- sansbold (as default style, but with bold heading font)
- sansboldbw (as sansbold style, but with darker heading background)

These styles are demonstrated by the following tables 4 - 7.

header	header
content	content
content	content
content	content
subhead	subhead
content	content
content	content

Table 4: default style

header	header
content	content
content	content
content	content
subhead	subhead
content	content
content	content

Table 5: roman style

header	header
content	content
content	content
content	content
subhead	subhead
content	content
content	content

Table 6: sansbold style

<i>header</i>	<i>header</i>
content	content
content	content
content	content
subhead	subhead
content	content
content	content

Table 7: sansboldbw style

2.4 Column type definitions

- 1 L^AT_EX defines itself only the columntypes left (l), right(c) and center (c). Unfor-
c tunatley these do not automatically wrap text in cells. even if the text overruns
r the width of the page. For fixed width columns this can be solved using the
p p-columntype. These default columntypes are shown in listing 12 and table 8.
p However a combination of paragraph like columns and non-justified alignment
requires the definition of new column types, which are shown hereafter.

```

\begin{table}
\tablestyle
%
\begin{tabular}{|l|c|r|p{0.3\linewidth}}|}
\theadstart
  \thead \centering header l &
  \thead \centering header c &
  \thead \centering header r &
  \thead \centering header p \tabularnewline
\tbody
%
left & center & right &
text which is considerably longer than the width of the column \\

```

```

\tend
\end{tabular}
\caption{standard column types (lcrp)}
\end{table}

```

Listing 12: example of default column types.

header l	header c	header r	header p
left	center	right	text which is considerably longer than the width of the column

Table 8: standard column types (lcrp)

- m The array package provides further the column types middle (m) and bottom (b),
- b which act like the p-type, but provide a different vertical alignment as demonstrated in listing 13 and table 9.

```

\begin{table}
\tablestyle
%
\begin{tabular}{|p{0.3\linewidth}|m{0.3\linewidth}|b{0.3\linewidth}|}
\theadstart
\thead \centering header p &
\thead \centering header m &
\thead \centering header b \tabularnewline
\tbody
%
text which is considerably longer than the width of the column &
text which is considerably longer than the width of the column &
text which is considerably longer than the width of the column \\
\tend
\end{tabular}
\caption{column types (pmb)}
\end{table}

```

Listing 13: example of paragraph column types.

header p	header m	header b
text which is considerably longer than the width of the column	text which is considerably longer than the width of the column	text which is considerably longer than the width of the column

Table 9: column types (pmb)

- X The tabularx-package provides a further column type (X), which uses automati-
 Z cally calculated column width. This package defines additional column types bases
 Y on X for centered (Z) and right (Y) aligned columns, shown table ??, which is
 using the column definition:

```
\begin{table}
\tablestyle
\begin{tabularx}{\textwidth}{|X|Z|Y|}
...
```

Listing 14: example of tabularx type columns.

header X	header Z	header Y
text which is considerably longer than the width of the column	text which is considerably longer than the width of the column	text which is considerably longer than the width of the column

Table 10: column types (XZY)

- L Furthermore, this package provides additional column types with fixed with and
 C left (L), centered (C) and right (R) aligned column, each derived from the p-
 R columntype. In the example (table ??) the advantage of using ragged text align-
 ment (L) in tables over the justified alignment (p) is clearly visible.

header p	header L	header C	header R	header X
text which is considerably longer than the width of the column	text which is considerably longer than the width of the column	text which is considerably longer than the width of the column	text which is considerably longer than the width of the column	text which is considerably longer than the width of the column

Table 11: column types (LCR)

The column definition for table 11 is shown in listing 15:

```
\begin{table}
\tablestyle
\begin{tabularx}{\textwidth}{|p{0.17\linewidth}|L{0.17\linewidth}|C{0.17\linewidth}|R{0.17\linewidth}|X|}
...
```

Listing 15: example of new p-type columns.

Further column definitions can be defined similar to the definitions used here. The implementation is demonstrated in section 3.6.

2.5 Items in tables

Itemized, enumerated or other list typically introduce a horizontal indentation which should not appear within a tabular environment. Therefore the command `\tableitemize` is introduced to remove the indentation.

`\tableitemize`

header X	header items (X)	header enums (X)
The \LaTeX document preparation system is a special version of Donald Knuth's \TeX program. \TeX is a sophisticated program designed to produce high-quality typesetting, especially for mathematical text.	<ul style="list-style-type: none"> • The \LaTeX document preparation system is a special version of Donald Knuth's \TeX program. • \TeX is a sophisticated program designed to produce high-quality typesetting, • especially for mathematical text. 	<ol style="list-style-type: none"> 1. The \LaTeX document preparation system is a special version of Donald Knuth's \TeX program. 2. \TeX is a sophisticated program designed to produce high-quality typesetting, 3. especially for mathematical text.

Table 12: table item list using `tableitemize`

In table 12 this is realized by inserting the command `\tableitemize` directly before the beginning of a list, as shown in listing 16.

```
...
The ...
&
\tableitemize % <---
\begin{itemize}
\item The ...
\item \TeX{} is a ...
\item especially ...
\end{itemize}
&
\tableitemize % <---
\begin{enumerate}
\item The ...
\item \TeX{} is a ...
\item especially ...
\end{enumerate}
\tabularnewline
...
```

Listing 16: example of `tableitemize`.

3 Implementation

3.1 Setup and packages

```
19 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
20 \ProvidesPackage{tablestyles}
21     [2011/10/01 v0.1 tablestyles]
22 %
```

The following packages are loaded by this package. array is required for most of the tables anyway, xcolor with the option ‘table’ is required for the color definitions (table loads colortbl for all commands of colors in tables. ragged2e is used for the new columntypes and etoolbox for the implementation of the code.

```
23 \RequirePackage{array}
24 \RequirePackage{etoolbox}
25 %% check that xcolor is not loaded
26 \@ifpackageloaded{xcolor}{%
27     \ifcsdef{rowcolors}{%
28         \PackageError{tablestyles}%
29         {%
30             \MessageBreak%
31             Package 'xcolor' must be loaded with the option\MessageBreak%
32             'tables'. Otherwise the command rowcolors\MessageBreak%
33             is not available. \MessageBreak%
34             }{%
35             \endinput%
36             }%
37     }{%
38         % xcolor not loaded
39         \PassOptionsToPackage{table}{xcolor}
40         \RequirePackage{xcolor}
41     }
42 \RequirePackage{ragged2e}
43 \AtBeginDocument{%
44     \@ifpackageloaded{tabularx}{%
45         \PackageError{tablestyles}%
46         {%
47             \MessageBreak%
48             Package 'tabularx' must be loaded, otherwise tablestyles will not
49             work.
50             }%
51     }% end of ifpackageloaded
52 }% end of AtBeginDocument
53 %
```

3.2 Configuration commands

The heart of this package is the definition of the table styles, which are very simply only saved as a command sequence. An error are printed out if a style is requested, which has not been defined before.

```
\setuptablestyle 54 %% define table styles (save the code)
55 \newcommand{\setuptablestyle}[2]{%
56   \ifcsdef{tsty@style@#1}
57   {
58     \PackageWarning{tablestyles}{
59       \MessageBreak
60       table style `#1' is already defined.
61     }{}
62   }{}
63   \csdef{tsty@style@#1}{#2}
64 }
65 %
```

```
\tablestyle 66 %% apply table styles (use the code)
67 \newcommand{\tablestyle}[1][default]{%
68   \ifcsdef{tsty@style@#1}{}{
69     \PackageError{tablestyles}{
70       \MessageBreak
71       table style `#1' is not defined.
72       \endinput%
73     }{}
74   }
75   \csuse{tsty@style@#1}
76 }
77 %
```

Font sizes are save in a special command sequence, so that they can be reused, and identified using their names:

```
\setuptablefontsize 8 %% save font size with name
79 \newcommand{\setuptablefontsize}[2]{
80   \csdef{tsty@font@#1@size}{#2}
81 }
82 %
```

```
\tablefontsize 83 %% apply font size with name
84 \newcommand{\tablefontsize}[1]{
85   \ifcsdef{tsty@font@#1@size}%
86   {\csuse{tsty@font@#1@size}%
```



```

87   }{ % not defined
88     \PackageError{tablestyles}{
89       \MessageBreak
90       table font size `#1' is not defined.
91     \endinput%
92   }{}%
93 }%
94 }%
95 %

```

The default values for the predefined sizes *body* and *head* are set to `\small` since tables should always be printed slightly smaller than the normal text.

```

96 %% define default font sizes
97 \setuptablefontsize{body}{\small}
98 \setuptablefontsize{head}{\small}
99 %

```

The definition of colors is almost identical to the definition of font sizes. This definition is restricted to the `xcolor` way of defining colors.

```

\setuptablecolor0 %% save color with name
101 \newcommand{\setuptablecolor}[2]{%
102   \colorlet{tsty@color@#1}{#2}%
103 }
104 %

```

```

\tablecolor5 %% apply colorwith name
106 \newcommand{\tablecolor}[1]{tsty@color@#1}
107 %

```

The default colors are defined as

```

\tsty@defaultcolors8 %% default colors
109 \newcommand{\tsty@defaultcolors}{
110   \setuptablecolor{body}{white!100}
111   \setuptablecolor{row}{gray!10}
112   \setuptablecolor{head}{gray!25}
113   \setuptablecolor{subhead}{gray!30}
114 }
115 %

```

and executed afterwards.

```

116 %% executed default colors
117 \tsty@defaultcolors
118 %

```

Alternating row colors are simplified with the command `\tablealtcolored`, which reuses the `\rowcolors` and the table color definitions for *row* and *body*.

```
\tablealtcolored9 %% shortcut for alternating rowcolors
120 \newcommand\tablealtcolored{%
121   \rowcolors{1}{\tablecolor{row}}{\tablecolor{body}}%
122 }
123 %
```

The alternating rows are disabled by changing the row color identical to the body color. Since every `tablestyle` should load `\tsty@defaultcolors` this is reset at every applied `tablestyle`.

```
\disablealternatcolors4 %% disable alternating rowcolors by changing the colors
125 \newcommand{\disablealternatcolors}{%
126   \setuptablecolor{row}{\tablecolor{body}}%
127 }
128 %
```

The `\coloredhline` uses the `\arrayrulecolor` of the `colortlb`-package.

```
\coloredhline9 %% colored horizontal lines (hlines)
130 \providecommand{\coloredhline}[1]{%
131   \arrayrulecolor{#1}\hline
132   \arrayrulecolor{black}
133 }%
134 %
```

The `\coloredvline` applies the color to the `\vline` with a standard `\color` command.

```
\coloredvline5 %% colored vertical lines (vlines)
136 \providecommand{\coloredvline}[1]{%
137   \color{#1}\vline
138 }%
139 %
```

3.3 Style applying commands

The lines for the top (above the header), middle (below the header) and the bottom of the table (at the end) are defined with `\tlinetop`, `\tlinemid` and `\tlinebottom`

```

\tlinetop0 %% commands for the style of the lines
141 \newcommand{\tlinetop}{\coloredhline{black}}
142 %

```

```

\tlinemid3 \newcommand{\tlinemid}{\coloredhline{black}}
144 %

```

```

\tlinebottom5 \newcommand{\tlinebottom}{\coloredhline{black}}
146 %

```

These are supposed to be inserted in the table with the commands `\tbegin`, `\tbody` and `\tend`, which prints out the last line and restores color, font and size.

```

\tbegin7 %% commands for the separations (includes the lines)
148 \newcommand{\tbegin}{\tlinetop}
149 %

```

```

tbody60 \newcommand{\tbody}{\tlinemid}
151 %

```

```

\tend2 \newcommand{\tend}{\tlinebottom}
153 %

```

The predefined `\thead` includes only the font size. The same applies for `\tsubhead`

```

\thead4 %% default head style (only fontsize)
155 \newcommand\thead{%
156 \tablefontsize{head}
157 }
158 %

```

```

\tsubhead9 %% default subhead style (only fontsize)
160 \newcommand{\tsubhead}{%
161 \tablefontsize{head}
162 }
163 %

```

Both head and subhead have a start and an end command. By default the start includes a hline and the rowcolor, whereas the end command does nothing.

```

\theadstart 164 %% style of header start (line and row color)
165 \newcommand\theadstart{%
166 \tlinetop%
167 \rowcolor{\tablecolor{head}}%
168 }
169 %

```

```

\theadend 170 %% style of header end - empty
171 \newcommand\theadend{}
172 %

```

```

\tsubheadstart 173 %% style of subheader start (row color)
174 \newcommand{\tsubheadstart}{%
175 \rowcolor{\tablecolor{subhead}}
176 }
177 %

```

```

\tsubheadend 178 %% style of subheader end - empty
179 \newcommand\tsubheadend{}
180 %

```

The commands `\theadrow` and the equivalent `\tsubheadrow` define a multicolumn row for the whole table width. It should be used to have a full width spanning header. This command includes the background color and the cell properties using `\thead` and `\tsubhead`.

```

\theadrow 181 %% shortcut for multi column row in header
182 \newcommand{\theadrow}[2]{%
183 \multicolumn{#1}{>{\columncolor{\tablecolor{head}}}}1}{%
184 \thead #2%
185 }%
186 \tabularnewline%
187 }%
188 %

```

```

\tsubheadrow 189 %% shortcut for multi column row in subheader
190 \newcommand{\tsubheadrow}[2]{%
191 \multicolumn{#1}{>{\columncolor{\tablecolor{subhead}}}}1}{%
192 \tsubhead #2%
193 }%
194 \tabularnewline%
195 }%
196 %

```

3.4 Style definitions

The following `\resettablestyle` is meant to be used with in the definition of a table style. It resets most commands to a default value equivalent to what a table would look like without any style applied. New styles should be built on this basis. The content of this command can be changed. However it means that this changes are applied to all styles of this package.

```
\resettablestyle  %% reset to default styles
198 \newcommand{\resettablestyle}{%
199   \normalcolor\normalfont\normalsize%
200   \renewcommand{\arraystretch}{1}%
201   \rmfamily%
202   \tsty@defaultcolors%
203   \renewcommand{\tlinetop}{\coloredhline{black}}
204   \renewcommand{\tlinemid}{\coloredhline{black}}
205   \renewcommand{\tlinebottom}{\coloredhline{black}}
206   \renewcommand{\theadstart}{\tlinetop\rowcolor{\tablecolor{head}}}
207   \renewcommand{\tbegin}{\tlinetop}
208   \renewcommand{\tbody}{\tlinemid}
209   \renewcommand{\tend}{\tlinebottom}
210 }
211 %
```

default

```
112 %% style: default
113 \setuptablestyle{default}{%
114   \resettablestyle
115   \renewcommand{\arraystretch}{1.4}
116   \centering
117   \sffamily
118   \upshape%
119   \tablefontsize{body}
120   \tablealtcolored%
121   \renewcommand{\thead}{%
122     \tablefontsize{head}%
123     \ignorespaces%
124   }
125   \renewcommand{\tsubhead}{%
126     \tablefontsize{head}%
127     \ignorespaces%
128   }
129 }
130 %
```

roman

This style differs from the default style only in the exchange of the `\sffamily` with `\rmfamily`.

```
231 %% style: roman
232 \setuptablestyle{roman}{%
233   \resettablestyle
234   \renewcommand{\arraystretch}{1.4}
235   \centering
236   \rmfamily
237   \upshape%
238   \tablefontsize{body}
239   \tablealtcolored%
240   \renewcommand{\thead}{%
241     \tablefontsize{head}%
242     \ignorespaces%
243   }
244   \renewcommand{\tsubhead}{%
245     \tablefontsize{head}%
246     \ignorespaces%
247   }
248 }
249 %
```

sansbold

This style differs from the default style in the inclusion of `\bfseries` in the heading commands.

```
250 %% style: sansbold
251 \setuptablestyle{sansbold}{%
252   \resettablestyle
253   \renewcommand{\arraystretch}{1.4}
254   \centering
255   \sffamily
256   \upshape%
257   \tablefontsize{body}
258   \tablealtcolored%
259   \renewcommand{\thead}{%
260     \tablefontsize{head}%
261     \bfseries
262     \upshape%
263     \ignorespaces%
264   }
265   \renewcommand{\tsubhead}{%
266     \tablefontsize{head}%
267     \upshape%
268     \ignorespaces%
269   }
270 }
```

```

269 }
270 }
271 %

```

sansboldbw

This style differs from the sansbold style in exchange of the table colors with much darker heading colors and a white heading font, which is printed slanted.

```

272 %% style: sansboldbw
273 \setuptablestyle{sansboldbw}{%
274   \resettablestyle
275   \setuptablecolor{body}{white!100}
276   \setuptablecolor{row}{gray!15}
277   \setuptablecolor{head}{black!60}
278   \setuptablecolor{subhead}{gray!40}
279 %

```

```

280   \renewcommand{\arraystretch}{1.4}
281   \centering
282   \sffamily
283   \upshape%
284   \tablefontsize{body}
285   \tablealtcolored%
286   \renewcommand{\thead}{%
287     \leavevmode\color{white}
288     \tablefontsize{head}%
289     \bfseries
290     \slshape
291     \ignorespaces%
292   }
293   \renewcommand{\tsubhead}{%
294     \tablefontsize{head}%
295     \bfseries
296     \ignorespaces%
297   }
298 }
299 %

```

3.5 Support for itemized lists in tables

```

\tsty@removeindentation0 %% bugfix code for lists in tables
301 \newcommand{\tsty@removeindentation}{%
302   \leftmargini=\labelsep%
303   \advance\leftmargini by \labelsep%
304 }
305 %

```

```

\tableitemize6 \newcommand\tableitemize{
307   \@minipagetrue%
308   \tsty@removeindentation
309 }
310 %

```

3.6 New columntype definitions

The `\arraybackslash` is required to restore the definition of `\` after its redefinition. `\hspace{0pt}` is included to allow hyphenation also of the first word.

New columntype definitions:

```

1 %% New columntype definitions:
2 %% centered (X):
3 \newcolumntype{Z}{>{\centering\arraybackslash\hspace{0pt}}X}
4 %% right (X):
5 \newcolumntype{Y}{>{\RaggedLeft\arraybackslash\hspace{0pt}}X}
6 %% left (X):
7 \newcolumntype{W}{>{\RaggedRight\arraybackslash\hspace{0pt}}X}
8 %% left (p):
9 \newcolumntype{L}[1]{>{\RaggedRight\arraybackslash\hspace{0pt}}p{#1}}
10 %% right (p):
11 \newcolumntype{R}[1]{>{\RaggedLeft\arraybackslash\hspace{0pt}}p{#1}}
12 %% centered (p):
13 \newcolumntype{C}[1]{>{\Centering\arraybackslash\hspace{0pt}}p{#1}}
14 %

```

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

B		L		<code>\r</code> <i>11</i>
<code>\b</code> <i>12</i>		<code>\L</code> <i>13</i>		<code>\resettablestyle</code> . <u>1</u> , <i>9</i>
		<code>\l</code> <i>11</i>		
C		M		S
<code>\C</code> <i>13</i>				<code>\setuptablecolor</code> . <u>1</u> , <i>6</i>
<code>\c</code> <i>11</i>		<code>\m</code> <i>12</i>		<code>\setuptablefontsize</code>
<code>\coloredhline</code> <u>1</u> , <i>8</i>			 <u>1</u> , <i>6</i>
<code>\coloredvline</code> <u>1</u> , <i>8</i>		P		<code>\setuptablestyle</code> . <u>1</u> , <i>9</i>
		<code>\p</code> <i>11</i>		
D		T		
<code>\disablealternatcolors</code>		R		<code>\tablealtcolored</code> . <u>1</u> , <i>7</i>
..... <u>1</u> , <i>7</i>		<code>\R</code> <i>13</i>		<code>\tablecolor</code> <u>1</u> , <i>6</i>

<code>\tablefontsize</code> ... 1 , 6	<code>\tlinebottom</code> 1	X	
<code>\tableitemize</code> 1 , 13	<code>\tlinemid</code> 1	<code>\X</code>	12
<code>\tablestyle</code> 1 , 4 , 9	<code>\tlinetop</code> 1		
<code>\tbegin</code> 1 , 4	<code>\tsty@defaultcolors</code> . 1		
<code>\tbody</code> 1 , 4	<code>\tsty@removeindentation</code>	Y	
<code>\tend</code> 1 , 4 1	<code>\Y</code>	12
<code>\thead</code> 1 , 5	<code>\tsubhead</code> 1 , 5		
<code>\theadend</code> 1 , 5	<code>\tsubheadend</code> 1 , 5		
<code>\theadrow</code> 1 , 5	<code>\tsubheadrow</code> 1 , 5	Z	
<code>\theadstart</code> 1 , 5	<code>\tsubheadstart</code> ... 1 , 5	<code>\Z</code>	12