



Tabularray

基于 L^AT_EX³ 的表格 (Tabulars) 和阵列 (Arrays) 排版宏包

Author

Jianrui Lyu (tolvjr@163.com)

翻译

耿楠 (nangeng@nwafu.edu.cn)

Version

2021M (2021-08-03)

Code

<https://github.com/lvjrr/tabularray>

Code

<https://bitbucket.org/lvjrr/tabularray>

Issue

<https://github.com/lvjrr/tabularray/issues>

Forum

<https://github.com/lvjrr/tabularray/discussions>



```
\begin{tblr}{
  colspec = {lX}, colsep = 12mm, hlines = {2pt, white},
  row{odd} = {azure8}, row{even} = {gray8},
  row{1} = {6em,azure2,fg=white,font=\LARGE\bfseries\sffamily},
  row{2-Z} = {3em,font=\Large},
}
Tabularray & 基于\LaTeX3 的表格 (Tabulars) 和阵列 (Arrays) 排版宏包 \\
Author     & Jianrui Lyu (tolvjr@163.com) \\
翻译      & 耿楠 (nangeng@nwafu.edu.cn) \\
Version    & \myversion\ (\the\year-\mylpad\month-\mylpad\day) \\
Code       & \url{https://github.com/lvjr/tabularray} \\
Code       & \url{https://bitbucket.org/lvjr/tabularray} \\
Issue      & \url{https://github.com/lvjr/tabularray/issues} \\
Forum      & \url{https://github.com/lvjr/tabularray/discussions} \\
\end{tblr}
```

目录

第一章 概述	3
1.1 垂直间距	3
1.2 多行单元格	4
1.3 单元格对齐方式	4
1.4 单元格行合并	5
1.5 单元格行合并和列合并	7
1.6 列格式	8
1.7 行格式	9
1.8 表格横线与表格竖线	10
1.9 彩色表格	10
第二章 用户接口	12
2.1 新旧用户接口	12
2.2 表格横线和竖线	12
2.3 单元格与单元格合并选项	16
2.4 行 (rows) 和列 (columns) 选项	19
2.5 colspec 和 rowspec 选项	22
第三章 附加接口	24
3.1 表格参数	24
3.2 参数默认值	27
3.3 定义表格新命令	28
3.4 表格内容宏的提前展开	28
3.5 原文排版命令	29
3.6 计数器和长度	29
3.7 跟踪 Tabularray	29
第四章 长表格	30
4.1 简单示例	30
4.2 个性化模板	36
4.3 改变样式	40
4.4 定义主题	41
4.5 分页控制	41
4.6 添加表注	41

目录	2
第五章 使用扩展库	42
5.1 booktabs 库	42
5.2 diagbox 库	42
5.3 siunitx 库	43
第六章 宏包源码	45

第一章 概述

1.1 垂直间距

在导言区加载`tabulararray`宏包后，可以使用`tblr`环境排版表格 (tabulars) 或阵列 (arrays)。环境`tblr`的名称是`tabulararray`或`top-bottom-left-right`的缩写。以下示例展示了常规`tabular`环境与`tblr`环境排版表格的异同：

```
\begin{tabular}{lccr}
\hline
Alpha & Beta & Gamma & Delta \\
\hline
Epsilon & Zeta & Eta & Theta \\
\hline
Iota & Kappa & Lambda & Mu \\
\hline
\end{tabular}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

```
\begin{tblr}{lccr}
\hline
Alpha & Beta & Gamma & Delta \\
\hline
Epsilon & Zeta & Eta & Theta \\
\hline
Iota & Kappa & Lambda & Mu \\
\hline
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

显然，用`tblr`环境排版表格时，在表格各行的上方和下方会增加额外垂直间距。该间距使表格排版更为美观。当然，也可以使用`\SetTblrDefault`命令根据需要进行调整该垂直间距，例如，可以用如下代码取消间距：

```
\SetTblrInner{rowsep=Opt}
\begin{tblr}{lccr}
\hline
Alpha & Beta & Gamma & Delta \\
\hline
Epsilon & Zeta & Eta & Theta \\
\hline
Iota & Kappa & Lambda & Mu \\
\hline
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

但多数情况下，使用`rowsep`间距将使表格排版更为美观，例如如下带有大量分式的表格排版：

```


$$\begin{array}{rrr}
\hline
\frac{2}{3} & \frac{2}{3} & \frac{1}{3} \\
\frac{2}{3} & -\frac{1}{3} & -\frac{2}{3} \\
\frac{1}{3} & -\frac{2}{3} & \frac{2}{3} \\
\hline
\end{array}$$


```

$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{3}$
$\frac{2}{3}$	$-\frac{1}{3}$	$-\frac{2}{3}$
$\frac{1}{3}$	$-\frac{2}{3}$	$\frac{2}{3}$
$\frac{1}{3}$	$-\frac{2}{3}$	$\frac{2}{3}$

```


$$\begin{tblr}{rrr}
\hline
\frac{2}{3} & \frac{2}{3} & \frac{1}{3} \\
\frac{2}{3} & -\frac{1}{3} & -\frac{2}{3} \\
\frac{1}{3} & -\frac{2}{3} & \frac{2}{3} \\
\hline
\end{tblr}$$


```

$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{3}$
$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$
$\frac{2}{3}$	$-\frac{1}{3}$	$-\frac{2}{3}$
$\frac{1}{3}$	$-\frac{2}{3}$	$\frac{2}{3}$
$\frac{1}{3}$	$-\frac{2}{3}$	$\frac{2}{3}$

需要说明的是，tblr环境既可以用于文本模式，也可以用于数学模式。

1.2 多行单元格

在使用tblr环境排版表格时，不使用定宽列格式即可实现多行单元格排版。此时，仅需用大括号将需要多行排版的单元格包围起来，然后在大括号内的内容中使用\\手动断行即可，如：

```


$$\begin{tblr}{|l|c|r|}
\hline
Left & {Center \\ Cent \\ C} & {Right \\ R} \\
\hline
{L \\ Left} & {C \\ Cent \\ Center} & R \\
\hline
\end{tblr}$$


```

Left	Center Cent C	Right R
L Left	C Cent Center	R

1.3 单元格对齐方式

Tabularray宏包提供了Q列格式用于同时指定单元格水平和垂直对齐方式。其实，Q格式是Tabularray宏包提供的唯一列格式的元格式，其它的列格式都是通过为Q元格式设置不同参数定义的。

```


$$\begin{tblr}{|Q[l,t]|Q[c,m]|Q[r,b]|}
\hline
{Top Baseline \\ Left Left} & Middle Center & {Right Right \\ Bottom Baseline} \\
\hline
\end{tblr}$$


```

		Right Right
Top Baseline	Middle Center	Bottom Baseline
Left Left		

注意，对于顶端基线对齐，可使用含义更为明确的t格式替代p格式。对于熟悉字处理工具的用户而言，类似t和b这些格式是违背直觉的，其中，t是顶端与基线对齐，不是顶端对齐，而b是底端与基线对齐，不是底端对齐。在Tabularray宏包中，还定义了h和f格式，以分别实现常规意义上的

顶端对齐和底端对齐：

```
\begin{tblr}{Q[h,4em]Q[t,4em]Q[m,4em]Q[b,4em]Q[f,4em]}
\hline
{row\head} & {top\line} & {middle} & {line\bottom} & {row\foot} \\
\hline
{row\head} & {top\line} & {11\22\mid\44\55} & {line\bottom} & {row\foot} \\
\\
\hline
\end{tblr}
```

row			line	
head	top	middle	bottom	row
	line			foot
<hr/>				
row		11		
head		22	line	
	top	mid	bottom	
	line	44		row
		55		foot
<hr/>				

1.4 单元格行合并

在Tabularray中，当用\SetCell命令进行单元格行合并时，h和f对齐格式是非常必要的。

```
\begin{tabular}{|l|l|l|l|}
\hline
\multirow[t]{4}{1.5cm}{Multirow Cell One} & Alpha & & \\
\multirow[b]{4}{1.5cm}{Multirow Cell Two} & Alpha & & \\
& Beta & & Beta \\
& Gamma & & Gamma \\
& Delta & & Delta \\
\hline
\end{tabular}
```

Multirow Cell One	Alpha Beta Gamma Delta		Alpha Beta Gamma Delta
		Multirow Cell Two	

```
\begin{tblr}{|l|l|l|l|}
\hline
\SetCell[r=4]{h,1.5cm} Multirow Cell One & Alpha &
\SetCell[r=4]{f,1.5cm} Multirow Cell Two & Alpha \\
& Beta & & Beta \\
& Gamma & & Gamma \\
& Delta & & Delta \\
\hline
\end{tblr}
```

Multirow Cell One	Alpha		Alpha
	Beta		Beta
	Gamma	Multirow	Gamma
	Delta	Cell Two	Delta

注意,由于Tabularray宏包并不依赖于multirow宏包,因此在行合并前并不需要载入multirow宏包。此外, Tabularray宏包将自动处理单元格行合并后的垂直对齐, 这使用即便是在某些行的行高较大时, 也能够正确地实现行合并后的垂直居中对齐。

```
\begin{tabular}{|l|m{4em}|}
\hline
\multirow[c]{4}{1.5cm}{Multirow} & Alpha \\
& Beta \\
& Gamma \\
& Delta Delta Delta \\
\hline
\end{tabular}
```

Multirow	Alpha
	Beta
	Gamma
	Delta
	Delta
	Delta

```
\begin{tblr}{|l|m{4em}|}
\hline
\SetCell[r=4]{m,1.5cm} Multirow & Alpha \\
& Beta \\
& Gamma \\
& Delta Delta Delta \\
\hline
\end{tblr}
```

Multirow	Alpha
	Beta
	Gamma
	Delta
	Delta
	Delta

同时, 如果行合并后行高过大, 则Tabularray会自动扩展合并后单元格的行高, 从而确保不会发生文字垂直溢出现象。

```
\begin{tabular}{|l|m{4em}|}
\hline
\multirow[c]{2}{1cm}{Line \\ Line} & Alpha \\
\cline{2-2}
& Beta \\
\hline
\end{tabular}
```

Line	Alpha
Line	Beta

Line

Line


```

\begin{tblr}{|l|m{4em}|}
\hline
\SetCell[r=2]{m,1cm} {Line \ Line \ Line \ Line} & Alpha \
\cline{2}
& Beta \
\hline
\end{tblr}

```

Line	Alpha
Line	
Line	Beta
Line	

1.5 单元格行合并和列合并

同时实现单元格行合并和列合并是比较困难的，例如：

```

\begin{tabular}{|c|c|c|c|c|}
\hline
\multirow{2}{*}{2 Rows}
& \multicolumn{2}{c|}{2 Columns}
& \multicolumn{2}{c|}{\multirow{2}{*}{2 Rows 2 Columns}} \
\cline{2-3}
& 2-2 & 2-3 & \multicolumn{2}{c|}{} \
\hline
3-1 & 3-2 & 3-3 & 3-4 & 3-5 \
\hline
\end{tabular}

```

2 Rows	2 Columns		2 Rows 2 Columns	
	2-2	2-3		
3-1	3-2	3-3	3-4	3-5

在Tabularray宏包中，可以使用\SetCell命令实现单元格合并。在\SetCell命令的可选参数中，r选项用于指定需要合并的行数，c选项用于指定需要合并的列数。在\SetCell命令的必选参数中，可以指定合并后的水平和垂直对齐方式。因此，在Tabularray中，单元格合并更为简捷：

```

\begin{tblr}{|c|c|c|c|c|}
\hline
\SetCell[r=2]{c} 2 Rows
& \SetCell[c=2]{c} 2 Columns
& & \SetCell[r=2,c=2]{c} 2 Rows 2 Columns & \\
\hline
& 2-2 & 2-3 & & \\
\hline
3-1 & 3-2 & 3-3 & 3-4 & 3-5 \\
\hline
\end{tblr}

```

2 Rows	2 Columns		2 Rows 2 Columns	
	2-2	2-3		
3-1	3-2	3-3	3-4	3-5

使用`\multicolumn`命令时，**必须删除**需要合并的其它单元格。相反，在使用`\multirow`命令时，则**必须保留**需要合并的其它单元格。`\SetCell`命令则与`\multirow`命令相同。

使用`tblr`环境时，由于会自动忽略合并单元格的`\hline`线段，因此，在以上的示例中，可以直接使用`\hline`命令绘制表格横线。同时，无论单元格是否留空，任何合并时省略的单元格在排版时都会被忽略。基于此，在排版表格时，可以将行号、列号或其它标识性内容写入需要省略的单元格，这有助于在排版复杂表格时的单元格定位，如：

```

\begin{tblr}{|l|l|c|rr|}
\hline
\SetCell[r=3,c=2]{h} r=3 c=2 & 1-2 & \SetCell[r=2,c=3]{r} r=2 c=3 & 1-4 & 1-5 \\
2-1 & 2-2 & 2-3 & 2-4 & 2-5 \\
\hline
3-1 & 3-2 & MIDDLE & \SetCell[r=3,c=2]{f} r=3 c=2 & 3-5 \\
\hline
\SetCell[r=2,c=3]{l} r=2 c=3 & 4-2 & 4-3 & 4-4 & 4-5 \\
5-1 & 5-2 & 5-3 & 5-4 & 5-5 \\
\hline
\end{tblr}

```

r=3 c=2	r=2 c=3	
	MIDDLE	
r=2 c=3		r=3 c=2

1.6 列格式

`Tabularray`宏包支持所有常规的列格式，包括第一次出现在`tabularx`宏包中，并被`tabu`宏包优化后的可扩展的`X`列格式。

```
\begin{tblr}{|X[2,1]|X[3,1]|X[1,r]|X[r]|}
\hline
Alpha & Beta & Gamma & Delta \\
\hline
\end{tblr}
```

Alpha	Beta	Gamma	Delta
-------	------	-------	-------

此外，还可以在X列格式中使用负系数：

```
\begin{tblr}{|X[2,1]|X[3,1]|X[-1,r]|X[r]|}
\hline
Alpha & Beta & Gamma & Delta \\
\hline
\end{tblr}
```

Alpha	Beta	Gamma	Delta
-------	------	-------	-------

在使用X列格式排版表格时，需要指定表格宽度。如未设置，则默认为 `\linewidth`。如需更改表格宽度，则必须将所有列格式置入 `colspec={...}` 选项的参数中：

```
\begin{tblr}{width=0.8\linewidth,colspec={|X[2,1]|X[3,1]|X[-1,r]|X[r]|}}
\hline
Alpha & Beta & Gamma & Delta \\
\hline
\end{tblr}
```

Alpha	Beta	Gamma	Delta
-------	------	-------	-------

可以使用 `\NewColumnType` 命令定义新的列格式。例如，在 `Tabularray` 宏包中，`b` 和 `X` 列格式是用 `Q` 元格式通过指定必要参数实现的：

```
\NewColumnType{b}[1]{Q[b,wd=#1]}
\NewColumnType{X}[1][ ]{Q[co=1,#1]}
```

1.7 行格式

除了可以使用 `colspec` 选项指定列格式外，也可以通过 `rowspec` 选项指定行格式，如：

```
\begin{tblr}{colspec={Q[1]Q[c]Q[r]},rowspec={|Q[t]|Q[m]|Q[b]|}}
{Alpha \\ Alpha} & Beta & & Gamma \\
Delta & & Epsilon & {Zeta \\ Zeta} \\
Eta & & {Theta \\ Theta} & Iota \\
\end{tblr}
```

Alpha	Beta	Gamma
Alpha		
Delta	Epsilon	Zeta
		Zeta
	Theta	
Eta	Theta	Iota

与列格式类似，Q 是唯一的行格式的元格式，其它行格式都是通过为 Q 元格式指定不同参数实现的。强烈建议在 `colspec` 中指定水平对齐方式，在 `rowspec` 中指定垂直对齐方式。

在 `rowspec` 中，| 用于指定表格横线格式。因此，无需再在表格内容中使用 `\hline` 命令，这会使表格代码更为清晰，代码可维护性更强。

1.8 表格横线与表格竖线

在 `Tabularray` 宏包中，重新设计了表格横线和竖线命令，可以通过参数指定其的宽度、线型、颜色等样式：

```
\begin{tblr}{|l|[dotted]| [2pt]c|r|[solid]| [dashed] |}
\hline
One & Two & Three \\
\hline\hline[dotted]\hline
Four & Five & Six \\
\hline[dashed]\hline[1pt]
Seven & Eight & Nine \\
\hline
\end{tblr}
```

One	Two	Three
Four	Five	Six
Seven	Eight	Nine

1.9 彩色表格

如需排版彩色表格，则需加载 `xcolor` 宏包，`Tabularray` 宏包一旦发现用户加载了 `xcolor` 宏包，则会自动加载 `ninecolors` 宏包，以使用前景与背景对比度更为合理的颜色。例如，可以在 `rowspec/colspec` 选项中通过 Q 元格式的背景参数为行/列指定背景色：

```
\begin{tblr}{colspec={lcr},rowspec={|Q[cyan7]|Q[azure7]|Q[blue7]|}}
Alpha & Beta & Gamma \\
Epsilon & Zeta & Eta \\
Iota & Kappa & Lambda \\
\end{tblr}
```

Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda

```
\begin{tblr}{colspec={Q[l,brown7]Q[c,yellow7]Q[r,olive7]},rowspec={|Q|Q|Q|}}
Alpha & Beta & Gamma \\
Epsilon & Zeta & Eta \\
Iota & Kappa & Lambda \\
\end{tblr}
```

Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda

当然，也可以使用 `\SetRow` 或 `\SetColumn` 命令统一为指定的行或列设置颜色：

```
\begin{tblr}{colspec={lcr},rowspec={|Q|Q|Q|}}
\SetRow{cyan7} Alpha & Beta & Gamma \\
\SetRow{azure7} Epsilon & Zeta & Eta \\
\SetRow{blue7} Iota & Kappa & Lambda \\
\end{tblr}
```

Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda

```
\begin{tblr}{colspec={lcr},rowspec={|Q|Q|Q|}}
\SetColumn{brown7}
Alpha & \SetColumn{yellow7}
Beta & \SetColumn{olive7}
Gamma \\
Epsilon & Zeta & Eta \\
Iota & Kappa & Lambda \\
\end{tblr}
```

Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda

此外，还可以为表格横线和竖线指定颜色：

```
\begin{tblr}{colspec={lcr},rowspec={|[2pt,green7]Q|[teal7]Q|[green7]Q|[3pt,teal7]}}
Alpha & Beta & Gamma \\
Epsilon & Zeta & Eta \\
Iota & Kappa & Lambda \\
\end{tblr}
```

Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda

```
\begin{tblr}{colspec={|[2pt,violet5]l|[2pt,magenta5]c|[2pt,purple5]r|[2pt,red5]}}
Alpha & Beta & Gamma \\
Epsilon & Zeta & Eta \\
Iota & Kappa & Lambda \\
\end{tblr}
```

Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda

第二章 用户接口

2.1 新旧用户接口

在 `Tabularray` 宏包中，可以通过用户接口实现对表格样式的控制。

旧的用户接口由置于表格内容中的一系列命令组成。此时，与 `tabular` 和 `array` 环境类似，所有的命令**必须**用于单元格的文本内容之前。另外，如果需要新的表格命令，则**必须**使用 `\NewTableCommand` 命令进行定义。

新的用户接口由 `tblr` 环境必选参数的选项实现，因此，使用新用户接口可以实现表格内容与格式的完全分离。

表 2.1: 新旧用户接口

旧接口	新接口
<code>\SetHlines</code>	<code>hlines</code>
<code>\SetHline</code> , <code>\hline</code> , <code>\cline</code>	<code>hline</code> , <code>rowspec</code>
<code>\SetVlines</code>	<code>vlines</code>
<code>\SetVline</code> , <code>\vline</code> , <code>\rline</code>	<code>vline</code> , <code>colspec</code>
<code>\SetCells</code>	<code>cells</code>
<code>\SetCell</code>	<code>cell</code>
<code>\SetRows</code>	<code>rows</code>
<code>\SetRow</code>	<code>row</code> , <code>rowspec</code>
<code>\SetColumns</code>	<code>columns</code>
<code>\SetColumn</code>	<code>column</code> , <code>colspec</code>

2.2 表格横线和竖线

表格横线和竖线 `hlines` 和 `vlines` 选项的所有有效键值参见表 2.2。

表 2.2: hlines 和 vlines 的键值

键	说明与可选键值	初始值
<code>dash</code>	线型: solid, dashed or dotted	solid
<code>text</code>	hline/vline 的替换文本 (如在 <code>colspec</code> 中将竖线替换为!)	×
<code>wd</code>	线宽	×
<code>fg</code>	颜色	×

注意: 多数情况下, 对于带有下划线的键, 可以省略键名而只给出键值。

2.2.1 新接口中的表格横线与竖线

`hlines` 和 `vlines` 选项分别用于设置表格所有横线与竖线样式。如果其键值留空, 则会把表格所有横线/竖线都设置为实线。

```
\begin{tblr}{hlines,vlines}
Alpha & Beta & Gamma & Delta \\
Epsilon & Zeta & Eta & Theta \\
Iota & Kappa & Lambda & Mu \\
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

可以在一对大括号中通过参数指定横线/竖线的样式。

```
\begin{tblr}{
hlines = {1pt,solid}, vlines = {red3,dashed},
}
Alpha & Beta & Gamma & Delta \\
Epsilon & Zeta & Eta & Theta \\
Iota & Kappa & Lambda & Mu \\
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

也可以在大括号前用另一对大括号选择需要设置的横线/竖线的单元格线段。

```
\begin{tblr}{
vlines = {1,3,5}{dashed},
vlines = {2,4,6}{solid},
}
Alpha & Beta & Gamma & Delta \\
Epsilon & Zeta & Eta & Theta \\
Iota & Kappa & Lambda & Mu \\
Nu & Xi & Omicron & Pi \\
Rho & Sigma & Tau & Upsilon \\
Phi & Chi & Psi & Omega \\
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu
Nu	Xi	Omicron	Pi
Rho	Sigma	Tau	Upsilon
Phi	Chi	Psi	Omega

以上示例也可以简单地通过设置`odd`和`even`选择算子进行选择 (可以通过 `\NewChildSelector` 命令定义选择算子, 高级用户请阅读 `Tabularray` 源代码, 通过模仿实现需要的选择算子的定义)。

```

\begin{tblr}{
  vlines = {odd}{dashed},
  vlines = {even}{solid},
}
Alpha & Beta & Gamma & Delta & \\
Epsilon & Zeta & Eta & Theta & \\
Iota & Kappa & Lambda & Mu & \\
Nu & Xi & Omicron & Pi & \\
Rho & Sigma & Tau & Upsilon & \\
Phi & Chi & Psi & Omega & \\
\end{tblr}

```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu
Nu	Xi	Omicron	Pi
Rho	Sigma	Tau	Upsilon
Phi	Chi	Psi	Omega

可以再增加一对大括号用于设置多重横线/竖线 (其中, - 表示选择所有单元格线段)。

```

\begin{tblr}{
  hlines = {1}{-}{dashed},
  hlines = {2}{-}{solid},
}
Alpha & Beta & Gamma & Delta & \\
Epsilon & Zeta & Eta & Theta & \\
Iota & Kappa & Lambda & Mu & \\
\end{tblr}

```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

`hline{i}`和`vline{j}`选项分别用于设置指定的横线/竖线, `i`或`j`的值与 `hlines` and `vlines` 的参数含义相同:

```

\begin{tblr}{
  hline{1,7} = {1pt,solid},
  hline{3-5} = {blue3,dashed},
  vline{1,5} = {3-4}{dotted},
}
Alpha & Beta & Gamma & Delta & \\
Epsilon & Zeta & Eta & Theta & \\
Iota & Kappa & Lambda & Mu & \\
Nu & Xi & Omicron & Pi & \\
Rho & Sigma & Tau & Upsilon & \\
Phi & Chi & Psi & Omega & \\
\end{tblr}

```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu
Nu	Xi	Omicron	Pi
Rho	Sigma	Tau	Upsilon
Phi	Chi	Psi	Omega

也可以使用`X`、`Y`和`Z`分别指定最后三条横线/竖线的样式, 这在排版长表格时特别有用。

```

\begin{tblr}{
  hline{1,Z} = {2pt},
  hline{2,Y} = {1pt},
  hline{3-X} = {dashed},
}
Alpha & Beta & Gamma & Delta & \\
Epsilon & Zeta & Eta & Theta & \\
Iota & Kappa & Lambda & Mu & \\
Nu & Xi & Omicron & Pi & \\
Rho & Sigma & Tau & Upsilon & \\
Phi & Chi & Psi & Omega & \\
\end{tblr}

```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu
Nu	Xi	Omicron	Pi
Rho	Sigma	Tau	Upsilon
Phi	Chi	Psi	Omega

下面的示例演示了 `text` 键的使用方式*：

```
\begin{tblr}{
  vlines, hlines,
  colspec = {lX[c]X[c]X[c]X[c]},
  vline{2} = {1}{text=\clap{:}},
  vline{3} = {1}{text=\clap{\ch{+}}},
  vline{4} = {1}{text=\clap{\ch{->}}},
  vline{5} = {1}{text=\clap{\ch{+}}},
}
Equation & \ch{CH4} & \ch{2 O2} & \ch{CO2} & \ch{2 H2O} \\
Initial  & $n_1$      & $n_2$      & 0          & 0          \\
Final    & $n_1-x$    & $n_2-2x$   & $x$        & $2x$      \\
\end{tblr}
```

Equation :	CH ₄	+	2 O ₂	→	CO ₂	+	2 H ₂ O
Initial	n_1		n_2		0		0
Final	$n_1 - x$		$n_2 - 2x$		x		$2x$

注意，为使用 `\ch` 命令，必须载入 `chemmacros` 宏包。

2.2.2 旧接口中的表格横线与竖线

可以在 `\hline` 命令中通过选项指定横线样式，其有效的键值见表 2.2。

```
\begin{tblr}{llll}
\hline
Alpha & Beta & Gamma & Delta \\
\hline[dashed]
Epsilon & Zeta & Eta & Theta \\
\hline[dotted]
Iota & Kappa & Lambda & Mu \\
\hline[2pt,blue5]
\end{tblr}
```

Alpha	Beta	Gamma	Delta

Epsilon	Zeta	Eta	Theta
.....			
Iota	Kappa	Lambda	Mu
=====			

`\cline` 命令的选项与 `\hline` 相同。

```
\begin{tblr}{llll}
\cline{1-4}
Alpha & Beta & Gamma & Delta \\
\cline[dashed]{1,3}
Epsilon & Zeta & Eta & Theta \\
\cline[dashed]{2,4}
Iota & Kappa & Lambda & Mu \\
\cline[2pt,blue5]{-}
\end{tblr}
```

Alpha	Beta	Gamma	Delta
-----		-----	
Epsilon	Zeta	Eta	Theta
-----			-----
Iota	Kappa	Lambda	Mu
=====			

也可以在 `\cline` 命令的必选参数中通过选择算子实现行/列的选择。

*代码来自 <https://tex.stackexchange.com/questions/603023/tabularray-and-tabularx-column-separator>。

```

\begin{tblr}{l1l1l1}
\cline{1-4}
Alpha & Beta & Gamma & Delta \\
\cline[dashed]{odd}
Epsilon & Zeta & Eta & Theta \\
\cline[dashed]{even}
Iota & Kappa & Lambda & Mu \\
\cline[2pt,blue5]{-}
\end{tblr}

```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

\SetHline 命令组合了 \hline 和 \cline 命令：

```

\begin{tblr}{l1l1l1}
\SetHline{1-3}{blue5,1pt}
Alpha & Beta & Gamma & Delta \\
Epsilon & Zeta & Eta & Theta \\
Iota & Kappa & Lambda & Mu \\
\SetHline{2-4}{teal5,1pt}
\end{tblr}

```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

```

\begin{tblr}{l1l1l1}
\SetHline[1]{1-3}{blue5,1pt}
\SetHline[2]{1-3}{azure5,1pt}
Alpha & Beta & Gamma & Delta \\
Epsilon & Zeta & Eta & Theta \\
Iota & Kappa & Lambda & Mu \\
\SetHline[1]{2-4}{teal5,1pt}
\SetHline[2]{2-4}{green5,1pt}
\end{tblr}

```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

本质上, 在第*i*行前使用的表格命令 `\SetHline[<index>]{<columns>}{<styles>}` 与表格选项 `hline{i}={<index>}{<columns>}{<styles>}` 的作用完全相同。

同样, 在某些行前使用的表格命令 `\SetHlines[<index>]{<columns>}{<styles>}` 与表格选项 `hlines={<index>}{<columns>}{<styles>}` 的作用完全相同。

`\vline`、`\rline`、`\SetVline`和 `\SetVlines`命令的使用方法分别与 `\hline`、`\cline`、`\SetHline`、`\SetHlines`的使用方法相同。但通常情况下, 一般不建议直接使用这些命令。

2.3 单元格与单元格合并选项

单元格选项的所有有效键值见表2.3和表 2.4.

表 2.3: cells 键与键值

键	说明与可选键值	初始值
<code>halign</code>	水平对齐方式: l (left), c (center) 或 r (right)	l
<code>valign</code>	垂直对齐方式: t (top), m (middle), b (bottom), h (head) 或 f (foot)	t
<code>wd</code>	宽度	×
<code>bg</code>	背景颜色	×

续下页

表 2.3: cells 键与键值 (接前页)

键	说明与可选键值	初始值
fg	前景颜色	×
font	字体命令	×
preto	单元格前导文本	×
appto	单元格附加文本	×
cmd	对单元格文本要执行的命令	×

注意: 多数情况下, 对于带有下列划线的键, 可以省略键名而只给出键值。

表 2.4: 单元格合并键与键值

键	说明与可选键值	初始值
r	合并行数	1
c	合并列数	1

2.3.1 单元格与单元格合并新接口

cells 选项用于设置所有单元格样式。

```
\begin{tblr}{hlines={white},cells={c,blue7}}
Alpha & Beta & Gamma & Delta \\
Epsilon & Zeta & Eta & Theta \\
Iota & Kappa & Lambda & Mu \\
Nu & Xi & Omicron & Pi \\
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu
Nu	Xi	Omicron	Pi

cell{i}{j} 选项用于设置指定单元格样式。

```
\begin{tblr}{
hlines = {white},
vlines = {white},
cell{1,6}{odd} = {teal7},
cell{1,6}{even} = {green7},
cell{2,4}{1,4} = {red7},
cell{3,5}{1,4} = {purple7},
cell{2}{2} = {r=4,c=2}{c,azure7},
}
Alpha & Beta & Gamma & Delta \\
Epsilon & Zeta & Eta & Theta \\
Iota & Kappa & Lambda & Mu \\
Nu & Xi & Omicron & Pi \\
Rho & Sigma & Tau & Upsilon \\
Phi & Chi & Psi & Omega \\
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta		Theta
Iota			Mu
Nu			Pi
Rho			Upsilon
Phi	Chi	Psi	Omega

2.3.2 单元格与单元格合并旧接口

`\SetCell`命令的必选参数用于设置当前单元格的样式。其键与键值的有效值见表 2.3.

```
\begin{tblr}{llll}
\hline[1pt]
Alpha & \SetCell{bg=teal2,fg=white} Beta & Gamma \\
\hline
Epsilon & Zeta & \SetCell{r,font=\scshape} Eta \\
\hline
Iota & Kappa & Lambda \\
\hline[1pt]
\end{tblr}
```

Alpha	Beta	Gamma
Epsilon	Zeta	ETA
Iota	Kappa	Lambda

`\SetCell`命令也可以使用可选参数设置当前单元格需要合并的列数和行数。其键和键值的有效值见表 2.4.

```
\begin{tblr}{|X|X|X|X|X|X|}
\hline
Alpha & Beta & Gamma & Delta & Epsilon & Zeta \\
\hline
\SetCell[c=2]{c} Eta & 2-2
& \SetCell[c=2]{c} Iota & 2-4
& \SetCell[c=2]{c} Lambda & 2-6 \\
\hline
\SetCell[c=3]{c} Nu & 3-2 & 3-3
& \SetCell[c=3]{c} Pi & 3-5 & 3-6 \\
\hline
\SetCell[c=6]{c} Tau & 4-2 & 4-3 & 4-4 & 4-5 & 4-6 \\
\hline
\end{tblr}
```

Alpha	Beta	Gamma	Delta	Epsilon	Zeta
Eta		Iota		Lambda	
Nu			Pi		
Tau					

```
\begin{tblr}{|X|X|X|X|X|X|}
\hline
Alpha & Beta & Gamma & Delta & Epsilon & Zeta \\
\hline
\SetCell[r=2]{m} Eta
& Theta & Iota & Kappa & Lambda & \SetCell[r=2]{m} Mu \\
\hline
Nu & Xi & Omicron & Pi & Rho & Sigma \\
\hline
\end{tblr}
```

Alpha	Beta	Gamma	Delta	Epsilon	Zeta
Eta	Theta	Iota	Kappa	Lambda	Mu
	Xi	Omicron	Pi	Rho	

本质上，在第*i*行、第*j*列的单元格前使用的表格命令 `\SetCell[]{<styles>}` 与表格选项 `cell{i}{j}={}{<styles>}` 的作用完全相同。

同样，表格命令 `\SetCells[]{<styles>}` 与表格选项 `cells={}{<styles>}` 的作用完全相同。

2.4 行 (rows) 和列 (columns) 选项

行 (rows) 和列 (columns) 选项的键及键值的有效值见表 2.5 和表 2.6。

表 2.5: rows 选项的键和键值

键	说明与可选键值	初始值
<u>halign</u>	水平对齐方式: l (left)、c (center) 或 r (right)	l
<u>valign</u>	垂直对齐方式: t (top)、m (middle)、b (bottom)、h (head) 或 f (foot)	t
<u>ht</u>	行高	×
<u>bg</u>	背景颜色	×
<u>fg</u>	前景颜色	×
<u>font</u>	字体命令	×
<u>abovesep</u>	行前垂直间距	2pt
<u>abovesep+</u>	行前垂直间距增量	×
<u>belowsep</u>	行后垂直间距	2pt
<u>belowsep+</u>	行后垂直间距增量	×
<u>rowsep</u>	行前行后垂直间距	2pt
<u>rowsep+</u>	行前行后垂直间距增量	×
<u>preto</u>	单元格前导文本 (如 rowspec 选项中的 >)	×
<u>appto</u>	单元格附加文本 (如 rowspec 选项中的 <)	×
<u>cmd</u>	对单元格文本要执行的命令	×

注意: 多数情况下, 对于带下划线的键, 可以省略键名而只给出键值。

表 2.6: columns 选项的键和键值

键	说明与可选键值	初始值
<u>halign</u>	水平对齐方式: l (left)、c (center) 或 r (right)	l
<u>valign</u>	垂直对齐方式: t (top)、m (middle)、b (bottom)、h (head) 或 f (foot)	t
<u>wd</u>	列宽	×
<u>co</u>	可扩展列的扩展系数 (X 列)	×
<u>bg</u>	背景颜色	×

续下页

表 2.6: columns 选项的键和键值 (接前页)

键	说明与可选键值	初始值
fg	前景颜色	×
font	字体命令	×
leftsep	列左侧水平间距	6pt
leftsep+	列左侧水平间距增量	×
rightsep	列右侧水平间距	6pt
rightsep+	列右侧水平间距增量	×
rowsep	列左右水平间距	6pt
rowsep+	列左右水平间距增量	×
preto	单元格前导文本 (如 colspec 选项中的 >)	×
appto	单元格附加文本 (如 colspec 选项中的 <)	×
cmd	对单元格文本要执行的命令	×

注意: 多数情况下, 对于带下划线的键, 可以省略键名而只给出键值。

2.4.1 行和列设置新接口

rows 和 columns 选项分别用于设置表格的所有行/列格式。

```
\begin{tblr}{
  hlines, vlines,
  rows = {7mm}, columns = {15mm,c},
}
Alpha & Beta & Gamma & Delta \\
Epsilon & Zeta & Eta & Theta \\
Iota & Kappa & Lambda & Mu \\
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

row{i} 和 column{j} 选项分别用于设置表格中指定行/列格式。

```
\begin{tblr}{
  hlines = {1pt,white},
  row{odd} = {blue7},
  row{even} = {azure7},
  column{1} = {purple7,c},
}
Alpha & Beta & Gamma & Delta \\
Epsilon & Zeta & Eta & Theta \\
Iota & Kappa & Lambda & Mu \\
Nu & Xi & Omicron & Pi \\
Rho & Sigma & Tau & Upsilon \\
Phi & Chi & Psi & Omega \\
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu
Nu	Xi	Omicron	Pi
Rho	Sigma	Tau	Upsilon
Phi	Chi	Psi	Omega

下面的示例演示了 `bg`、`fg` 和 `font` 键的基本用法。

```
\begin{tblr}{
  row{odd} = {bg=azure8},
  row{1}   = {bg=azure3, fg=white, font=\sffamily},
}
Alpha & Beta & Gamma \\
Delta & Epsilon & Zeta \\
Eta & Theta & Iota \\
Kappa & Lambda & Mu \\
Nu Xi Omicron & Pi Rho Sigma & Tau Upsilon Phi \\
\end{tblr}
```

Alpha	Beta	Gamma
Delta	Epsilon	Zeta
Eta	Theta	Iota
Kappa	Lambda	Mu
Nu Xi Omicron	Pi Rho Sigma	Tau Upsilon Phi

下面的示例演示了 `abovesep`、`belowsep`、`leftsep`、`rightsep` 键的用法。

```
\begin{tblr}{
  hlines, vlines,
  rows = {abovesep=1pt,belowsep=5pt},
  columns = {leftsep=1pt,rightsep=5pt},
}
Alpha & Beta & Gamma & Delta \\
Epsilon & Zeta & Eta & Theta \\
Iota & Kappa & Lambda & Mu \\
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

下面的示例说明了可以用 `belowsep+` 键替代 `\\[dimen]` 命令。

```
\begin{tblr}{
  hlines, row{2} = {belowsep+=5pt},
}
Alpha & Beta & Gamma & Delta \\
Epsilon & Zeta & Eta & Theta \\
Iota & Kappa & Lambda & Mu \\
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

2.4.2 行和列设置旧接口

可以通过 `\SetRow` 命令的必选参数设置当前行的格式。其有效参数见表 2.5。

```

\begin{tblr}{l1l1l}
\hline[1pt]
\SetRow{azure8} Alpha & Beta & Gamma & Delta \\
\hline
\SetRow{blue8,c} Epsilon & Zeta & Eta & Theta \\
\hline
\SetRow{violet8} Iota & Kappa & Lambda & Mu \\
\hline[1pt]
\end{tblr}

```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

本质上，在第*i*行开始的表格命令`\SetRow{<styles>}`与表格选项`row{i}={<styles>}`的作用相同。

另外，在一些行开始的表格命令`\SetRows{<styles>}`与表格选项`rows={<styles>}`的作用相同。

`\SetColumn` 和 `\SetColumns`表格命令的用法分别与 `\SetRow` 和 `\SetRows`命令类似。但一般不直接使用这两个命令。

2.5 colspec 和 rowspec 选项

`colspec/rowspec` 选项用于使用 `column/row` 格式参数设置 `column/row` 的格式。

2.5.1 colspec 选项和 width 选项

`width` 选项用于设置具备可扩展列的表格的宽度。下面的示例演示了 `width` 选项的用法。

```

\begin{tblr}{width=0.8\textwidth, colspec={|l|X[2]|X[3]|X[-1]|}}
Alpha & & Beta & & Gamma & & Delta \\
Epsilon & & Zeta & & Eta & & Theta \\
Iota & & Kappa & & Lambda & & Mu \\
\end{tblr}

```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

2.5.2 列格式

`Tabularray` 宏包仅设计了一个Q列格式的元素格式。其它的列格式都是通过为Q元素格式指定不同的参数实现定义的。


```

\NewColumnType{l}{Q[l]}
\NewColumnType{c}{Q[c]}
\NewColumnType{r}{Q[r]}
\NewColumnType{t}[1]{Q[t,wd=#1]}
\NewColumnType{m}[1]{Q[m,wd=#1]}
\NewColumnType{b}[1]{Q[b,wd=#1]}
\NewColumnType{h}[1]{Q[h,wd=#1]}
\NewColumnType{f}[1]{Q[f,wd=#1]}
\NewColumnType{X}[1]{}{Q[co=1,#1]}

```

```

\begin{tblr}{|t{15mm}|m{15mm}|b{20mm}|}
Alpha & Beta & {Gamma\\Gamma} \\
Epsilon & Zeta & {Eta\\Eta} \\
Iota & Kappa & {Lambda\\Lambda} \\
\end{tblr}

```

		Gamma
Alpha	Beta	Gamma
		Eta
Epsilon	Zeta	Eta
		Lambda
Iota	Kappa	Lambda

任何新的列格式都需要使用 `\NewColumnType` 命令进行定义。在定义时，可以使用可选参数。

2.5.3 行格式

同样，`Tabularray` 宏包仅设计了一个Q行格式的元素。其他的行格式都是通过为Q元素格式指定不同的参数实现定义的。

```

\NewRowType{l}{Q[l]}
\NewRowType{c}{Q[c]}
\NewRowType{r}{Q[r]}
\NewRowType{t}[1]{Q[t,ht=#1]}
\NewRowType{m}[1]{Q[m,ht=#1]}
\NewRowType{b}[1]{Q[b,ht=#1]}
\NewRowType{h}[1]{Q[h,ht=#1]}
\NewRowType{f}[1]{Q[f,ht=#1]}

```

```

\begin{tblr}{rowspec={|t{12mm}|m{10mm}|b{10mm}|}}
Alpha & Beta & {Gamma\\Gamma} \\
Epsilon & Zeta & {Eta\\Eta} \\
Iota & Kappa & {Lambda\\Lambda} \\
\end{tblr}

```

Alpha	Beta	Gamma
		Gamma
		Eta
Epsilon	Zeta	Eta
		Lambda
Iota	Kappa	Lambda

任何新的行格式都需要使用 `\NewRowType` 命令进行定义。在定义时，可以使用可选参数。

第三章 附加接口

3.1 表格参数

表格全局设置的键与键值详见表3.1。

表 3.1: 表格全局设置的键与键值

键	说明与可选键值	初始值
rulesep	两条表格横线或竖线间的间距	2pt
stretch	单元格文本行距拉伸系数	1
abovesep	每行前的垂直间距	2pt
belowsep	每行后的垂直间距	2pt
rowsep	每行前后的垂直间距	2pt
leftsep	每列左边的水平间距	6pt
rightsep	每列右边的水平间距	6pt
colsep	每列左右的水平间距	6pt
hspan	单元格水平合并算法: default、even 或 minimal	default
vspan	单元格垂直合并算法: default 或 even	default

下面的示例演示了使用 `rulesep` 键替代 `\doublerulesep` 参数的方法。

```
\begin{tblr}{  
  colspec={|l|l|l|l|},rowsec={|QQQ|},  
  rulesep=4pt,  
}  
Alpha & Beta & Gamma & Delta \\  
Epsilon & Zeta & Eta & Theta \\  
Iota & Kappa & Lambda & Mu \\  
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

下面的示例演示了使用 `stretch` 键替代 `\arraystretch` 参数的方法。

```
\begin{tblr}{hlines,stretch=1.5}
Alpha & Beta & Gamma & Delta \\
Epsilon & Zeta & Eta & Theta \\
Iota & Kappa & Lambda & Mu \\
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

下面的示例演示了使用 `rowsep` 和 `colsep` 键设置行和列间距设置。

```
\SetTblrInner{rowsep=2pt,colsep=2pt}
\begin{tblr}{hlines,vlines}
Alpha & Beta & Gamma & Delta \\
Epsilon & Zeta & Eta & Theta \\
Iota & Kappa & Lambda & Mu \\
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

使用 `hspan=default` 或 `hspan=even` 时, `Tabularray` 宏包会使用合并宽度计算列宽度。但是, 使用 `hspan=minimal` 时, 则使用列宽度计算合并宽度。下面的示例演示了使用 `hspan` 键的设置结果。

```
\SetTblrInner{hlines, vlines, hspan=default}
\begin{tblr}{cell{2}{1}={c=2}{1},cell{3}{1}={c=3}{1},cell{4}{2}={c=2}{1}}
111 111 & 222 222 & 333 333 \\
12 Multi Columns Multi Columns 12 & & 333 \\
13 Multi Columns Multi Columns Multi Columns 13 & & \\
111 & 23 Multi Columns Multi Columns 23 & \\
\end{tblr}
```

111 111	222 222	333 333
12 Multi Columns Multi Columns 12	333	
13 Multi Columns Multi Columns Multi Columns 13		
111	23 Multi Columns Multi Columns 23	

```
\SetTblrInner{hlines, vlines, hspan=even}
\begin{tblr}{cell{2}{1}={c=2}{1},cell{3}{1}={c=3}{1},cell{4}{2}={c=2}{1}}
111 111 & 222 222 & 333 333 \\
12 Multi Columns Multi Columns 12 & & 333 \\
13 Multi Columns Multi Columns Multi Columns 13 & & \\
111 & 23 Multi Columns Multi Columns 23 & \\
\end{tblr}
```

111 111	222 222	333 333
12 Multi Columns Multi Columns 12	333	
13 Multi Columns Multi Columns Multi Columns 13		
111	23 Multi Columns Multi Columns 23	

```
\SetTblrInner{hlines, vlines, hspan=minimal}
\begin{tblr}{cell{2}{1}={c=2}{1},cell{3}{1}={c=3}{1},cell{4}{2}={c=2}{1}}
  111 111 & 222 222 & 333 333 \\
  12 Multi Columns Multi Columns 12 & & 333 \\
  13 Multi Columns Multi Columns Multi Columns 13 & & \\
  111 & 23 Multi Columns Multi Columns 23 & \\
\end{tblr}
```

111 111	222 222	333 333
12 Multi Columns Multi Columns 12	333	
13 Multi Columns Multi Columns Multi Columns 13		
111	23 Multi Columns Multi Columns 23	

下面的示例演示了使用 `vspan` 键的设置结果。

```
\SetTblrInner{hlines, vlines, vspan=default}
\begin{tblr}{column{2}={3.25cm}, cell{2}{2}={r=3}{1}}
  Column1 & Column2 \\
  Row1 & Long text that needs multiple lines.  

            Long text that needs multiple lines.  

            Long text that needs multiple lines. \\
  Row2 & \\
  Row3 & \\
  Row4 & Short text \\
\end{tblr}
```

Column1	Column2
Row1	Long text that needs multiple lines. Long text that needs multiple lines. Long text that needs multiple lines.
Row2	
Row3	
Row4	Short text

```

\SetTblrInner{hlines, vlines, vspan=even}
\begin{tblr}{column{2}={3.25cm}, cell{2}{2}={r=3}{1}}
  Column1 & Column2 \\
  Row1 & Long text that needs multiple lines.
         Long text that needs multiple lines.
         Long text that needs multiple lines. \\
  Row2 & \\
  Row3 & \\
  Row4 & Short text \\
\end{tblr}

```

Column1	Column2
Row1	Long text that needs multiple lines. Long text that needs multiple lines. Long text that needs multiple lines.
Row2	
Row3	
Row4	Short text

3.2 参数默认值

Tabularray 宏包提供了 `\SetTblrInner` 和 `\SetTblrOuter` 两个命令，这两个命令用于设置表格的内部和外部参数的默认值。内部参数是指在 `tblr` 环境的必选参数，而外部参数是指在 `tblr` 环境的可选参数。目前，本宏包的外部参数主要用于长表格的参数设置（参见第四章）。

下面的示例中，第一行代码用于设置此后所有表格都绘制表格横线和竖线，第二行代码用于设置此后所有表格的垂直对齐方式为底端基线对齐。

```

\SetTblrInner{hlines,vlines}
\SetTblrOuter{valign=b}

```

可以使用 `\NewTblrEnviron` 命令定义新的 Tabularray 环境：

```

\NewTblrEnviron{mytblr}
\SetTblrInner[mytblr]{hlines,vlines}
\SetTblrOuter[mytblr]{valign=b}
Text \begin{mytblr}{cccc}
  Alpha & Beta & Gamma & Delta \\
  Epsilon & Zeta & Eta & Theta \\
  Iota & Kappa & Lambda & Mu \\
\end{mytblr} Text

```

	Alpha	Beta	Gamma	Delta	
	Epsilon	Zeta	Eta	Theta	
Text	Iota	Kappa	Lambda	Mu	Text

如果在 `\SetTblrInner` 或 `\SetTblrOuter` 命令中未指定可选参数，则是为 `tblr` 环境指定默认参数。因此，不同的 Tabularray 环境可以具有不同的默认参数。

3.3 定义表格新命令

必须使用 `\NewTableCommand` 命令定义所有用于改变表格参数的命令。下面的示例演示了如何定义一个新的表格命令：

```
\NewTableCommand\myhline{\hline[0.1em,red5]}
\begin{tblr}{llll}
\myhline
Alpha & Beta & Gamma & Delta \\
Epsilon & Zeta & Eta & Theta \\
Iota & Kappa & Lambda & Mu \\
\myhline
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu

3.4 表格内容宏的提前展开

`Tabularray` 宏包使用 `l3regex` 解析表格内容，因此，需要知道每一个 `&` 和 `\\` 符号。所以，不能将表格内容放在任何由 `\NewTableCommand` 命令定义的表格命令中。但是，可以使用 `expand` 选项为 `Tabularray` 宏包指定在解析表格内容之前 一次 性展开包含表格内容的宏。

```
\def\tblrbody{
\hline
20 & 30 & 40 \\
50 & 60 & 70 \\
\hline
}
\begin{tblr}[expand=\tblrbody]{ccc}
\hline
AA & BB & CC \\
\tblrbody
DD & EE & FF \\
\tblrbody
GG & HH & II \\
\hline
\end{tblr}
```

AA	BB	CC
20	30	40
50	60	70
DD	EE	FF
20	30	40
50	60	70
GG	HH	II

通过 `expand` 选项，可以利用 `environ` 宏包定义基于 `tblr` 环境的新环境：

```
\NewEnviron{fancytblr}{
Before Text
\begin{tblr}[expand=\BODY]{hlines}
\BODY
\end{tblr}
After Text
}
\begin{fancytblr}
One & Two & Three \\
Four & Five & Six \\
Seven & Eight & Nine \\
\end{fancytblr}
```

	One	Two	Three	
Before Text	Four	Five	Six	After Text
	Seven	Eight	Nine	

3.5 原文排版命令

通过 `verb` 参数，可以在单元格文本中使用 `\verb` 原文排版命令：

```
\begin{tblr}{hlines,verb}
  20 & 30 & \verb!\hello{world}!40 \\
  50 & \verb!\hello!60 & 70 \\
\end{tblr}
```

20	30	\hello{world}40
50	\hello60	70

3.6 计数器和长度

可以在单元格文本中使用 `rownum`、`colnum`、`rowcount` 和 `colcount` 计数器：

```
\begin{tblr}{hlines}
Cell[\arabic{rownum}][\arabic{colnum}] &
Cell[\arabic{rownum}][\arabic{colnum}] &
Cell[\arabic{rownum}][\arabic{colnum}] &
Cell[\arabic{rownum}][\arabic{colnum}] \\
Row=\arabic{rowcount}, Col=\arabic{colcount} &
Row=\arabic{rowcount}, Col=\arabic{colcount} &
Row=\arabic{rowcount}, Col=\arabic{colcount} &
Row=\arabic{rowcount}, Col=\arabic{colcount} \\
Cell[\arabic{rownum}][\arabic{colnum}] &
Cell[\arabic{rownum}][\arabic{colnum}] &
Cell[\arabic{rownum}][\arabic{colnum}] &
Cell[\arabic{rownum}][\arabic{colnum}] \\
\end{tblr}
```

Cell[1][1]	Cell[1][2]	Cell[1][3]	Cell[1][4]
Row=3, Col=4	Row=3, Col=4	Row=3, Col=4	Row=3, Col=4
Cell[3][1]	Cell[3][2]	Cell[3][3]	Cell[3][4]

也可以在单元格文本中也可以使用 `\leftsep`、`\rightsep`、`\abovesep` 和 `\belowsep` 长度。

3.7 跟踪 Tabularray

可使用 `\SetTblrTracing` 命令跟踪 `tblr` 的内部数据流。例如，`\SetTblrTracing{all}` 用于打开所有跟踪，`\SetTblrTracing{none}` 用于关闭所有跟踪。而 `\SetTblrTracing{+row,+column}` 仅跟踪指定的行和列的数据流。所有的跟踪结果都将写入 `log` 文件。

第四章 长表格

本章中说明的接口是 **实验性的** 并且在未来可能会发生变化，因此，请勿在重要的文档中使用它们。

4.1 简单示例

在排版有表头和表尾的长表格时，最好将表头/表尾分开设计为 题注/尾注(包括标题、表注、说明、续表文本) 和 标题行/尾行(每一页都重复出现的行)。例如，通过这种办法，交替对各行使用有不同颜色，就可以实现“斑马色”表格。

表 4.1: 一个长长长长长长长长的表格

Head	Head	Head
Head	Head	Head
Alpha	Beta	Gamma
Epsilon	Zeta ^a	Eta
Iota	Kappa [†]	Lambda
Nu	Xi	Omicron
Rho	Sigma	Tau
Phi	Chi	Psi
Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda
Nu	Xi	Omicron
Rho	Sigma	Tau
Phi	Chi	Psi
Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Foot	Foot	Foot

续下页

表 4.1: 一个长长长长长长长长的表格 (接前页)

Head	Head	Head
Head	Head	Head
Iota	Kappa	Lambda
Nu	Xi	Omicron
Rho	Sigma	Tau
Phi	Chi	Psi
Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda
Nu	Xi	Omicron
Rho	Sigma	Tau
Phi	Chi	Psi
Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda
Nu	Xi	Omicron
Rho	Sigma	Tau
Phi	Chi	Psi
Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda
Nu	Xi	Omicron
Rho	Sigma	Tau
Phi	Chi	Psi
Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda
Nu	Xi	Omicron
Rho	Sigma	Tau
Phi	Chi	Psi
Alpha	Beta	Gamma
Foot	Foot	Foot

续下页

表 4.1: 一个长长长长长长长长的表格 (接前页)

Head	Head	Head
Head	Head	Head
Epsilon	Zeta	Eta
Iota	Kappa	Lambda
Nu	Xi	Omicron
Rho	Sigma	Tau
Phi	Chi	Psi
Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda
Nu	Xi	Omicron
Rho	Sigma	Tau
Phi	Chi	Psi
Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda
Nu	Xi	Omicron
Rho	Sigma	Tau
Phi	Chi	Psi
Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda
Nu	Xi	Omicron
Rho	Sigma	Tau
Phi	Chi	Psi
Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda
Nu	Xi	Omicron
Rho	Sigma	Tau
Phi	Chi	Psi
Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda
Nu	Xi	Omicron
Rho	Sigma	Tau
Phi	Chi	Psi
Foot	Foot	Foot

续下页

表 4.1: 一个长长长长长长长长的表格 (接前页)

Head	Head	Head
Head	Head	Head
Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda
Nu	Xi	Omicron
Rho	Sigma	Tau
Phi	Chi	Psi
Alpha	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda
Nu	Xi	Omicron
Rho	Sigma	Tau
Phi	Chi	Psi
Foot	Foot	Foot

^a 第一个表注。

[†] 第二个长长长长长长的表注。

注意: 一些常规说明, 一些常规说明, 一些常规说明。

来源: 自力更生, 自力更生, 自力更生。

显然, `Tabularray`宏包中的长表格与`threeparttable`和`threeparttablex`宏包的表格类似, 并且`Tabularray`支持表格尾注, 但表格尾注不在页脚排版。

上述表格的 L^AT_EX 源代码如下，该源码具备足够的自明性。

```

\NewTblrTheme{fancy}{
  \SetTblrStyle{firsthead}{font=\bfseries}
  \SetTblrStyle{firstfoot}{fg=blue2}
  \SetTblrStyle{middlefoot}{\itshape}
  \SetTblrStyle{caption-tag}{red2}
}
\begin{longtblr}[
  theme = fancy,
  caption = {一个长长长长长长长长的表格},
  entry = {短标题},
  label = {tblr:test},
  note{a} = {第一个表注。},
  note{${\dag$}} = {第二个长长长长长长的表注。},
  remark{注意} = {一些常规说明，一些常规说明，一些常规说明。},
  remark{来源} = {自力更生，自力更生，自力更生。},
]{
  colspec = {XXX}, width = 0.85\linewidth,
  rowhead = 2, rowfoot = 1,
  row{odd} = {gray9}, row{even} = {brown9},
  row{1-2} = {purple7}, row{Z} = {blue7},
}
\hline
Head & Head & Head & \\\
\hline
Head & Head & Head & \\\
\hline
Alpha & Beta & Gamma & \\\
\hline
Epsilon & Zeta\TblrNote{a} & & Eta & \\\
\hline
Iota & Kappa\TblrNote{${\dag$}} & & Lambda & \\\
\hline
Nu & Xi & Omicron & \\\
\hline
Rho & Sigma & Tau & \\\
\hline
.....
\hline
Nu & Xi & Omicron & \\\
\hline
Rho & Sigma & Tau & \\\
\hline
Phi & Chi & Psi & \\\
\hline
Foot & Foot & Foot & \\\
\hline
\end{longtblr}

```

在 Tabularray 宏包中，longtblr 环境用于排版长表格，并完全实现了表格样式与内容的分离。

标题行和尾行是出现在每页的表格行，它们由表格中指定的行构成。需要通过 longtblr 环境的必选参数中的内部参数指定标题行和尾行。例如，上述代码中，分别通过 rowhead=2 和 rowfoot=1 指定了标题行和尾行。

表 4.2: 标题行和尾行内部参数

键	含义	初始值
rowhead	每页要显示的标题行行数，从表格的起始行开始向后计数	0
rowfoot	每页要显示的尾行行数，从表格最后一行开始向前计数	0

表格题注和尾注由标题、表注、说明和续表文本构成。需要通过 `longtblr` 环境的可选参数中的外部参数指定题注和尾注。

表 4.3: 表格题注和尾注外部参数

键	含义	初始值
headsep	表格题注与表格之间的垂直间距	6pt
footsep	表格尾注与表格之间的垂直间距	6pt
presep	表格题注与其之前文本之间的垂直间距	<code>1.5\bigskipamount</code>
postsep	表格尾注与其之后文本之间的垂直间距	<code>1.5\bigskipamount</code>
theme	表格主题 (包括模板及样式设置)	×
caption	表格标题	×
entry	用于目录的表格短标题	×
label	表格标签	×
<code>note{<name>}</code>	表格表注，其中 <code><name></code> 是表注标签	×
<code>remark{<name>}</code>	表格说明，其中 <code><name></code> 是说明标签	×

如果使用了 `entry=none`，则不会表格目录中添加任何条目。因此，`caption=text,entry=none` 与 `longtable` 中的 `\caption[] {text}` 的功能类似。

如使用了 `label=none`，`Tabularray` 宏包的 `table` 计数器将不自增，并将 `caption-tag` 和 `caption-sep` 模板元素置空 (见后续示例)。因此，`caption=text,entry=none,label=none` 除了计数器处理外，与 `longtable` 宏包的 `\caption* {text}` 功能类似。

4.2 个性化模板

4.2.1 模板概述

`Tabularray`的题注和尾注模板系统的设计主要受`beamer`、`caption` 和 `longtable` 宏包的启发。可以使用 `\DefTblrTemplate` 命令定义或修改一个模板，用 `\SetTblrTemplate` 命令选择默认模板。在定义模板时，可以用 `\UseTblrTemplate` 和 `\ExpTblrTemplate` 命令引入其它模板。

表 4.4: 题注和尾注模板元素

元素名称	元素说明和默认模板
<code>contfoot-text</code>	表格在每页尾部的续表文本, 一般是“Continued on next page”
<code>contfoot</code>	表格在每页尾部的续表段落, 一般包括 <code>contfoot-text</code> 模板
<code>conthead-text</code>	表格在每页标题中的续表文本, 一般是“(Continued)”
<code>conthead</code>	表格在每页标题中的续表段落, 一般包括 <code>conthead-text</code> 模板
<code>caption-tag</code>	标题标签, 一般类似“Table 4.2”
<code>caption-sep</code>	标题分隔符, 一般类似“: ”
<code>caption-text</code>	标题文本, 一般由用户提供内容
<code>caption</code>	包括 <code>caption-tag</code> + <code>caption-sep</code> + <code>caption-text</code> 的组合
<code>note-tag</code>	尾注标签, 一般由用户提供
<code>note-sep</code>	尾注分隔符, 一般类似“ ”
<code>note-text</code>	尾注内容, 一般由用户提供
<code>note</code>	包括 <code>note-tag</code> + <code>note-sep</code> + <code>note-text</code> 的组合
<code>remark-tag</code>	注释标签, 一般由用户提供
<code>remark-sep</code>	注释分隔符, 一般类似“: ”
<code>remark-text</code>	注释文本, 一般由用户提供
<code>remark</code>	包括 <code>remark-tag</code> + <code>remark-sep</code> + <code>remark-text</code> 的组合
<code>firsthead</code>	第一页的表头, 一般包括 <code>caption</code> 模板
<code>middlehead</code>	中间页的表头, 一般包括 <code>caption</code> 和 <code>conthead</code> 模板
<code>lasthead</code>	最后一页的表头, 一般包括 <code>caption</code> 和 <code>conthead</code> 模板
<code>head</code>	<code>firsthead</code> 、 <code>middlehead</code> 和 <code>lasthead</code> 的所有设置
<code>firstfoot</code>	第一页的表尾, 一般包括 <code>contfoot</code> 模板
<code>middlefoot</code>	中间页的表尾, 一般包括 <code>contfoot</code> 模板
<code>lastfoot</code>	最后一页的表尾, 一般包括 <code>note</code> 和 <code>remark</code> 模板
<code>foot</code>	<code>firstfoot</code> 、 <code>middlefoot</code> 和 <code>lastfoot</code> 的所有设置

仅包含短文本的元素称为子元素。一般在子元素的名称中有一个-符号。包含一个或多个段落的元素称为主元素。通常, 在主元素的名称中不包含-符号。

除了上述模板, `Tabularray` 预定义了 `normal` 和 `empty` 两个模板。可以使用 `\SetTblrTemplate` 命令进行选择。

4.2.2 续表模板

首先是续表文本的模板定义:

```
\DefTblrTemplate{contfoot-text}{normal}{Continued on next page}
\SetTblrTemplate{contfoot-text}{normal}
\DefTblrTemplate{conthead-text}{normal}{(Continued)}
\SetTblrTemplate{conthead-text}{normal}
```

在以上代码中, `\DefTblrTemplate` 命令定义了名为 `normal` 的模板, 然后用 `\SetTblrTemplate` 命令将名为 `normal` 的模板设置为默认模板。 `normal` 模板总是被定义的, 并且会被 `Tabularray` 设置为任何一个元素的默认模板。因此, 在定义一个新模板时, 需要将其命名为其它名称。

如果在 `\DefTblrTemplate` 命令中使用 `default` 作为模板名称, 则会在定义时同时将其设置为默认模板。因此, 上述代码也可以修改为:

```
\DefTblrTemplate{contfoot-text}{default}{Continued on next page}
\DefTblrTemplate{conthead-text}{default}{(Continued)}
```

可以通过修改这些代码以适应自己的需求。

`contfoot` 和 `conthead` 模板一般使用 `\UseTblrTemplate` 命令包含它们子元素的模板。但是, 可以使用诸如水平对齐等设置参数。

```
\DefTblrTemplate{contfoot}{default}{\UseTblrTemplate{contfoot-text}{default}}
\DefTblrTemplate{conthead}{default}{\UseTblrTemplate{conthead-text}{default}}
```

4.2.3 标题模板

通常, 标题由三部分构成, 可以用如下代码定义其模板:

```
\DefTblrTemplate{caption-tag}{default}{Table\hspace{0.25em}\thetable}
\DefTblrTemplate{caption-sep}{default}{:\enskip}
\DefTblrTemplate{caption-text}{default}{\InsertTblrText{caption}}
```

`\InsertTblrText{caption}` 命令用于为 `caption` 键插入标题内容, 在 `longtblr` 环境的可选参数中, 也可以写入标题内容。

`caption` 模板通常用 `\UseTblrTemplate` 插入三个子模板: `caption` 模板将被用于 `firsthead` 模板。

```
\DefTblrTemplate{caption}{default}{
  \UseTblrTemplate{caption-tag}{default}
  \UseTblrTemplate{caption-sep}{default}
  \UseTblrTemplate{caption-text}{default}
}
```

当然, `capcont` 模板也包含 `conthead` 模板。 `capcont` 模板也被用于 `middlehead` 和 `lasthead` 模板。

```
\DefTblrTemplate{capcont}{default}{
  \UseTblrTemplate{caption-tag}{default}
  \UseTblrTemplate{caption-sep}{default}
  \UseTblrTemplate{caption-text}{default}
  \UseTblrTemplate{conthead}{default}
}
```


4.2.4 表注和说明模板

表注模板可以按如下方式定义：

```
\DefTblrTemplate{note-tag}{default}{\textsuperscript{\InsertTblrNoteTag}}
\DefTblrTemplate{note-sep}{default}{\space}
\DefTblrTemplate{note-text}{default}{\InsertTblrNoteText}
```

```
\DefTblrTemplate{note}{default}{
  \MapTblrNotes{
    \noindent
    \UseTblrTemplate{note-tag}{default}
    \UseTblrTemplate{note-sep}{default}
    \UseTblrTemplate{note-text}{default}
  }
}
```

`\MapTblrNotes` 命令用于遍历所有表格的表注，这些表注在 `longtblr` 环境的可选参数中设置。在遍历中，可以分别通过 `\InsertTblrNoteTag` 命令 `\InsertTblrNoteText` 插入当前标签和表注文本。

注释模板的定义与表注模板的定义类似。

```
\DefTblrTemplate{remark-tag}{default}{\InsertTblrRemarkTag}
\DefTblrTemplate{remark-sep}{default}{:\space}
\DefTblrTemplate{remark-text}{default}{\InsertTblrRemarkText}
```

```
\DefTblrTemplate{remark}{default}{
  \MapTblrRemarks{
    \noindent
    \UseTblrTemplate{remark-tag}{default}
    \UseTblrTemplate{remark-sep}{default}
    \UseTblrTemplate{remark-text}{default}
  }
}
```

4.2.5 表头和表尾模板

表格的表头和表尾模板被定义为包含其它模板：

```

\DefTblrTemplate{firsthead}{default}{
  \UseTblrTemplate{caption}{default}
}
\DefTblrTemplate{middlehead,lasthead}{default}{
  \UseTblrTemplate{capcont}{default}
}
\DefTblrTemplate{firstfoot,middlefoot}{default}{
  \UseTblrTemplate{contfoot}{default}
}
\DefTblrTemplate{lastfoot}{default}{
  \UseTblrTemplate{note}{default}
  \UseTblrTemplate{remark}{default}
}

```

注意，可以在 `\DefTblrTemplate` 命令中为多个元素定义同一个模板。

4.3 改变样式

模板元素的有效设置详见表 4.5.

表 4.5: 元素的样式

键	含义
<code>fg</code>	前景颜色
<code>font</code>	字体命令
<code>halign</code>	水平对齐方式: l (left)、c (center) 或 r (right)
<code>indent</code>	段落缩进值
<code>hang</code>	悬挂缩进值

注意：多数情况下，可以省略带下划线的键名而仅给出键值。`halign`、`indent` 和 `hang` 仅对主模板有效。

可以使用 `\SetTblrStyle` 命令改变元素的样式：

```

\SetTblrStyle{firsthead}{font=\bfseries}
\SetTblrStyle{firstfoot}{fg=blue2}
\SetTblrStyle{middlefoot}{\itshape}
\SetTblrStyle{caption-tag}{red2}

```

在模板定义中，当使用 `\UseTblrTemplate{element}{default}` 时，除了包含模板 `element` 代码外，会自动设置模板 `element` 的前景颜色和字体命令。相反，`\ExpTblrTemplate{element}{default}` 将仅包含模板代码。

4.4 定义主题

可以使用 `\NewTblrTheme` 命令定义表头与表尾的样式主题。一个主题由模板和样式设置组成，例如：

```
\NewTblrTheme{fancy}{
  \DefTblrTemplate{conthead}{default}{[[Continued]]}
  \SetTblrStyle{firsthead}{font=\bfseries}
  \SetTblrStyle{firstfoot}{fg=blue2}
  \SetTblrStyle{middlefoot}{\itshape}
  \SetTblrStyle{caption-tag}{red2}
}
```

定义了 `fancy` 主题后，在 `longtblr` 环境的可选参数中使用 `theme=fancy` 选项使用该主题。

4.5 分页控制

像 `longtable` 宏包一样，在 `longtblr` 环境中，可以使用 `*` 或 `\nopagebreak` 禁用分页，用 `\pagebreak` 实现强制分页。

4.6 添加表注

`Tabularray` 宏包提供了 `talltblr` 环境，以替代 `threeparttable` 环境。该环境不可跨页，但可以用于 `table` 环境中。

```
TEXT\begin{talltblr}[
  caption = {长长长长长长长的表格},
  entry = {短标题},
  label = {tblr:tall},
  note{a} = {第一个表注。},
  note{${\dag}} = {第二个长长长长长长长的表注。},
]{
  colspec = {XXX}, width = 0.5\linewidth, hlines,
}
Alpha & Beta & Gamma \\
Epsilon & Zeta & Eta\TblrNote{a} \\
Iota & Kappa & Lambda\TblrNote{${\dag}} \\
\end{talltblr}TEXT
```

表 4.6: 长长长长长长长的表格

Alpha	Beta	Gamma
Epsilon	Zeta	Eta ^a
Iota	Kappa	Lambda [†]

^a 第一个表注。

[†] 第二个长长长长长长长的表注。

第五章 使用扩展库

Tabularray 宏包模仿或修改了其它宏包的一些命令, 为避免冲突, 需要使用 `\UseTblrLibrary` 载入这些扩展库。

5.1 booktabs 库

如果使用了 `\UseTblrLibrary{booktabs}`, 则 Tabularray 宏包将定义 `\toprule`、`\midrule`、`\bottomrule` 和 `\cmidrule` 命令, 这些命令可以直接用于 tblr 环境中。

```
\begin{tblr}{l1l1l1}
\toprule
Alpha & Beta & Gamma & Delta \\
\midrule
Epsilon & Zeta & Eta & Theta \\
\cmidrule{1-3}
Iota & Kappa & Lambda & Mu \\
\cmidrule{2-4}
Nu & Xi & Omicron & Pi \\
\bottomrule
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu
Nu	Xi	Omicron	Pi

目前, `\cmidrule` 命令还不支持 `trim` 选项 (一种变通方法是插入一个空列分隔两个 `\cmidrule`)。但是, 可以用类似 `\hline` 和 `\cline` 命令的方式指定线条颜色。

```
\begin{tblr}{l1l1l1}
\toprule[purple3]
Alpha & Beta & Gamma & Delta \\
\midrule[blue3]
Epsilon & Zeta & Eta & Theta \\
\cmidrule[azure3]{1-3}
Iota & Kappa & Lambda & Mu \\
\cmidrule[azure3]{2-4}
Nu & Xi & Omicron & Pi \\
\bottomrule[purple3]
\end{tblr}
```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu
Nu	Xi	Omicron	Pi

5.2 diagbox 库

当在导言区使用了 `\UseTblrLibrary{diagbox}` 后, Tabularray 宏包会载入 `diagbox` 宏包, 然后, 就可以在 `tblr` 环境中使用 `\diagbox` 和 `\diagboxthree` 命令排版斜线表头。

```
\begin{tblr}{hlines,vlines}
\diagbox{Aa}{Pp} & Beta & Gamma \\
Epsilon & Zeta & Eta \\
Iota & Kappa & Lambda \\
\end{tblr}
```

Pp \diagdown Aa	Beta	Gamma
Epsilon	Zeta	Eta
Iota	Kappa	Lambda

```
\begin{tblr}{hlines,vlines}
\diagboxthree{Aa}{Pp}{Hh} & Beta & Gamma \\
Epsilon & Zeta & Eta \\
Iota & Kappa & Lambda \\
\end{tblr}
```

Pp \diagdown Aa	Hh \diagdown Aa	Beta	Gamma
Epsilon	Zeta	Eta	
Iota	Kappa	Lambda	

也可以在数学模式中使用 `\diagbox`和 `\diagboxthree` 命令。

```
$$\begin{tblr}{|c|cc|}
\hline
\diagbox{X_1}{X_2} & 0 & 1 \\
\hline
0 & 0.1 & 0.2 \\
1 & 0.3 & 0.4 \\
\hline
\end{tblr}$$
```

X ₂ \diagdown X ₁	0	1
0	0.1	0.2
1	0.3	0.4

5.3 siunitx 库

当在导言区使用了`\UseTblrLibrary{siunitx}`后, `Tabularray` 宏包会载入 `siunitx` 宏包, 并定义了S列格式, 表示带有si键的Q列格式。

```
\begin{tblr}{
hlines, vlines,
colspec={
S[table-format=3.2]
S[table-format=3.2]
S[table-format=3.2]
}
}
{{{\Head}}} & {{{Head}}} & {{{Head}}} \\
111 & 111 & 111 \\
2.1 & 2.2 & 2.3 \\
33.11 & 33.22 & 33.33 \\
\end{tblr}
```

Head	Head	Head
111	111	111
2.1	2.2	2.3
33.11	33.22	33.33

```

\begin{tblr}{
  hlines, vlines,
  colspec={
    Q[si={table-format=3.2},c]
    Q[si={table-format=3.2},c]
    Q[si={table-format=3.2},c]
  }
}
{{Head}} & {{Head}} & {{Head}} \\
111      & 111      & 111      \\
2.1     & 2.2     & 2.3     \\
33.11   & 33.22   & 33.33   \\
\end{tblr}

```

Head	Head	Head
111	111	111
2.1	2.2	2.3
33.11	33.22	33.33

注意，需要使用 三重 大括号对以确保单元格是非数字模式。

另外，也必须使用 l、c 或 r 设置非数字单元格的水平对齐方式。

```

\begin{tblr}{
  hlines, vlines, columns={6em},
  colspec={
    Q[si={table-format=3.2,table-number-alignment=left},l,blue7]
    Q[si={table-format=3.2,table-number-alignment=center},c,teal7]
    Q[si={table-format=3.2,table-number-alignment=right},r,purple7]
  }
}
{{Head}} & {{Head}} & {{Head}} \\
111      & 111      & 111      \\
2.1     & 2.2     & 2.3     \\
33.11   & 33.22   & 33.33   \\
\end{tblr}

```

Head	Head	Head
111	111	111
2.1	2.2	2.3
33.11	33.22	33.33

此时，S 和 s 列格式都可用。实质上，这两个列格式是按如下方式定义的：

```

\NewColumnType{S}[1] [] {Q[si={##1},c]}
\NewColumnType{s}[1] [] {Q[si={##1},c,cmd=\TblrUnit]}

```

第六章 宏包源码