

tabular2*: 一个实验性的表格排版方案

Ms_yam (Ms_yam@163.com)

二零二五年九月三日

摘要

本宏包是一个实验性的表格排版解决方案。它采用一套全新的表格输入输出接口，使得输入输出的方式多样化且更加可读。代码层上采用 `expl3` 语法，完全摆脱了对传统的底层表格命令的依赖¹。

目录

1	用户文档	3
1.1	数据输入	4
1.1.1	录入环境	4
1.1.2	行列名设置	4
1.1.3	内容录入	4
1.1.4	格式设置	6
1.2	渲染输出	7
1.3	选项参数	8
1.3.1	宏包选项	8
1.3.2	设置命令	9
1.4	调试	9
2	关键逻辑	9
2.1	单元格大小	9
2.1.1	间距	9
2.1.2	宽度	10
2.1.3	高度	10

*原本计划叫 `xtable/xtabular`，但因与其它宏冲突取消。后受 R 的 `ggplot2` 的启发，用于表示这是一个全新的 `tabular`。

¹但仍然使用了部分底层命令，如 `\hspace` 等。如有更好的 `expl3` 实现方法，欢迎指正。

3	基础定义	11
3.1	通用变体	11
3.2	l3draw 函数	11
3.3	消息定义	12
3.4	通用变量	12
3.4.1	全局常量	12
3.4.2	专用变量	12
3.4.3	缓存变量	14
3.4.4	临时变量	14
3.5	通用选项	15
3.5.1	宏包选项	15
3.5.2	全局线样式	16
3.6	通用函数	17
3.6.1	常规设置	17
3.6.2	线样式相关	18
4	表格核心	19
4.1	核心存储	19
4.1.1	行名与列名	19
4.1.2	单元格内容	20
4.1.3	单元格样式	20
4.1.4	合并单元格	21
4.1.5	中间变量	21
4.2	行列定位	22
4.2.1	行列名设置	22
4.2.2	行列名解析	23
4.3	内容处理	24
4.3.1	初始化与保存	24
4.3.2	内容设置	26
4.3.3	内容查询	27
4.4	格式处理	28
4.4.1	样式设置	28
4.4.2	样式查询	31
4.5	调试	32
5	中间计算	33
5.1	列标题	33
5.2	行列尺寸	33
5.2.1	边距设置	33
5.2.2	宽度计算	34
5.2.3	高度计算	37
5.2.4	查询尺寸	40
5.2.5	刷新坐标	41

1	用户文档	3
6	输入接口	42
6.1	变量与选项	42
6.1.1	常量定义	42
6.1.2	状态参数	43
6.1.3	局部变量	44
6.1.4	局部选项	44
6.2	数据解析	44
6.2.1	标准格式	44
6.2.2	CSV 数据	45
6.2.3	JSON 数据	46
6.3	数据录入	50
6.4	格式设置	54
7	渲染输出	55
7.1	通用内容	55
7.1.1	局部变量	55
7.1.2	渲染函数	56
7.2	打印表格	57
7.3	渲染表格	59
7.3.1	渲染内容	59
7.3.2	渲染边框-三线表	61
7.3.3	渲染边框-网格线	62
7.3.4	渲染边框-框架	64
7.3.5	渲染表格	65
	Index	66

1 用户文档

与传统表格不同，本模块将表格的输入与输出拆开，前者使用设置表格数据（内容及内容样式）²，后者确认渲染条件（边框、颜色等）。

为方便定位，本宏包为表格提供两种定位方式：数字坐标与行列名。数字坐标从 1 开始计数（不含表头 Header³）；行列名则可由用户指定的⁴，但不能是以下名称之一：`header`、`title`、`first`、`last`。

²原计划只涉及内容，但由于宽高与对齐影响单元格大小，所以把这两部分放在前面。如有更好的意见，欢迎指点。

³在设计上，表头是与列名绑定的，不能直接修改。如果表格有表头，那么修改列名就会同步修改表头。

⁴当列名与数字坐标冲突时，定位以数字坐标处理。即使第 1 列的列名为 3，3 也始终指向第 3 列。

1.1 数据输入

1.1.1 录入环境

```
\xtable \begin {xtable} [(选项)]
      嵌套的环境与命令...
\end {xtable}
```

一个表格输入环境，它本身不能直接输入，也不会渲染输出表格。它会先初始化一个空表，然后执行嵌套的数据输入环境与命令，最后做一些必要的计算。

TeX 黑客笔记： 所有表格数据，必须通过本环境嵌套的环境与命令输入。

1.1.2 行列名设置

本宏包可以单独设置表格的行名与列名，以便设计表格的结构及方便后续需要坐标的命令与环境定位。

```
\excelcolname \excelcolname [(n)]
```

将前 $\langle n \rangle$ 列的列名设置为 A 、 B 、... (类似 Excel 列名)，默认 $\langle n \rangle = 26$ 。本命令不会影响表格的大小。

```
\rowname \rowname ((i)) [(分隔符)] {(名称列表)}
\colname \colname ((i)) [(分隔符)] {(名称列表)}
```

从第 $\langle i \rangle$ 行/列开始依次设置行名/列名。其中，名称由 $\langle \text{名称列表} \rangle$ 指定，后者使用 $\langle \text{分隔符} \rangle$ 分隔。默认 $\langle i \rangle = 1$ ， $\langle \text{分隔符} \rangle = “,”$ 。本命令会调整表格大小，使其可容纳 $\langle \text{名称列表} \rangle$ 中的全部的行名/列名。

1.1.3 内容录入

```
\data \begin {data} [(选项)]
      A, B, C \\
      1, 2, 3...
\end {data}
```

表格内容录入环境。

TeX 黑客笔记： 本宏包只定义了其在 `xtable` 环境中的作用，对环境外不作干涉。**【下同】**

标准输入以 “\\” 为行分隔符⁵，以指定的分隔符（默认为 “,”）为列分隔符，不做任何其它特殊处理。这种输入方式简单易解码，处理速度快，因此推荐作为主要输入方式。

⁵在标准输入中，如果单元格里也有换行，请使用 `\newline`。

以下是一个类似之前的表格输入方式的示例：

```
\begin{xtable}
% loc 用于指定基准位置, sep 用于指定分隔符
\begin{data}[loc={2,1}, sep={&}]
  Name & Sex & Age\\
  John & man & 20
\end{data}
\end{xtable}
```

Name	Sex	Age
John	man	20

CSV 作为最传统的输入方式，本宏包也添加了支持⁶。由于 T_EX 读取普通显示字符时，换行符会被忽略，因此在输入时也使用 “\\” 作为换行。

此外，本宏包还有限地支持 JSON 数据源，其中最主要的限制就是不允许嵌套。以下是一个 JSON 的输入示例。

```
\begin{xtable}
% format 用于指定输入的格式
\begin{data}[format=json]
[
{
  "Name": " 张三",
  "Age": 30,
  "City": " 北京"
},
{
  "Name": " 李四",
  "Age": 25,
  "City": " 上海"
}
]
\end{data}
\end{xtable}
```

Name	Age	City
张三	30	北京
李四	25	上海

`\file` `\file` [(选项)] {(文件名)}

导入数据文件，规则与 `data` 类似。**【此功能尚未实现】**

T_EX 黑客笔记： 文件中的换行符无需处理，其他待确认。

⁶标准输入其实就是一个特殊的 CSV 格式，但因其不需要考虑特殊字符，处理效率显著强于后者。

```

\row \row (<行坐标>,<列坐标>) [<分隔符>] {<内容>}
\col \col (<行坐标>,<列坐标>) [<分隔符>] {<内容>}
\cell \cell(<行坐标>,<列坐标>) {<内容>}

```

按行/列/单元格的方式设定表格的内容。对于行与列，若坐标缺省，则自动在末尾新增行或列；若只提供一个维度，则另一个维度自动补充为 1。

TeX 黑客笔记： <行坐标>、<列坐标> 坐标支持数据坐标或行名/列名，若行名/列名不存在，则自动添加到行尾/列尾。【下同】

```

\savetable \savetable {<名称>}
\loadtable \loadtable {<名称>}

```

将当前的表格数据保存到指定 <名称> 中，或加载之前保存的名为 <名称> 的表⁷。

1.1.4 格式设置

```

\rowheight \rowheight [<全局样式>] {<样式列表>}
\colwidth \colwidth [<全局样式>] [<总宽度>] {<样式列表>}

```

设置表格的行高与列宽样式，样式支持 `auto`、`same`、`fill`（仅列宽）、`samefill`（仅列宽）及指定数值。<样式列表> 若干个 [<坐标 => 样式] 组成，并以逗号分隔。

如果省略坐标，则坐标为“前一个坐标 +1”。如果是 <样式列表> 第 1 个，则行默认为 `header`⁸（有 Header）或 1（没有 Header）；而列默认为 1。

TeX 黑客笔记： 样式 `auto` 表示自然尺寸；`same` 则表示同组所有行/列的值相同⁹；`fill` 则会根据总宽度来补偿，即为“自然尺寸 + 补偿值”¹⁰；`samefill` 等于 `same`¹¹ + `fill`。所有 `fill` 的补偿值是一样的。样式支持缩写为首字母。

```

\rowalign \rowalign [<全局样式>] {<样式列表>}
\colalign \colalign [<全局样式>] {<样式列表>}

```

设置表格的行与列对齐方式；行对齐（垂直对齐）样式支持 `t`、`m`、`b`，列对齐（水平对齐）样式支持 `l`、`c`、`r`。设置方式与 `\rowheight` 类似。

TeX 黑客笔记： 当 <样式列表> 不含索引时，可以省略逗号，即类似“`llcrl`”一样输入。

⁷它会替代当前表的全部内容，所以应当在输入其它数据之前加载。

⁸在设计上，`header` 映射为 0，可以确保下一行为首行。

⁹同时兼容所有行/列的自然尺寸，即同组自然尺寸的最大值。

¹⁰补偿值的目的是用于保证总宽度，即为控制总宽度，需要使用缩放这些列的宽度。

¹¹但与纯 `same` 不同组，即它们的宽度是另外一个值。

1.2 渲染输出

`\printtable` `\printtable[*]` [(缩进量)]

逐行打印表格，输出的表格不带格式（如边框与背景色），但会处理基本的对齐。

TeX 黑客笔记： 逐行输出，每行是一个水平盒子，因此可以正常分页。另外，此命令是本宏包中渲染表格最快的命令。

以下章节 4 中的 JSON 输入示例的输出效果：（缩进量为 4em）

```
Name Age City
张三 30 北京
李四 25 上海
```

由于使用了水平盒子 + `\\`，其副作用是在嵌套环境中可能报错 “There’s no line here to end.”。要在类似环境中使用本命令输出表格，请使用星号版本，区别在于它不会输出 `\\`。

```
\begin{table}[!htp]
  \centering
  \caption{嵌套的 \cs{printtable}}
  \parbox[t]{\textwidth}{\printtable*}
  \label{tab: 嵌套的 printtable}
\end{table}
```

以上示例展示如何在表格浮动体中使用 `\printtable`，其效果如表 1。

表 1: 嵌套的 `\printtable`

```
Name Age City
张三 30 北京
李四 25 上海
```

```

\rendertable \rendertable [booktabs] [{中间的线}]
\rowborder   \rendertable [grid]
\colborder   {
  \rowborder [(默认值)] {边框列表}
  \colborder [(默认值)] {边框列表}
}

```

按预定义的渲染方式渲染表格。功能开发中，目前参数支持有限。前面介绍输入时显示的表格就是使用本命令（无参数）渲染的。

TeX 黑客笔记： 在渲染逻辑上，它是通过 `l3draw` 绘制的，即它是一个整体。可以直接嵌套在其它环境中（如表 2、表 3），但它不会分页。

表 2: `\rendertable[booktabs]`

Name	Sex	Age
	man	
	woman	18~25
	unknown	
John	man	18
Leon	man	20
Lily	woman	21

表 3: `\rendertable[grid]`

Name	Sex	Age
	man	
	woman	18~25
	unknown	
John	man	18
Leon	man	20
Lily	woman	21

1.3 选项参数

1.3.1 宏包选项

```

xtable/package/rowsep colsep = 0.5em, rowsep = 0.6ex
xtable/package/colsep

```

设置表格的行间/列间间距。

```

xtable/package/margin margin = {0.4em, 0.6ex}

```

设置表格的单元格的边距。

```

xtable/package/vspace vspace = {0.5ex, 0ex}

```

设置表格的上下边距。暂时只有顶边距生效。

```

xtable/package/minwidth minwidth = 1.5em, lineskip = 3ex
xtable/package/lineskip

```

设置单元格的最小宽度及行基线间距。

1.3.2 设置命令

`\linepatternset` `\linepatternset` {<名称>} {<定义>}

将指定 <名称> 的线型映射到指定 <定义>。线型定义的方式由一系列长度值组成，即 {线, 空, ...}。

`\linewidthset` `\linewidthset` {<名称>} {<尺寸>}

将指定 <名称> 的线宽映射到指定 <尺寸>。

`\linestyleset` `\linestyleset` {<名称>} {<线型>} {<线宽>} {<颜色>}

将指定 <名称> 的线样式映射到指定 <线型>、<线宽> 及 <颜色>。<线型>、<线宽> 可以是已定义的名称或字面值。

1.4 调试

本小节的命令主要用于输出一些表格的内部状态数据，以供调试使用。

`\showtable` `\showtable`

以文本形式显示当前表格的数据。效果如下：

当前表格内容如下：

No., Name, Sex, Age;

1, , man

woman

unknown, 18~25;

2, John, man, 18;

3, Leon, man, 20;

4, Lily, woman, 21。

`\logtable` `\logtable`

在日志中显示表格的核心数据。

2 关键逻辑

2.1 单元格大小

2.1.1 间距

对于无边框打印表格，列间距由 `\rowsep` 设置，表格四周无空隙。行间距由 L^AT_EX 自行按普通行处理。

对于有边框的渲染表格，单元格的边距由 `\margin` 确认。相当于行列间距有 2 倍的边距)，表格四周有单倍的边距。

2.1.2 宽度

宽度的计算步骤如下：

1. 计算所有列的自然宽度（该列中所有单元格的自然宽度的最大值）；
2. 将固定宽度的列的宽度设置为指定值；
3. 将 `same` 与 `samefill` 的宽度设置为同组最大值；
4. 计算当前总宽度与目标总宽度的差异，计算补偿值；
5. 将所有 `fill` 与 `samefill` 的列宽度加上补偿值。

注：列宽度不包含边距，但总宽度有计算。

2.1.3 高度

高度的计算步骤如下：

1. 计算所有单元格的首行填充值（用于平衡居中对齐时的基线问题）；
2. 计算所有行的自然高度与深度（该行中所有单元格的自然高度与浓度的最大值）；
3. 将固定高度的列的高度设置为（指定值-深度）；
4. 将 `same` 的高度与深度设置为同组最大值。

图 1 展示了基本的计算逻辑。

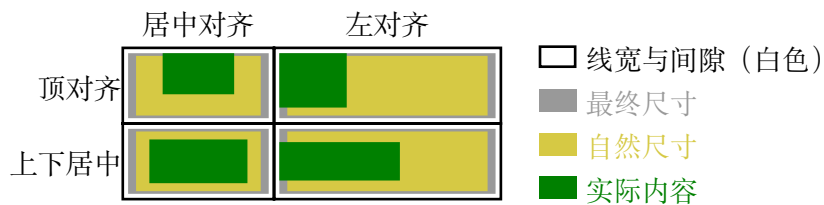


图 1: 表格渲染逻辑

3 基础定义

本宏包的开发测试环境使用的 L^AT_EX2e 的版本为 <2023-11-01>，L3 编程层的版本为 <2024-02-20>。¹²

```

1 <*package>
2 <@@=xtable>
3 \NeedsTeXFormat{LaTeX2e}[2023-11-01]
4 \ProvidesExplPackage{tabular2}{2025-09-03}{0.3}
5   {A new table implementation base on expl3}
6 \RequirePackage{l3draw}[2024-02-20]

```

3.1 通用变体

定义一些必要的系统函数的变体。

```

7 \cs_generate_variant:Nn \msg_error:nnn {nnV}
8 \cs_generate_variant:Nn \str_const:Nn {Ne}
9 \cs_generate_variant:Nn \seq_set_split:Nnn {NVn, NVV}
10 \cs_generate_variant:Nn \seq_set_item:Nnn {NnV, Nne}
11 \cs_generate_variant:Nn \seq_gset_item:Nnn {NnV, Nne}
12 \cs_generate_variant:Nn \prop_gput:Nnn {Nne, Nen, Nee, NeV, NVe, NVV}
13 \cs_generate_variant:Nn \prop_gpop:NnN {NeN}
14 \cs_generate_variant:Nn \color_stroke:n {e}
15 \cs_generate_variant:Nn \regex_extract_all:NnN {NVN}

```

3.2 l3draw 函数

在 2025-06-30 发布的新版 l3draw 中，将部分设置函数的函数名添加了 set 部分。但考虑到大部分用户可能并未升级到此版本，故增加此部分以兼容旧版本。

```

16 \cs_if_free:NT \draw_set_linewidth:n
17   { \cs_gset_eq:NN \draw_set_linewidth:n \draw_linewidth:n }
18 \cs_if_free:NT \draw_set_dash_pattern:nn
19   { \cs_gset_eq:NN \draw_set_dash_pattern:nn \draw_dash_pattern:nn }

```

定义 l3draw 函数的变体。

```

20 \cs_generate_variant:Nn \draw_set_linewidth:n {V, e}
21 \cs_generate_variant:Nn \draw_set_dash_pattern:nn {Vn, en}

```

¹²同时使用以下环境验证与打包：L^AT_EX2e 的版本为 <2025-06-01>，L3 编程层的版本为 <2025-06-30>。

3.3 消息定义

常见的错误消息。

22	<code>\msg_new:nnn {xtable} {unknown_row_name}</code>	{未知的行名称: <#1>}
23	<code>\msg_new:nnn {xtable} {unknown_col_name}</code>	{未知的列名称: <#1>}
24	<code>\msg_new:nnn {xtable} {unknown_input}</code>	{未知的输入 <#1>}
25	<code>\msg_new:nnn {xtable} {unknown_format}</code>	{未知的格式 <#1>}
26	<code>\msg_new:nnn {xtable} {unknown_cell}</code>	{未知的单元格数据 @<#1>}
27	<code>\msg_new:nnn {xtable} {outside_table}</code>	{<#1> 应当在 xtable 的输入环境中使用}
28	<code>\msg_new:nnn {xtable} {outside_render}</code>	{<#1> 应当在 xtable 的输出环境中使用}
29	<code>\msg_new:nnn {xtable} {unsaved_table}</code>	{未保存的表: <#1>}

3.4 通用变量

3.4.1 全局常量

<code>\c__xtable_space_str</code>	定义一些不易直接输入的字符常量。
<code>\c__xtable_escape_str</code>	30 <code>\str_const:Nc \c__xtable_space_str { \char_generate:nn {32} {10} } % <space></code>
<code>\c__xtable_lbrace_str</code>	31 <code>\str_const:Nc \c__xtable_escape_str { \char_generate:nn {92} {12} } % \</code>
<code>\c__xtable_rbrace_str</code>	32 <code>\str_const:Nc \c__xtable_lbrace_str { \char_generate:nn {123} {12} } % {</code> 33 <code>\str_const:Nc \c__xtable_rbrace_str { \char_generate:nn {125} {12} } % }</code>

(End of definition for \c__xtable_space_str and others.)

<code>\c__xtable_std_ht_dim</code>	单行文字的标准高度与深度。
<code>\c__xtable_std_dp_dim</code>	34 <code>\dim_const:Nn \c__xtable_std_ht_dim {1.91ex} % 约等于汉字的高度,字母高度 {1.67ex}</code> 35 <code>\dim_const:Nn \c__xtable_std_dp_dim {0.48ex} % 约等于字母的深度</code>

(End of definition for \c__xtable_std_ht_dim and \c__xtable_std_dp_dim.)

`\s__xtable_mark` 定义内部标记。

36 `\scan_new:N \s__xtable_mark`

(End of definition for \s__xtable_mark.)

3.4.2 专用变量

本小节定义一些专用的变量，以供缓存或函数之间传递特定的数据。使用这些变量的函数应当在文档中声明。

<code>\l__xtable_row_loc_int</code>	用于存储行列（数字）坐标信息。
<code>\l__xtable_col_loc_int</code>	37 <code>\int_new:N \l__xtable_row_loc_int</code> 38 <code>\int_new:N \l__xtable_col_loc_int</code>

(End of definition for \l__xtable_row_loc_int and \l__xtable_col_loc_int.)

`\l__xtable_x_dim` 用于存储绘图坐标信息。

`\l__xtable_y_dim` 39 `\dim_new:N \l__xtable_x_dim`

40 `\dim_new:N \l__xtable_y_dim`

(End of definition for \l__xtable_x_dim and \l__xtable_y_dim.)

`\l__xtable_wd_dim` 用于存储宽高深等尺寸信息。

`\l__xtable_ht_dim` 41 `\dim_new:N \l__xtable_wd_dim`

`\l__xtable_dp_dim` 42 `\dim_new:N \l__xtable_ht_dim`

`\l__xtable_ufill_dim` 43 `\dim_new:N \l__xtable_dp_dim`

`\l__xtable_dfill_dim` 44 `\dim_new:N \l__xtable_ufill_dim`

45 `\dim_new:N \l__xtable_dfill_dim`

(End of definition for \l__xtable_wd_dim and others.)

`\l__xtable_line_dim` 用于存储线宽信息。

46 `\dim_new:N \l__xtable_line_dim`

(End of definition for \l__xtable_line_dim.)

`\l__xtable_style_tl` 用于存储样式与数据的凭据表。

`\l__xtable_data_tl` 47 `\tl_new:N \l__xtable_style_tl`

48 `\tl_new:N \l__xtable_data_tl`

(End of definition for \l__xtable_style_tl and \l__xtable_data_tl.)

`\l__xtable_row_align_tl` 用于存储各项专项样式的凭据表。

`\l__xtable_col_align_tl` 49 `\tl_new:N \l__xtable_row_align_tl`

`\l__xtable_pattern_tl` 50 `\tl_new:N \l__xtable_col_align_tl`

`\l__xtable_color_tl` 51 `\tl_new:N \l__xtable_pattern_tl`

52 `\tl_new:N \l__xtable_color_tl`

(End of definition for \l__xtable_row_align_tl and others.)

`\l__xtable_shared_seq` 用于函数之间共享数据的属性表。

53 `\seq_new:N \l__xtable_shared_seq`

(End of definition for \l__xtable_shared_seq.)

`\l__xtable_cell_box` 用于存储单元格内容的盒子。

54 `\box_new:N \l__xtable_cell_box`

(End of definition for \l__xtable_cell_box.)

3.4.3 缓存变量

本小节定义一些缓存的变量，以供函数内保存数据，而不受调用函数的影响。使用这些变量的函数应当在文档中声明。

`\l__xtable_cachea_int` 用于缓存整数。
`\l__xtable_cacheb_int` `55 \int_new:N \l__xtable_cachea_int`
`56 \int_new:N \l__xtable_cacheb_int`
(End of definition for \l__xtable_cachea_int and \l__xtable_cacheb_int.)

`\l__xtable_cachea_dim` 用于缓存尺寸信息。
`\l__xtable_cacheb_dim` `57 \dim_new:N \l__xtable_cachea_dim`
`58 \dim_new:N \l__xtable_cacheb_dim`
(End of definition for \l__xtable_cachea_dim and \l__xtable_cacheb_dim.)

`\l__xtable_cachea_tl` 用于缓存的凭据表。
`\l__xtable_cacheb_tl` `59 \tl_new:N \l__xtable_cachea_tl`
`60 \tl_new:N \l__xtable_cacheb_tl`
(End of definition for \l__xtable_cachea_tl and \l__xtable_cacheb_tl.)

`\l__xtable_cachea_seq` 用于缓存的属性表。
`\l__xtable_cacheb_seq` `61 \seq_new:N \l__xtable_cachea_seq`
`62 \seq_new:N \l__xtable_cacheb_seq`
(End of definition for \l__xtable_cachea_seq and \l__xtable_cacheb_seq.)

3.4.4 临时变量

本小节定义一些临时变量¹³，以供函数内部使用¹⁴。可以随意使用这些变量而无需在文档中声明¹⁵。

`\l__xtable_tmpa_int` 整数变量。
`\l__xtable_tmpb_int` `63 \int_new:N \l__xtable_tmpa_int`
`64 \int_new:N \l__xtable_tmpb_int`
(End of definition for \l__xtable_tmpa_int and \l__xtable_tmpb_int.)

`\l__xtable_tmpa_dim` 长度变量。
`\l__xtable_tmpb_dim` `65 \dim_new:N \l__xtable_tmpa_dim`
`66 \dim_new:N \l__xtable_tmpb_dim`

¹³不使用系统提供的临时变量，以防止和其它宏包冲突。

¹⁴不应当使用这些变量在函数之间传递数据。

¹⁵应当假定调用任何函数都会修改这些变量的值。

(End of definition for `\l__xtable_tmpa_dim` and `\l__xtable_tmpb_dim`.)

```
\l__xtable_tmpa_tl  凭据表与字符串。
\l__xtable_tmpb_tl  67 \tl_new:N \l__xtable_tmpa_tl
\l__xtable_tmpa_str 68 \tl_new:N \l__xtable_tmpb_tl
\l__xtable_tmpb_str 69 \str_new:N \l__xtable_tmpa_str
                    70 \str_new:N \l__xtable_tmpb_str
```

(End of definition for `\l__xtable_tmpa_tl` and others.)

```
\l__xtable_tmpa_seq 序列与属性表。
\l__xtable_tmpb_seq 71 \seq_new:N \l__xtable_tmpa_seq
\l__xtable_tmpa_prop 72 \seq_new:N \l__xtable_tmpb_seq
\l__xtable_tmpb_prop 73 \prop_new:N \l__xtable_tmpa_prop
                    74 \prop_new:N \l__xtable_tmpb_prop
```

(End of definition for `\l__xtable_tmpa_seq` and others.)

```
\l__xtable_tmpa_box 盒子对象。
\l__xtable_tmpb_box 75 \box_new:N \l__xtable_tmpa_box
                    76 \box_new:N \l__xtable_tmpb_box
```

(End of definition for `\l__xtable_tmpa_box` and `\l__xtable_tmpb_box`.)

3.5 通用选项

3.5.1 宏包选项

```
\g__xtable_row_sep_dim 表格间隙与边空。
\g__xtable_col_sep_dim 77 \dim_new:N \g__xtable_row_sep_dim          % 行与行的最小间距（通常无边框线）
\g__xtable_row_margin_dim 78 \dim_new:N \g__xtable_col_sep_dim
\g__xtable_col_margin_dim 79 \dim_new:N \g__xtable_row_margin_dim          % 行的最小边距（通常有边框线）
\g__xtable_above_space_dim 80 \dim_new:N \g__xtable_col_margin_dim
\g__xtable_below_space_dim 81 \dim_new:N \g__xtable_above_space_dim          % 表格上下边距
                        82 \dim_new:N \g__xtable_below_space_dim
                        83 \dim_new:N \g__xtable_cell_wd_min_dim          % 单元格的最小宽度
                        84 \dim_new:N \g__xtable_cell_lineskip_dim          % 多行单元格的行距
```

(End of definition for `\g__xtable_row_sep_dim` and others.)

`xtable/package` 定义宏包选项。

```
85 \keys_define:nn { xtable / package }
86 {
87   rowsep .dim_gset:N = \g__xtable_row_sep_dim,
88   rowsep .initial:n = {0.6ex},
```

```

89   colsep .dim_gset:N = \g__xtable_col_sep_dim,
90   colsep .initial:n = {0.5em},
91   margin .code:n     =
92   {
93     \dim_gset:Nn \g__xtable_row_margin_dim { \clist_item:nn {#1} {2} }
94     \dim_gset:Nn \g__xtable_col_margin_dim { \clist_item:nn {#1} {1} }
95   },
96   margin .initial:n = {0.4em, 0.6ex},
97   vspace .code:n     =
98   {
99     \dim_gset:Nn \g__xtable_above_space_dim { \clist_item:nn {#1} {1} }
100    \dim_gset:Nn \g__xtable_below_space_dim { \clist_item:nn {#1} {2} }
101  },
102   vspace .initial:n = {0.5ex, -1ex},
103   minwidth .dim_gset:N = \g__xtable_cell_wd_min_dim,
104   minwidth .initial:n = {1.5em},
105   lineskip .dim_gset:N = \g__xtable_cell_lineskip_dim,
106   lineskip .initial:n = {3ex}
107 }
108 \ProcessKeyOptions [ xtable / package ]

```

(End of definition for xtable/package. This variable is documented on page ??.)

3.5.2 全局线样式

`\g__xtable_line_pattern_prop` 表格线型、线宽与样式。样式的值为 {线型, 线宽, 颜色} (`\s__xtable_mark` 分隔)。

```

\g__xtable_line_wd_prop 109 \prop_new:N \g__xtable_line_pattern_prop
\g__xtable_line_style_prop 110 \prop_new:N \g__xtable_line_wd_prop
111 \prop_new:N \g__xtable_line_style_prop

```

(End of definition for `\g__xtable_line_pattern_prop`, `\g__xtable_line_wd_prop`, and `\g__xtable_line_style_prop`.)

`\linepatternset` 设置边框线的线宽、线型与样式。

```

\linewidthset 112 \NewDocumentCommand {\linepatternset} { m m }
\linestyleset 113 { \prop_gput:Nnn \g__xtable_line_pattern_prop {#1} {#2} }
114 \NewDocumentCommand {\linewidthset} { m m }
115 { \prop_gput:Nnn \g__xtable_line_wd_prop {#1} {#2} }
116 \NewDocumentCommand {\linestyleset} { m m m m }
117 {
118   \prop_get:NnNF \g__xtable_line_pattern_prop {#2} \l__xtable_tmpa_tl
119   { \t1_set:Nn \l__xtable_tmpa_tl {#2} }
120   \prop_get:NnNF \g__xtable_line_wd_prop {#3} \l__xtable_tmpb_tl
121   { \t1_set:Nn \l__xtable_tmpb_tl {#3} }

```



```

122 \prop_gput:Nne \g__xtable_line_style_prop {#1}
123 {\l__xtable_tmpa_tl \s__xtable_mark \l__xtable_tmpb_tl \s__xtable_mark #4}
124 }
125 \linepatternset {soild} {}
126 \linepatternset {dotted}{0.6em, 0.15em}
127 \linepatternset {dash} {0.6em, 0.15em, 0.05em, 0.15em}
128 \linewidthset {normal} {1pt}
129 \linewidthset {thick} {1.5pt}
130 \linewidthset {thin} {0.75pt}
131 \linewidthset {toprule} {0.08em}
132 \linewidthset {midrule} {0.05em}
133 \linewidthset {cmidrule} {0.03em}
134 \linewidthset {bottomrule} {0.08em}
135 \linestyleset {normal} {soild} {normal} {black}
136 \linestyleset {dotted} {dotted} {normal} {black}
137 \linestyleset {dash} {dash} {normal} {black}
138 \linestyleset {bold} {soild} {thick} {black}

```

(End of definition for \linepatternset, \linewidthset, and \linestyleset. These functions are documented on page 9.)

3.6 通用函数

3.6.1 常规设置

```

\__xtable_init_seq:Nnn 初始化序列 #1，使其元素个数为 #2，且元素内容均为 #3。
\__xtable_ginit_seq:Nnn 139 \cs_new:Nn \__xtable_init_seq:Nnn
140 {
141   \seq_clear:N #1
142   \prg_replicate:nn {#2} { \seq_put_right:Nn #1 {#3} }
143 }
144 \cs_new:Nn \__xtable_ginit_seq:Nnn
145 {
146   \seq_gclear:N #1
147   \prg_replicate:nn {#2} { \seq_gput_right:Nn #1 {#3} }
148 }
149 \cs_generate_variant:Nn \__xtable_init_seq:Nnn {Nne}

```

(End of definition for __xtable_init_seq:Nnn and __xtable_ginit_seq:Nnn.)

```

\__xtable_int_set_max:Nn 将变量 (#1) 的值与指定值 (#2) 中的最大值赋值给变量。
\__xtable_int_gset_max:Nn 150 \cs_new:Nn \__xtable_int_set_max:Nn
151 { \int_set:Nn #1 { \int_max:nn {#1} {#2} } }
152 \cs_new:Nn \__xtable_int_gset_max:Nn

```

```
153 { \int_gset:Nn #1 { \int_max:nn {#1} {#2} } }
```

(End of definition for `_xtable_int_set_max:Nn` and `_xtable_int_gset_max:Nn`.)

`_xtable_dim_set_max:NN` 将变量 (#1) 的值与指定值 (#2) 中的最大值赋值给变量。

```
\_xtable_dim_set_max:Nn 154 \cs_new:Nn \_xtable_dim_set_max:NN
155 { \dim_set:Nn #1 { \dim_max:nn {#1} {#2} } }
156 \cs_new:Nn \_xtable_dim_set_max:Nn
157 { \dim_set:Nn #1 { \dim_max:nn {#1} {#2} } }
```

(End of definition for `_xtable_dim_set_max:NN` and `_xtable_dim_set_max:Nn`.)

3.6.2 线样式相关

`_xtable_set_line_pattern:n` 设置当前线型、线宽及样式。

```
\_xtable_set_line_wd:n 158 \cs_new:Nn \_xtable_set_line_pattern:n
\_xtable_set_line_style:n 159 {
160   \prop_get:NnNTF \g__xtable_line_pattern_prop {#1} \l__xtable_tmpa_tl
161   { \draw_set_dash_pattern:Vn \l__xtable_tmpa_tl {Opt} }
162   { \draw_set_dash_pattern:nn {#1} {0pt} }
163 }
164 \cs_new:Nn \_xtable_set_line_wd:n
165 {
166   \prop_get:NnNTF \g__xtable_line_wd_prop {#1} \l__xtable_tmpa_tl
167   { \draw_set_linewidth:V \l__xtable_tmpa_tl }
168   { \draw_set_linewidth:n {#1} }
169 }
170 \cs_new:Nn \_xtable_set_line_style:n
171 {
172   \prop_get:NnNTF \g__xtable_line_style_prop {#1} \l__xtable_tmpa_tl
173   {
174     \seq_set_split:NnV \l__xtable_tmpa_seq {\s__xtable_mark} \l__xtable_tmpa_tl
175     \draw_set_dash_pattern:en { \seq_item:Nn \l__xtable_tmpa_seq {1} } {Opt}
176     \draw_set_linewidth:e { \seq_item:Nn \l__xtable_tmpa_seq {2} }
177     \color_stroke:e { \seq_item:Nn \l__xtable_tmpa_seq {3} }
178   }
179   {
180     \_xtable_set_line_pattern:n {soild}
181     \_xtable_set_line_wd:n {#1}
182     \color_stroke:e {black}
183   }
184 }
185 \cs_generate_variant:Nn \_xtable_set_line_style:n {V}
```

(End of definition for `_xtable_set_line_pattern:n`, `_xtable_set_line_wd:n`, and `_xtable_set_line_style:n`.)

```
\_xtable_parse_line_style:n 解析当前线的样式，并将结果分别保存在 \l__xtable_pattern_tl、\l__xtable_line_dim 与 \l__xtable_color_tl 中。
186 \cs_new:Nn \_xtable_parse_line_style:n
187   {
188     \prop_get:NnNTF \g__xtable_line_style_prop {#1} \l__xtable_tmpa_tl
189     {
190       \seq_set_split:NnV \l__xtable_tmpa_seq {,} \l__xtable_tmpa_tl
191       \tl_set:Ne \l__xtable_pattern_tl { \seq_item:Nn \l__xtable_tmpa_seq {1} }
192       \dim_set:Nn \l__xtable_line_dim { \seq_item:Nn \l__xtable_tmpa_seq {2} }
193       \tl_set:Ne \l__xtable_color_tl { \seq_item:Nn \l__xtable_tmpa_seq {3} }
194     }
195     {
196       \prop_get:NnNTF \g__xtable_line_wd_prop {#1} \l__xtable_tmpa_tl
197       { \dim_set:Nn \l__xtable_line_dim { \l__xtable_tmpa_tl } }
198       { \dim_set:Nn \l__xtable_line_dim {#1} }
199       \tl_if_empty:NT \l__xtable_pattern_tl
200       { \tl_set:Nn \l__xtable_pattern_tl { soild } }
201       \tl_if_empty:NT \l__xtable_color_tl
202       { \tl_set:Nn \l__xtable_color_tl { black } }
203     }
204   }
205 \cs_generate_variant:Nn \_xtable_parse_line_style:n {V}
```

(End of definition for `_xtable_parse_line_style:n`.)

4 表格核心

在表格的内部存储中，统一使用数字坐标¹⁶为索引（键）；行名与列名仅用于输入接口。

4.1 核心存储

本小节定义一些用于存储表格数据与格式的变量。

4.1.1 行名与列名

`\g__xtable_row_name_prop` 存储行名与列名与其数字坐标的映射关系（name -> number）。唯二不使用数字坐标为索引的（集合）变量。

¹⁶数据从 1 开始，行/列标题（不是表格标题）的坐标为 0。

206 `\prop_new:N \g__xtable_row_name_prop`

207 `\prop_new:N \g__xtable_col_name_prop`

(End of definition for `\g__xtable_row_name_prop` and `\g__xtable_col_name_prop`.)

4.1.2 单元格内容

表头 (Header) 的内容始终绑定到该列最新的列名上, 无需要额外存储, 故本小节的变量均不统计表头的内容。

`\c__xtable_cell_content_tl` 定义单元格的类型。

`\c__xtable_cell_formula_tl` 208 `\tl_const:Nn \c__xtable_cell_content_tl {content}`

`\c__xtable_cell_merged_tl` 209 `\tl_const:Nn \c__xtable_cell_formula_tl {formula}`

`\c__xtable_cell_empty_tl` 210 `\tl_const:Nn \c__xtable_cell_merged_tl {merged}`

211 `\tl_const:Nn \c__xtable_cell_empty_tl {empty}`

(End of definition for `\c__xtable_cell_content_tl` and others.)

`\g__xtable_has_header_bool` 定义存储的单元格内容的属性表及最大行列数。

`\g__xtable_row_count_int` 单元格内容: row,col -> {type, content, formula} (分隔符为 `\s__xtable_mark`)。

`\g__xtable_col_count_int` 212 `\bool_new:N \g__xtable_has_header_bool`

`\g__xtable_cell_data_prop` 213 `\int_new:N \g__xtable_row_count_int` % 不统计 header

214 `\int_new:N \g__xtable_col_count_int`

215 `\prop_new:N \g__xtable_cell_data_prop` % 不记录 header

(End of definition for `\g__xtable_has_header_bool` and others.)

`\g__xtable_row_summary_prop` 行汇总与列公式。

`\g__xtable_col_formula_prop` 行汇总定义: row -> {formula_code, range};

列公式定义: col -> {formula_code, dependencies}。

216 `\prop_new:N \g__xtable_row_summary_prop`

217 `\prop_new:N \g__xtable_col_formula_prop`

(End of definition for `\g__xtable_row_summary_prop` and `\g__xtable_col_formula_prop`.)

4.1.3 单元格样式

与单元格内容不同, 单元格样式会记录 Header 信息。

`\g__xtable_row_ht_style_seq` 行列的内容样式信息。

`\g__xtable_col_wd_style_seq` 218 `\seq_new:N \g__xtable_row_ht_style_seq`

`\g__xtable_row_align_seq` 219 `\seq_new:N \g__xtable_col_wd_style_seq`

`\g__xtable_col_align_seq` 220 `\seq_new:N \g__xtable_row_align_seq`

221 `\seq_new:N \g__xtable_col_align_seq`

(End of definition for `\g__xtable_row_ht_style_seq` and others.)

4.1.4 合并单元格

`\g__xtable_merge_count_int` 合并单元格数量、合并单元格信息表及引用表。
`\g__xtable_merge_info_prop` 信息表: `merge_id -> {start_row, start_col, end_row, end_col, content}`;
`\g__xtable_merge_ref_prop` 引用表: `row,col -> merge_id`。

```

222 \int_new:N \g__xtable_merge_count_int
223 \prop_new:N \g__xtable_merge_info_prop
224 \prop_new:N \g__xtable_merge_ref_prop

```

(End of definition for \g__xtable_merge_count_int, \g__xtable_merge_info_prop, and \g__xtable_merge_ref_prop.)

4.1.5 中间变量

`\g__xtable_col_header_seq` 用于存储列标题的序列，在输入完成后需要刷新此变量的值。

```

225 \seq_new:N \g__xtable_col_header_seq

```

(End of definition for \g__xtable_col_header_seq.)

`\g__xtable_row_sep_seq` 用于存储间隙信息，在渲染前需要首先刷新此变量的值。

```

226 \seq_new:N \g__xtable_row_sep_seq
227 \seq_new:N \g__xtable_col_sep_seq

```

(End of definition for \g__xtable_row_sep_seq and \g__xtable_col_sep_seq.)

`\g__xtable_row_ht_seq` 用于行/列的存储高深宽信息，在渲染前需要刷新此变量的值。如果列宽有使用填充样式，则需要刷新列间隙后再刷新列宽。

```

228 \seq_new:N \g__xtable_row_ht_seq
229 \seq_new:N \g__xtable_row_dp_seq
230 \seq_new:N \g__xtable_col_wd_seq

```

(End of definition for \g__xtable_row_ht_seq, \g__xtable_row_dp_seq, and \g__xtable_col_wd_seq.)

`\g__xtable_cell_ht_prop` 单元格填充值（补偿首行字高及未行字深）及填充后高度，在渲染前需要刷新此变量的值。

```

231 \prop_new:N \g__xtable_cell_ht_prop
232 \prop_new:N \g__xtable_cell_ufill_prop
233 \prop_new:N \g__xtable_cell_dfill_prop

```

(End of definition for \g__xtable_cell_ht_prop, \g__xtable_cell_ufill_prop, and \g__xtable_cell_dfill_prop.)

`\g__xtable_row_loc_seq` 用于存储表格的行列坐标（图形坐标）信息。需要考虑是否有 Header，需要在渲染条件确认后再刷新。

```

234 \seq_new:N \g__xtable_row_loc_seq
235 \seq_new:N \g__xtable_col_loc_seq

```

(End of definition for `\g__xtable_row_loc_seq` and `\g__xtable_col_loc_seq`.)

`\g__xtable_row_border_seq` 用于存储行列边框线。需要考虑是否有 Header。

`\g__xtable_col_border_seq` 236 `\seq_new:N \g__xtable_row_border_seq`

237 `\seq_new:N \g__xtable_col_border_seq`

(End of definition for `\g__xtable_row_border_seq` and `\g__xtable_col_border_seq`.)

4.2 行列定位

4.2.1 行列名设置

`_xtable_set_excel_col_names:n` 设置 Excel 风格的列名，#1 为终止列数。

238 `\cs_new:Nn _xtable_set_excel_col_names:n`

239 {

240 `\int_step_inline:nm {#1}`

241 {

242 `\prop_gput:Nen \g__xtable_col_name_prop`

243 `{ \int_to_Alph:n {##1} } {##1}`

244 }

245 }

(End of definition for `_xtable_set_excel_col_names:n`.)

`_xtable_set_row_name:nN` 设置行名/列名：#1 为行名/列名，#2 为数字列坐标。

`_xtable_set_col_name:nN` 246 `\cs_new:Nn _xtable_set_row_name:nN`

247 {

248 `\prop_gput:Nne \g__xtable_row_name_prop {#1} {\int_use:N #2}`

249 `_xtable_int_gset_max:Nn \g__xtable_row_count_int {#2}`

250 }

251 `\cs_new:Nn _xtable_set_col_name:nN`

252 {

253 `\prop_gput:Nne \g__xtable_col_name_prop {#1} {\int_use:N #2}`

254 `_xtable_int_gset_max:Nn \g__xtable_col_count_int {#2}`

255 }

256 `\cs_generate_variant:Nn _xtable_set_row_name:nN {VN}`

257 `\cs_generate_variant:Nn _xtable_set_col_name:nN {VN}`

(End of definition for `_xtable_set_row_name:nN` and `_xtable_set_col_name:nN`.)

4.2.2 行列名解析

`_xtable_get_loc:NnNn` 返回指定行名、列名对应的数字坐标。#1: 名称属性表, #2: 名称, #3: 结果变量, #4 查找失败时动作。

```

258 \cs_new:Nn \_xtable_get_loc:NnNn
259   {
260     \regex_match:nnTF {^-?[0-9]+$} {#2}
261     { \int_set:Nn #3 {#2} }
262     {
263       \prop_get:NnNTF #1 {#2} \l__xtable_tmpa_tl
264       { \int_set:Nn #3 {\l__xtable_tmpa_tl} }
265       { #4 }
266     }
267   }

```

(End of definition for _xtable_get_loc:NnNn.)

`_xtable_parse_row_loc:n` 解析坐标 (支持数字和名称混合), 如果解析失败则报错, #1: 行名/列名。解析的坐标保存在 `\l__xtable_row_loc_int` 或 `\l__xtable_col_loc_int` 中。

`_xtable_parse_col_loc:n`

```

268 \cs_new:Nn \_xtable_parse_row_loc:n
269   {
270     \_xtable_get_loc:NnNn \g__xtable_row_name_prop {#1} \l__xtable_row_loc_int
271     { \msg_error:nnn {xtable} {unknown_row_name} {#1} }
272   }
273 \cs_new:Nn \_xtable_parse_col_loc:n
274   {
275     \_xtable_get_loc:NnNn \g__xtable_col_name_prop {#1} \l__xtable_col_loc_int
276     { \msg_error:nnn {xtable} {unknown_col_name} {#1} }
277   }
278 \cs_generate_variant:Nn \_xtable_parse_row_loc:n {e, V}
279 \cs_generate_variant:Nn \_xtable_parse_col_loc:n {e, V}

```

(End of definition for _xtable_parse_row_loc:n and _xtable_parse_col_loc:n.)

`_xtable_parse_new_row_loc:n` 解析坐标 (支持数字和名称混合), 如果解析失败则尝试建议新关联, #1: 行名/列名。坐标保存在 `\l__xtable_row_loc_int` 或 `\l__xtable_col_loc_int` 中。

`_xtable_parse_new_col_loc:n`

```

280 \cs_new:Nn \_xtable_parse_new_row_loc:n
281   {
282     \_xtable_get_loc:NnNn \g__xtable_row_name_prop {#1} \l__xtable_row_loc_int
283     {
284       \int_if_zero:nT { \l__xtable_row_loc_int }
285       { \int_set:Nn \l__xtable_row_loc_int {\g__xtable_row_count_int + 1} }
286       \_xtable_set_row_name:nN {#1} \l__xtable_row_loc_int

```

```

287     }
288   }
289   \cs_new:Nn \__xtable_parse_new_col_loc:n
290   {
291     \__xtable_get_loc:NnNn \g__xtable_col_name_prop {#1} \l__xtable_col_loc_int
292     {
293       \int_if_zero:nT { \l__xtable_col_loc_int }
294       { \int_set:Nn \l__xtable_col_loc_int { \g__xtable_col_count_int + 1 } }
295       \__xtable_set_col_name:nN {#1} \l__xtable_col_loc_int
296     }
297   }
298   \cs_generate_variant:Nn \__xtable_parse_new_row_loc:n {e, V}
299   \cs_generate_variant:Nn \__xtable_parse_new_col_loc:n {e, V}

```

(End of definition for __xtable_parse_new_row_loc:n and __xtable_parse_new_col_loc:n.)

__xtable_parse_coord:n 解析坐标(支持数字和名称混合), #1:(行名, 列名)。解析的坐标保存在 \l__xtable_row_loc_int 与 \l__xtable_col_loc_int 中。两个版本的区别在于如果解析失败如何处理。

```

300   \cs_new:Nn \__xtable_parse_coord:n
301   {
302     \__xtable_parse_row_loc:e { \clist_item:nn {#1} {1} }
303     \__xtable_parse_col_loc:e { \clist_item:nn {#1} {2} }
304   }
305   \cs_new:Nn \__xtable_parse_new_coord:n
306   {
307     \__xtable_parse_new_row_loc:e { \clist_item:nn {#1} {1} }
308     \__xtable_parse_new_col_loc:e { \clist_item:nn {#1} {2} }
309   }

```

(End of definition for __xtable_parse_coord:n and __xtable_parse_new_coord:n.)

4.3 内容处理

4.3.1 初始化与保存

__xtable_table_init: 将初始化为空表。

```

310   \cs_new:Nn \__xtable_table_init:
311   {
312     \int_gzero:N \g__xtable_row_count_int
313     \int_gzero:N \g__xtable_col_count_int
314     \int_gzero:N \g__xtable_merge_count_int
315     \prop_gclear:N \g__xtable_row_name_prop

```



```

316 \prop_gclear:N \g__xtable_col_name_prop
317 \prop_gclear:N \g__xtable_cell_data_prop
318 \prop_gclear:N \g__xtable_row_summary_prop
319 \prop_gclear:N \g__xtable_col_formula_prop
320 \seq_gclear:N \g__xtable_row_ht_style_seq
321 \seq_gclear:N \g__xtable_col_wd_style_seq
322 \seq_gclear:N \g__xtable_row_align_seq
323 \seq_gclear:N \g__xtable_col_align_seq
324 \prop_gclear:N \g__xtable_merge_info_prop
325 \prop_gclear:N \g__xtable_merge_ref_prop
326 \prop_gput:Nnn \g__xtable_row_name_prop {header} {0}
327 \prop_gput:Nnn \g__xtable_row_name_prop {title} {0}
328 \bool_gset_true:N \g__xtable_has_header_bool
329 }

```

(End of definition for `__xtable_table_init:`.)

`__xtable_table_save:n` 保存表格。

```

330 \cs_new:Nn \__xtable_table_save:n
331 {
332   \cs_if_exist:cF { g__xtable_row_count_#1_int }
333   {
334     \bool_new:c { g__xtable_has_header_#1_bool }
335     \int_new:c { g__xtable_row_count_#1_int }
336     \int_new:c { g__xtable_col_count_#1_int }
337     \int_new:c { g__xtable_merge_count_#1_int }
338     \prop_new:c { g__xtable_row_name_#1_prop }
339     \prop_new:c { g__xtable_col_name_#1_prop }
340     \prop_new:c { g__xtable_cell_data_#1_prop }
341     \prop_new:c { g__xtable_row_summary_#1_prop }
342     \prop_new:c { g__xtable_col_formula_#1_prop }
343     \seq_new:c { g__xtable_row_ht_style_#1_seq }
344     \seq_new:c { g__xtable_col_wd_style_#1_seq }
345     \seq_new:c { g__xtable_row_align_#1_seq }
346     \seq_new:c { g__xtable_col_align_#1_seq }
347     \prop_new:c { g__xtable_merge_info_#1_prop }
348     \prop_new:c { g__xtable_merge_ref_#1_prop }
349   }
350   \bool_gset_eq:cN { g__xtable_has_header_#1_bool } \g__xtable_has_header_bool
351   \int_gset_eq:cN { g__xtable_row_count_#1_int } \g__xtable_row_count_int
352   \int_gset_eq:cN { g__xtable_col_count_#1_int } \g__xtable_col_count_int
353   \int_gset_eq:cN { g__xtable_merge_count_#1_int } \g__xtable_merge_count_int
354   \prop_gset_eq:cN { g__xtable_row_name_#1_prop } \g__xtable_row_name_prop

```

```

355 \prop_gset_eq:cN { g__xtable_col_name_#1_prop } \g__xtable_col_name_prop
356 \prop_gset_eq:cN { g__xtable_cell_data_#1_prop } \g__xtable_cell_data_prop
357 \prop_gset_eq:cN { g__xtable_row_summary_#1_prop } \g__xtable_row_summary_prop
358 \prop_gset_eq:cN { g__xtable_col_formula_#1_prop } \g__xtable_col_formula_prop
359 \seq_gset_eq:cN { g__xtable_row_ht_style_#1_seq } \g__xtable_row_ht_style_seq
360 \seq_gset_eq:cN { g__xtable_col_wd_style_#1_seq } \g__xtable_col_wd_style_seq
361 \seq_gset_eq:cN { g__xtable_row_align_#1_seq } \g__xtable_row_align_seq
362 \seq_gset_eq:cN { g__xtable_col_align_#1_seq } \g__xtable_col_align_seq
363 \prop_gset_eq:cN { g__xtable_merge_info_#1_prop } \g__xtable_merge_info_prop
364 \prop_gset_eq:cN { g__xtable_merge_ref_#1_prop } \g__xtable_merge_ref_prop
365 }

```

(End of definition for __xtable_table_save:n.)

__xtable_table_restore:n 恢复表格。

```

366 \cs_new:Nn \__xtable_table_restore:n
367 {
368   \cs_if_exist:cF { g__xtable_row_count_#1_int }
369     { \msg_error:nmn {xtable} {unsaved_table} {#1} }
370   \bool_gset_eq:Nc \g__xtable_has_header_bool { g__xtable_has_header_#1_bool }
371   \int_gset_eq:Nc \g__xtable_row_count_int { g__xtable_row_count_#1_int }
372   \int_gset_eq:Nc \g__xtable_col_count_int { g__xtable_col_count_#1_int }
373   \int_gset_eq:Nc \g__xtable_merge_count_int { g__xtable_merge_count_#1_int }
374   \prop_gset_eq:Nc \g__xtable_row_name_prop { g__xtable_row_name_#1_prop }
375   \prop_gset_eq:Nc \g__xtable_col_name_prop { g__xtable_col_name_#1_prop }
376   \prop_gset_eq:Nc \g__xtable_cell_data_prop { g__xtable_cell_data_#1_prop }
377   \prop_gset_eq:Nc \g__xtable_row_summary_prop { g__xtable_row_summary_#1_prop }
378   \prop_gset_eq:Nc \g__xtable_col_formula_prop { g__xtable_col_formula_#1_prop }
379   \seq_gset_eq:Nc \g__xtable_row_ht_style_seq { g__xtable_row_ht_style_#1_seq }
380   \seq_gset_eq:Nc \g__xtable_col_wd_style_seq { g__xtable_col_wd_style_#1_seq }
381   \seq_gset_eq:Nc \g__xtable_row_align_seq { g__xtable_row_align_#1_seq }
382   \seq_gset_eq:Nc \g__xtable_col_align_seq { g__xtable_col_align_#1_seq }
383   \prop_gset_eq:Nc \g__xtable_merge_info_prop { g__xtable_merge_info_#1_prop }
384   \prop_gset_eq:Nc \g__xtable_merge_ref_prop { g__xtable_merge_ref_#1_prop }
385 }

```

(End of definition for __xtable_table_restore:n.)

4.3.2 内容设置

__xtable_set_cell:nmnnn 设置单元格数据。#1#2 数字坐标，#3 类型，#4 内容，#5 公式。

```

386 \cs_new:Nn \__xtable_set_cell:nmnnn
387 {

```

```

388     \prop_gput:Nnn \g__xtable_cell_data_prop
389     {#1, #2} {#3 \s__xtable_mark #4 \s__xtable_mark #5}
390     \__xtable_int_gset_max:Nn \g__xtable_row_count_int {#1}
391     \__xtable_int_gset_max:Nn \g__xtable_col_count_int {#2}
392   }
393 \cs_generate_variant:Nn \__xtable_set_cell:nnnnn {eeVnn}

```

(End of definition for __xtable_set_cell:nnnnn.)

__xtable_set_cell:NNn 设置单元格内容。#1#2 数字坐标，#3 内容。

```

\__xtable_set_cell:n 394 \cs_new:Nn \__xtable_set_cell:NNn
395   {
396     \tl_if_empty:nTF {#3}
397     {
398       \prop_gpop:NeN \g__xtable_cell_data_prop
399       { \int_use:N #1 , \int_use:N #2 }
400       \l__xtable_tmpa_tl
401     }
402     {
403       \__xtable_set_cell:eeVnn
404       { \int_use:N #1 } { \int_use:N #2 }
405       \c__xtable_cell_content_tl {#3} {}
406     }
407   }

```

设置单元格内容。#1: 内容，坐标使用 \l__xtable_row_loc_int 与 \l__xtable_col_loc_int。

```

408 \cs_new:Nn \__xtable_set_cell:n
409   {
410     \__xtable_set_cell:NNn
411     \l__xtable_row_loc_int \l__xtable_col_loc_int {#1}
412   }
413 \cs_generate_variant:Nn \__xtable_set_cell:NNn {NNV}

```

(End of definition for __xtable_set_cell:NNn and __xtable_set_cell:n.)

4.3.3 内容查询

__xtable_parse_cell:nn 获取表格内容。#1#2 数字坐标，结果保存在 \l__xtable_data_tl 中。支持获取 Header 内容。

```

414 \cs_new:Nn \__xtable_parse_cell:nn
415   {
416     \prop_get:NnNTF \g__xtable_cell_data_prop {#1, #2} \l__xtable_tmpa_tl
417     {

```

```

418     \seq_set_split:NnV \l__xtable_tmpa_seq {\s__xtable_mark} \l__xtable_tmpa_tl
419     \str_case_e:nnF { \seq_item:Nn \l__xtable_tmpa_seq {1} }
420     {
421         { \c__xtable_cell_content_tl } % 内容
422         { \tl_set:Ne \l__xtable_data_tl { \seq_item:Nn \l__xtable_tmpa_seq {2} } }
423         { \c__xtable_cell_formula_tl } % 公式
424         { \tl_set:Ne \l__xtable_data_tl { \seq_item:Nn \l__xtable_tmpa_seq {2} } }
425         { \c__xtable_cell_merged_tl } % 合并
426         { \tl_set:Nn \l__xtable_data_tl {} }
427         { \c__xtable_cell_empty_tl } % 空
428         { \tl_set:Nn \l__xtable_data_tl {} }
429     }
430     { \msg_error:nnV {xtable} {unknown_cell} {#1,#2}}
431 }
432 {
433     \str_if_eq:nnTF {#1} {0}
434     { \tl_set:Ne \l__xtable_data_tl { \seq_item:Nn \g__xtable_col_header_seq {#2} } }
435     { \tl_set:Nn \l__xtable_data_tl {} } % 空单元格
436 }
437 }
438 \cs_generate_variant:Nn {\__xtable_get_cell:nnN} {eeN}
439 \cs_generate_variant:Nn {\__xtable_parse_cell:nn} {ee}

```

(End of definition for __xtable_parse_cell:nn.)

4.4 格式处理

4.4.1 样式设置

__xtable_set_row_style:NNn 设置行的样式。#1: 样式名, #2: 样式内容, #3: 默认值。本命令会修改变量 \l__xtable_row_loc_int 的值。

```

440 \cs_new:Nn \__xtable_set_row_style:NNn
441 {
442     \bool_if:NTF \g__xtable_has_header_bool
443     {
444         \int_set:Nn \l__xtable_tmpa_int {1}
445         \int_set:Nn \l__xtable_row_loc_int {0}
446     } {
447         \int_set:Nn \l__xtable_tmpa_int {0}
448         \int_set:Nn \l__xtable_row_loc_int {1}
449     }
450     \__xtable_int_gset_max:Nn \g__xtable_row_count_int
451     { \seq_count:N #1 - \l__xtable_tmpa_int }

```

```

452 \_xtable_ginit_seq:Nnn #1 {\g_xtable_row_count_int + 1} {#3}
453 \seq_map_inline:Nn #2
454 {
455   \str_if_in:nnTF {##1} {=}
456   {
457     \seq_set_split:Nnn \l__xtable_tmpa_seq {=} {##1}
458     \seq_get_left:NN \l__xtable_tmpa_seq \l__xtable_tmpa_tl
459     \seq_get_right:NN \l__xtable_tmpa_seq \l__xtable_tmpb_tl
460     \_xtable_parse_row_loc:V \l__xtable_tmpa_tl
461   }
462   { \tl_set:Nn \l__xtable_tmpb_tl {##1} }
463   \seq_gset_item:NnV #1
464   {\l__xtable_row_loc_int + 1} \l__xtable_tmpb_tl
465   \int_incr:N \l__xtable_row_loc_int
466 }
467 }

```

设置列的样式。#1: 样式名, #2: 样式内容, #3: 默认值。本命令会修改变量 `\l__xtable_col_loc_int` 的值。

```

468 \cs_new:Nn \_xtable_set_col_style:NNn
469 {
470   \int_set:Nn \l__xtable_col_loc_int {1}
471   \_xtable_int_gset_max:Nn \g_xtable_col_count_int {\seq_count:N #1}
472   \_xtable_ginit_seq:Nnn #1 {\g_xtable_col_count_int} {#3}
473   \seq_map_inline:Nn #2
474   {
475     \str_if_in:nnTF {##1} {=}
476     {
477       \seq_set_split:Nnn \l__xtable_tmpa_seq {=} {##1}
478       \seq_get_left:NN \l__xtable_tmpa_seq \l__xtable_tmpa_tl
479       \seq_get_right:NN \l__xtable_tmpa_seq \l__xtable_tmpb_tl
480       \_xtable_parse_col_loc:V \l__xtable_tmpa_tl
481     }
482     { \tl_set:Nn \l__xtable_tmpb_tl {##1} }
483     \seq_gset_item:NnV #1 {\l__xtable_col_loc_int} \l__xtable_tmpb_tl
484     \int_incr:N \l__xtable_col_loc_int
485   }
486 }

```

(End of definition for `_xtable_set_row_style:NNn` and `_xtable_set_col_style:NNn`.)

`_xtable_set_row_ht_style:Nn` 设置行高的样式。#1: 样式内容, #2: 默认值。

```

\_xtable_set_col_wd_style:NNn 487 \cs_new:Nn \_xtable_set_row_ht_style:Nn

```

```

488 {
489   \_xtable_set_row_style:Nn \g__xtable_row_ht_style_seq #1 {#2}
490   \int_step_inline:nn
491     { \seq_count:N \g__xtable_row_ht_style_seq }
492     {
493       \tl_set:Ne \l__xtable_tmpa_tl
494         { \seq_item:Nn \g__xtable_row_ht_style_seq {##1} }
495       \tl_replace_all:Nnn \l__xtable_tmpa_tl {auto} {a}
496       \tl_replace_all:Nnn \l__xtable_tmpa_tl {same} {s}
497       \seq_gset_item:NnV \g__xtable_row_ht_style_seq {##1} \l__xtable_tmpa_tl
498     }
499 }

```

设置列宽的样式。#1: 样式内容, #2: 默认值, #3: 总宽。

```

500 \cs_new:Nn \_xtable_set_col_wd_style:Nnn
501 {
502   \_xtable_set_col_style:Nn \g__xtable_col_wd_style_seq #1 {#2}
503   \int_step_inline:nn
504     { \seq_count:N \g__xtable_col_wd_style_seq }
505     {
506       \tl_set:Ne \l__xtable_tmpa_tl
507         { \seq_item:Nn \g__xtable_col_wd_style_seq {##1} }
508       \tl_replace_all:Nnn \l__xtable_tmpa_tl {auto} {a}
509       \tl_replace_all:Nnn \l__xtable_tmpa_tl {same} {s}
510       \tl_replace_all:Nnn \l__xtable_tmpa_tl {fill} {f}
511       \seq_gset_item:NnV \g__xtable_col_wd_style_seq {##1} \l__xtable_tmpa_tl
512     }
513   \seq_gput_right:Nn \g__xtable_col_wd_style_seq {#3}
514 }

```

(End of definition for _xtable_set_row_ht_style:Nn and _xtable_set_col_wd_style:Nnn.)

_xtable_set_row_align:Nn 设置行高样式。

```

\_xtable_set_col_align:Nn 515 \cs_new:Nn \_xtable_set_row_align:Nn
516 { \_xtable_set_row_style:Nn \g__xtable_row_align_seq #1 {#2} }
517 \cs_new:Nn \_xtable_set_col_align:Nn
518 { \_xtable_set_col_style:Nn \g__xtable_col_align_seq #1 {#2} }

```

(End of definition for _xtable_set_row_align:Nn and _xtable_set_col_align:Nn.)

_xtable_insure_style: 确保样式有设置好。本命令会修改变量 \l__xtable_shared_seq 的值。

```

519 \cs_new:Nn \_xtable_insure_style:
520 {
521   \seq_clear:N \l__xtable_shared_seq

```

```

522 \seq_if_empty:NT \g__xtable_row_ht_style_seq
523   { \__xtable_set_row_ht_style:Nn \l__xtable_shared_seq {a} }
524 \seq_if_empty:NT \g__xtable_col_wd_style_seq
525   { \__xtable_set_col_wd_style:Nnn \l__xtable_shared_seq {a} {\textwidth} }
526 \seq_if_empty:NT \g__xtable_row_align_seq
527   { \__xtable_set_row_align:Nn \l__xtable_shared_seq {b} }
528 \seq_if_empty:NT \g__xtable_col_align_seq
529   { \__xtable_set_col_align:Nn \l__xtable_shared_seq {c} }
530 }

```

(End of definition for `__xtable_insure_style:.`)

4.4.2 样式查询

`__xtable_parse_row_ht_style:n` 解析行高/列宽的样式。#1: 数字坐标, 结果保存在 `\l__xtable_style_tl` 中。

```

\__xtable_parse_col_wd_style:n 531 \cs_new:Nn \__xtable_parse_row_ht_style:n
532   {
533     \tl_set:Ne \l__xtable_style_tl
534       { \seq_item:Nn \g__xtable_row_ht_style_seq {#1 + 1} }
535   }
536 \cs_new:Nn \__xtable_parse_col_wd_style:n
537   {
538     \tl_set:Ne \l__xtable_style_tl
539       { \seq_item:Nn \g__xtable_col_wd_style_seq {#1} }
540   }

```

(End of definition for `__xtable_parse_row_ht_style:n` and `__xtable_parse_col_wd_style:n`.)

`__xtable_parse_row_align:n` 解析行高/列宽的样式。#1: 数字坐标, 结果保存在 `\l__xtable_row_align_tl` 与
`__xtable_parse_col_align:n` `\l__xtable_col_align_tl` 中。

```

541 \cs_new:Nn \__xtable_parse_row_align:n
542   {
543     \tl_set:Ne \l__xtable_row_align_tl
544       { \seq_item:Nn \g__xtable_row_align_seq {#1 + 1} }
545   }
546 \cs_new:Nn \__xtable_parse_col_align:n
547   {
548     \tl_set:Ne \l__xtable_col_align_tl
549       { \seq_item:Nn \g__xtable_col_align_seq {#1} }
550   }

```

(End of definition for `__xtable_parse_row_align:n` and `__xtable_parse_col_align:n`.)

4.5 调试

`\showtable` 显示表存储的内容，使用变量 `\l__xtable_data_tl`。

```
551 \NewDocumentCommand \showtable {}
552   {
```

输出列标题。

```
553   \prop_if_empty:NTF \g__xtable_cell_data_prop
554     {当前表格内容为空。}
555     {当前表格内容如下: \\No.}
556   \seq_map_inline:Nn \g__xtable_col_header_seq
557     { ,~\tl_to_str:n {##1} }
```

输出内容。

```
558   \int_step_inline:nn {\g__xtable_row_count_int}
559     {
560       ;\\##1
561       \int_step_inline:nn {\g__xtable_col_count_int}
562         {
563           \__xtable_parse_cell:nn {##1} {###1}
564           ,~\l__xtable_data_tl
565         }
566     }。
567 }
```

(End of definition for \showtable. This function is documented on page 9.)

`\logtable` 在日志中显示表格的核心数据。

```
568 \NewDocumentCommand \logtable {}
569   {
570     \int_log:N \g__xtable_row_count_int
571     \int_log:N \g__xtable_col_count_int
572     \int_log:N \g__xtable_merge_count_int
573     \prop_log:N \g__xtable_row_name_prop
574     \prop_log:N \g__xtable_col_name_prop
575     \prop_log:N \g__xtable_cell_data_prop
576     \prop_log:N \g__xtable_row_summary_prop
577     \prop_log:N \g__xtable_col_formula_prop
578     \seq_log:N \g__xtable_row_ht_style_seq
579     \seq_log:N \g__xtable_col_wd_style_seq
580     \seq_log:N \g__xtable_row_align_seq
581     \seq_log:N \g__xtable_col_align_seq
582     \prop_log:N \g__xtable_merge_info_prop
583     \prop_log:N \g__xtable_merge_ref_prop
```



```

584     \bool_log:N \g__xtable_has_header_bool
585   }

```

(End of definition for \logtable. This function is documented on page 9.)

5 中间计算

5.1 列标题

_xtable_process_header: 处理表格列标题。

```

586 \cs_new:Nn \_xtable_process_header:
587   {
588     \prop_clear:N \l__xtable_tmpa_prop
589     \seq_gclear:N \g__xtable_col_header_seq
590     \prop_map_inline:Nn \g__xtable_col_name_prop
591       { \prop_put:Nnn \l__xtable_tmpa_prop {##2} {##1} }
592     \int_step_inline:nn {\g__xtable_col_count_int}
593       {
594         \prop_get:NnNTF \l__xtable_tmpa_prop {##1} \l__xtable_tmpa_tl
595         { \seq_gput_right:NV \g__xtable_col_header_seq \l__xtable_tmpa_tl }
596         { \seq_gput_right:Nn \g__xtable_col_header_seq {<##1>} }
597       }
598   }

```

(End of definition for _xtable_process_header:.)

5.2 行列尺寸

5.2.1 边距设置

_xtable_set_row_sep:nn 设置列边距（内容到内容的距离）。#1: 最左边与最右边的边距，#2: 中间的边距。

```

\_xtable_set_col_sep:nn 599 \cs_new:Nn \_xtable_set_row_sep:nn
600   {
601     \seq_gclear:N \g__xtable_row_sep_seq
602     \prg_replicate:mn {\g__xtable_row_count_int}
603       { \seq_gput_right:Nn \g__xtable_row_sep_seq {#2} }
604     \seq_gput_left:Nn \g__xtable_row_sep_seq {#1}
605     \seq_gput_right:Nn \g__xtable_row_sep_seq {#1}
606   }
607 \cs_new:Nn \_xtable_set_col_sep:nn
608   {
609     \seq_gclear:N \g__xtable_col_sep_seq
610     \prg_replicate:mn {\g__xtable_col_count_int - 1}

```

```

611     { \seq_gput_right:Nn \g__xtable_col_sep_seq {#2} }
612   \seq_gput_left:Nn \g__xtable_col_sep_seq {#1}
613   \seq_gput_right:Nn \g__xtable_col_sep_seq {#1}
614 }
615 \cs_generate_variant:Nn \__xtable_set_row_sep:nn { ne, en, ee }
616 \cs_generate_variant:Nn \__xtable_set_col_sep:nn { ne, en, ee }

```

(End of definition for __xtable_set_row_sep:nn and __xtable_set_col_sep:nn.)

5.2.2 宽度计算

__xtable_measure_cell_wd:n 测量单元格内容 (#1) 的自然宽度，结果保存在 \l__xtable_wd_dim 中。顺便会更新首行/未行的高度差/深度差¹⁷，结果保存在 \l__xtable_ufill_dim \l__xtable_dfill_dim 中。

```

617 \cs_new:Nn \__xtable_measure_cell_wd:n
618 {
619   \dim_zero:N \l__xtable_ufill_dim
620   \dim_set_eq:NN \l__xtable_wd_dim \g__xtable_cell_wd_min_dim
621   \tl_if_empty:nF {#1}
622   {
623     \bool_set_true:N \l__xtable_tmpa_bool
624     \seq_set_split:Nnn \l__xtable_tmpa_seq {\} {#1}
625     \seq_map_inline:Nn \l__xtable_tmpa_seq
626     {
627       \hbox_set:Nn \l__xtable_tmpa_box {##1}
628       \__xtable_dim_set_max:Nn \l__xtable_wd_dim
629       { \box_wd:N \l__xtable_tmpa_box }
630       \bool_if:NT \l__xtable_tmpa_bool
631       {
632         \__xtable_dim_set_max:Nn \l__xtable_ufill_dim
633         { \c__xtable_std_ht_dim - \box_ht:N \l__xtable_tmpa_box }
634         \bool_set_false:N \l__xtable_tmpa_bool
635       }
636       \dim_zero:N \l__xtable_dfill_dim % 每次都清除上次信息
637       \__xtable_dim_set_max:Nn \l__xtable_dfill_dim
638       { \c__xtable_std_dp_dim - \box_dp:N \l__xtable_tmpa_box }
639     }
640   }
641 }
642 \cs_generate_variant:Nn \__xtable_measure_cell_wd:n { V, e }

```

(End of definition for __xtable_measure_cell_wd:n.)

¹⁷高度与 \c__xtable_ht_dim 的差正值 (为正)，如果高度更大，则为零；深度同理。

`_xtable_measure_col_wd:n` 测量表格的列 (#1) 的宽度, 结果保存在 `\l__xtable_wd_dim` 中; 同时会记录填充值 (使用变量 `\l__xtable_ufill_dim` 和 `\l__xtable_dfill_dim` 中转)。本函数会使用 `\l__xtable_data_tl` 变量。¹⁸

```

643 \cs_new:Nn \_xtable_measure_col_wd:n
644 {
645   \dim_zero:N \l__xtable_tmpa_dim
646   \bool_if:NTF \g__xtable_has_header_bool
647     { \int_set:Nn \l__xtable_tmpa_int {0} }
648     { \int_set:Nn \l__xtable_tmpa_int {1} }
649   \int_step_inline:nnn {\l__xtable_tmpa_int} {\g__xtable_row_count_int}
650   {
651     \_xtable_parse_cell:nn {##1} {#1}
652     \_xtable_measure_cell_wd:V \l__xtable_data_tl
653     \_xtable_dim_set_max:NN \l__xtable_tmpa_dim \l__xtable_wd_dim
654     \prop_gput:Nne \g__xtable_cell_ufill_prop
655       {##1,#1} {\dim_use:N \l__xtable_ufill_dim}
656     \prop_gput:Nne \g__xtable_cell_dfill_prop
657       {##1,#1} {\dim_use:N \l__xtable_dfill_dim}
658   }
659   \dim_set_eq:NN \l__xtable_wd_dim \l__xtable_tmpa_dim
660 }

```

(End of definition for `_xtable_measure_col_wd:n`.)

`_xtable_calc_col_wd:` 计算表格各列的宽度, 如果有需要调整的列宽, 必须刷新出列边距, 否则计算会失真。本命令会大量修改通用变量的值, 调用时需要注意。

```

661 \cs_new:Nn \_xtable_calc_col_wd:
662 {
663   \dim_zero:N \l__xtable_cachea_dim
664   \dim_zero:N \l__xtable_cacheb_dim
665   \seq_clear:N \l__xtable_cachea_seq
666   \seq_clear:N \l__xtable_cacheb_seq
667   \seq_gclear:N \g__xtable_col_wd_seq
668   \prop_gclear:N \g__xtable_cell_ufill_prop
669   \prop_gclear:N \g__xtable_cell_dfill_prop
670   \int_step_inline:nn {\g__xtable_col_count_int} % 自然宽度循环
671   {
672     \_xtable_measure_col_wd:n {##1}
673     \_xtable_parse_col_wd_style:n {##1}
674     \clist_if_in:nVTF {a,s,f,sf} \l__xtable_style_tl
675     {

```

¹⁸本函数需要确认 `\l__xtable_tmpa_dim` 不会被调用的函数修改。

```

676     \seq_gput_right:Ne \g__xtable_col_wd_seq
677     { \dim_use:N \l__xtable_wd_dim }
678     \tl_if_eq:VnT \l__xtable_style_tl {s}
679     {
680         \seq_put_right:Nn \l__xtable_cachea_seq {##1}
681         \__xtable_dim_set_max:NN \l__xtable_cachea_dim \l__xtable_wd_dim
682     }
683     \tl_if_eq:VnT \l__xtable_style_tl {sf}
684     {
685         \seq_put_right:Nn \l__xtable_cacheb_seq {##1}
686         \__xtable_dim_set_max:NN \l__xtable_cacheb_dim \l__xtable_wd_dim
687     }
688     }
689     { \seq_gput_right:NV \g__xtable_col_wd_seq \l__xtable_style_tl }
690     }
691 \seq_map_inline:Nn \l__xtable_cachea_seq % 相同宽度循环
692 {
693     \seq_gset_item:Nne \g__xtable_col_wd_seq
694     {##1} { \dim_use:N \l__xtable_cachea_dim }
695 }
696 \seq_map_inline:Nn \l__xtable_cacheb_seq
697 {
698     \seq_gset_item:Nne \g__xtable_col_wd_seq
699     {##1} { \dim_use:N \l__xtable_cacheb_dim }
700 }
701 \int_zero:N \l__xtable_cachea_int % 填充循环
702 \seq_clear:N \l__xtable_cachea_seq
703 \dim_set:Nn \l__xtable_wd_dim
704 { \seq_item:Nn \g__xtable_col_wd_style_seq {-1} }
705 \seq_map_inline:Nn \g__xtable_col_sep_seq
706 { \dim_sub:Nn \l__xtable_wd_dim {##1} }
707 \int_step_inline:nn {\g__xtable_col_count_int}
708 {
709     \dim_sub:Nn \l__xtable_wd_dim
710     { \seq_item:Nn \g__xtable_col_wd_seq {##1} }
711     \__xtable_parse_col_wd_style:n {##1}
712     \tl_if_in:NnT \l__xtable_style_tl {f}
713     {
714         \int_incr:N \l__xtable_cachea_int
715         \seq_put_right:Nn \l__xtable_cachea_seq {##1}
716     }
717 }

```

```

718 \int_if_zero:nF { \l__xtable_cachea_int }
719 {
720   \dim_set:Nn \l__xtable_cachea_dim
721     { \l__xtable_wd_dim / \l__xtable_cachea_int }
722 }
723 \seq_map_inline:Nn \l__xtable_cachea_seq
724 {
725   \dim_set:Nn \l__xtable_wd_dim
726     { \seq_item:Nn \g__xtable_col_wd_seq {##1} }
727   \dim_add:Nn \l__xtable_wd_dim { \l__xtable_cachea_dim }
728   \seq_gset_item:Nne \g__xtable_col_wd_seq
729     {##1} { \dim_use:N \l__xtable_wd_dim }
730 }
731 }

```

(End of definition for __xtable_calc_col_wd:.)

5.2.3 高度计算

__xtable_measure_cell_ht:n 测量单元格内容在限制宽度内的高度尺寸。#1: 垂直对齐方式，#2: 宽度，#3 内容；变量 \l__xtable_ufill_dim 和 \l__xtable_dfill_dim 用于传递需要垂直填充的值。结果保存在 \l__xtable_ht_dim 和 \l__xtable_dp_dim 中。

```

732 \cs_new:Nn \__xtable_measure_cell_ht:nnn
733 {
734   \dim_set_eq:NN \l__xtable_ht_dim \c__xtable_std_ht_dim
735   \dim_set_eq:NN \l__xtable_dp_dim \c__xtable_std_dp_dim
736   \tl_if_empty:nF {#3}
737   {
738     \str_if_eq:nnTF {#1} {t}
739     { \vbox_set_top:Nn } { \vbox_set:Nn }
740     \l__xtable_tmpa_box
741     {
742       \baselineskip=\g__xtable_cell_lineskip_dim
743       \hsize=#2 \parindent=0pt \noindent #3
744     }
745     \__xtable_dim_set_max:Nn \l__xtable_ht_dim
746     { \box_ht:N \l__xtable_tmpa_box + \l__xtable_ufill_dim }
747     \__xtable_dim_set_max:Nn \l__xtable_dp_dim
748     { \box_dp:N \l__xtable_tmpa_box + \l__xtable_dfill_dim }
749   }
750 }
751 \cs_generate_variant:Nn \__xtable_measure_cell_ht:nnn { VeV }

```

(End of definition for _xtable_measure_cell_ht:n.)

_xtable_measure_row_ht:n 测量表格的行(#1)的高度,结果保存在 \l__xtable_ht_dim 与 \l__xtable_dp_dim 中,同时会缓存高度数据备用。本函数会修改变量 \l__xtable_ufill_dim、\l__xtable_dfill_dim 和 \l__xtable_row_align_tl 的值。¹⁹

```

752 \cs_new:Nn \_xtable_measure_row_ht:n
753 {
754   \dim_zero:N \l__xtable_tmpa_dim
755   \dim_zero:N \l__xtable_tmpb_dim
756   \_xtable_parse_row_align:n {#1}
757   \int_step_inline:nn {\g__xtable_col_count_int}
758   {
759     \_xtable_parse_cell:nn {#1} {##1}
760     \_xtable_parse_cell_fill:nn {#1} {##1}
761     \_xtable_measure_cell_ht:VeV \l__xtable_row_align_tl
762     { \seq_item:Nn \g__xtable_col_wd_seq {##1} }
763     \l__xtable_data_tl
764     \prop_gput:Nne \g__xtable_cell_ht_prop {#1,##1}
765     { \dim_use:N \l__xtable_ht_dim }
766     \_xtable_dim_set_max:NN \l__xtable_tmpa_dim \l__xtable_ht_dim
767     \_xtable_dim_set_max:NN \l__xtable_tmpb_dim \l__xtable_dp_dim
768   }
769   \dim_set_eq:NN \l__xtable_ht_dim \l__xtable_tmpa_dim
770   \dim_set_eq:NN \l__xtable_dp_dim \l__xtable_tmpb_dim
771 }

```

(End of definition for _xtable_measure_row_ht:n.)

_xtable_calc_row_ht: 计算表格各行的高度。本命令会大量修改通用变量的值,调用时需要注意。

```

772 \cs_new:Nn \_xtable_calc_row_ht:
773 {
774   \dim_zero:N \l__xtable_cachea_dim
775   \dim_zero:N \l__xtable_cacheb_dim
776   \seq_clear:N \l__xtable_cachea_seq
777   \seq_gclear:N \g__xtable_row_ht_seq
778   \seq_gclear:N \g__xtable_row_dp_seq
779   \prop_gclear:N \g__xtable_cell_ht_prop
780   \int_step_inline:nnn {0} {\g__xtable_row_count_int} % 自然高度循环
781   {
782     \_xtable_parse_row_ht_style:n {##1}
783     \_xtable_measure_row_ht:n {##1}

```

¹⁹本函数需要确认 \l__xtable_tmpa_dim 与 \l__xtable_tmpb_dim 不会被调用的函数修改。

```

784     \clist_if_in:nVTF {a,s} \l__xtable_style_tl
785     {
786         \seq_gput_right:Ne \g__xtable_row_ht_seq
787         { \dim_use:N \l__xtable_ht_dim }
788         \seq_gput_right:Ne \g__xtable_row_dp_seq
789         { \dim_use:N \l__xtable_dp_dim }
790         \tl_if_eq:VnT \l__xtable_style_tl {s}
791         {
792             \seq_put_right:Nn \l__xtable_cachea_seq {##1}
793             \__xtable_dim_set_max:NN \l__xtable_cachea_dim \l__xtable_ht_dim
794             \__xtable_dim_set_max:NN \l__xtable_cacheb_dim \l__xtable_dp_dim
795         }
796     }
797     {
798         \seq_gput_right:Ne \g__xtable_row_ht_seq
799         { \dim_eval:n { \l__xtable_style_tl - \l__xtable_dp_dim } }
800         \seq_gput_right:Ne \g__xtable_row_dp_seq
801         { \dim_use:N \l__xtable_dp_dim }
802     }
803 }
804 \seq_map_inline:Nn \l__xtable_cachea_seq % 相同高度循环
805 {
806     \__xtable_parse_row_align:n {##1}
807     \str_if_eq:VnTF \l__xtable_row_align_tl {t}
808     {
809         \dim_set:Nn \l__xtable_ht_dim
810         { \seq_item:Nn \g__xtable_row_ht_seq {##1+1} }
811         \dim_set:Nn \l__xtable_dp_dim
812         { \l__xtable_cachea_dim + \l__xtable_cacheb_dim - \l__xtable_ht_dim }
813         \seq_gset_item:Nne \g__xtable_row_dp_seq
814         {##1+1} { \dim_use:N \l__xtable_dp_dim }
815     }
816     {
817         \dim_set:Nn \l__xtable_dp_dim
818         { \seq_item:Nn \g__xtable_row_dp_seq {##1+1} }
819         \dim_set:Nn \l__xtable_ht_dim
820         { \l__xtable_cachea_dim + \l__xtable_cacheb_dim - \l__xtable_dp_dim }
821         \seq_gset_item:Nne \g__xtable_row_ht_seq
822         {##1+1} { \dim_use:N \l__xtable_ht_dim }
823     }
824 }
825 }

```

(End of definition for `_xtable_calc_row_ht:`.)

5.2.4 查询尺寸

`_xtable_parse_cell_size:nn` 查询单元格的排版尺寸。`#1#2` 为行列数字坐标。结果保存在 `\l__xtable_wd_dim`、`\l__xtable_ht_dim` 及 `\l__xtable_dp_dim` 中。

```
826 \cs_new:Nn \_xtable_parse_cell_size:nn
827   {
828     \dim_set:Nn \l__xtable_wd_dim
829       { \seq_item:Nn \g__xtable_col_wd_seq {#2} }
830     \dim_set:Nn \l__xtable_ht_dim
831       { \seq_item:Nn \g__xtable_row_ht_seq {#1+1} }
832     \dim_set:Nn \l__xtable_dp_dim
833       { \seq_item:Nn \g__xtable_row_dp_seq {#1+1} }
834   }
```

(End of definition for `_xtable_parse_cell_size:nn`.)

`_xtable_parse_cell_fill:nn` 查询单元格的首行高度填充值。`#1#2` 为行列数字坐标。结果保存在 `\l__xtable_ufill_dim` 与 `\l__xtable_dfill_dim` 中。

```
835 \cs_new:Nn \_xtable_parse_cell_fill:nn
836   {
837     \prop_get:NnNF \g__xtable_cell_ufill_prop {#1,#2} \l__xtable_tmpa_tl
838     { \tl_set:Nn \l__xtable_tmpa_tl {Opt} }
839     \prop_get:NnNF \g__xtable_cell_dfill_prop {#1,#2} \l__xtable_tmpb_tl
840     { \tl_set:Nn \l__xtable_tmpb_tl {Opt} }
841     \dim_set:Nn \l__xtable_ufill_dim {\l__xtable_tmpa_tl}
842     \dim_set:Nn \l__xtable_dfill_dim {\l__xtable_tmpb_tl}
843   }
```

(End of definition for `_xtable_parse_cell_fill:nn`.)

`_xtable_parse_cell_ht:nn` 查询单元格的实际高度。`#1#2` 为行列数字坐标。结果保存在 `\l__xtable_ht_dim` 中。

```
844 \cs_new:Nn \_xtable_parse_cell_ht:nn
845   {
846     \prop_get:NnN \g__xtable_cell_ht_prop {#1,#2} \l__xtable_tmpa_tl
847     \dim_set:Nn \l__xtable_ht_dim {\l__xtable_tmpa_tl}
848   }
```

(End of definition for `_xtable_parse_cell_ht:nn`.)

5.2.5 刷新坐标

`_xtable_calc_coord:` 计算表格的（边框线所处位置的）实际输出坐标，其受有无 Header 的影响。本命令使用 `\l_xtable_wd_dim` 与 `\l_xtable_ht_dim` 变量。

```

849 \cs_new:Nn \_xtable_calc_coord:
850   {
851     \dim_zero:N \l_xtable_wd_dim
852     \seq_gclear:N \g_xtable_col_loc_seq
853     \seq_gput_right:NV \g_xtable_col_loc_seq \l_xtable_wd_dim
854     \dim_add:Nn \l_xtable_wd_dim
855       { (\seq_item:Nn \g_xtable_col_sep_seq {1})/2 }
856     \int_step_inline:nn {\g_xtable_col_count_int}
857       {
858         \dim_add:Nn \l_xtable_wd_dim
859           { (\seq_item:Nn \g_xtable_col_sep_seq {##1})/2 }
860         \dim_add:Nn \l_xtable_wd_dim
861           { \seq_item:Nn \g_xtable_col_wd_seq {##1} }
862         \dim_add:Nn \l_xtable_wd_dim
863           { (\seq_item:Nn \g_xtable_col_sep_seq {##1+1})/2 }
864         \seq_gput_right:NV \g_xtable_col_loc_seq \l_xtable_wd_dim
865       }
866     \dim_add:Nn \l_xtable_wd_dim
867       { (\seq_item:Nn \g_xtable_col_sep_seq {-1})/2 }
868     \seq_gset_item:NnV \g_xtable_col_loc_seq {-1} \l_xtable_wd_dim

```

为方便计算，Y 坐标采用负坐标。

```

869     \dim_zero:N \l_xtable_ht_dim
870     \seq_gclear:N \g_xtable_row_loc_seq
871     \seq_gput_right:NV \g_xtable_row_loc_seq \l_xtable_ht_dim
872     \bool_if:NTF \g_xtable_has_header_bool
873       { \int_set:Nn \l_xtable_tmpa_int {0} }
874       { \int_set:Nn \l_xtable_tmpa_int {1} }
875     \dim_sub:Nn \l_xtable_ht_dim
876       { \seq_item:Nn \g_xtable_row_sep_seq {1} }
877     \int_step_inline:nnn {\l_xtable_tmpa_int} {\g_xtable_row_count_int}
878       {
879         \int_compare:nNnF {##1} = {\l_xtable_tmpa_int}
880           {
881             \dim_sub:Nn \l_xtable_ht_dim
882               { (\seq_item:Nn \g_xtable_row_sep_seq {##1+1})/2 }
883           }
884         \dim_sub:Nn \l_xtable_ht_dim
885           { \seq_item:Nn \g_xtable_row_ht_seq {##1+1} }

```

```

886     \dim_sub:Nn \l__xtable_ht_dim
887     { \seq_item:Nn \g__xtable_row_dp_seq {##1+1} }
888     \dim_sub:Nn \l__xtable_ht_dim
889     { (\seq_item:Nn \g__xtable_row_sep_seq {##1+2})/2 }
890     \seq_gput_right:NV \g__xtable_row_loc_seq \l__xtable_ht_dim
891   }
892   \dim_sub:Nn \l__xtable_ht_dim
893   { (\seq_item:Nn \g__xtable_row_sep_seq {-1})/2 }
894   \seq_gset_item:NnV \g__xtable_row_loc_seq {-1} \l__xtable_ht_dim
895 }

```

(End of definition for __xtable_calc_coord:.)

6 输入接口

6.1 变量与选项

6.1.1 常量定义

`\c__xtable_csv_row_regex` 用于解析 CSV 中的行的正则表达式。其中单元格表达式每个匹配生成三个项目，它们分别是所有匹配，常规字段值，引号字段值。

`\c__xtable_csv_cell_regex`

```

896 \regex_const:Nn \c__xtable_csv_row_regex
897 { (?:(?:[^\c{\}\}]|"(?:"|[^\"])+")+)(?:\c{\}\}) }
898 \regex_const:Nn \c__xtable_csv_cell_regex
899 { (?:([^\c{\}\},]*)|"(?:"|[^\"])+")(?:,|\c{\}\}) }

```

(End of definition for \c__xtable_csv_row_regex and \c__xtable_csv_cell_regex.)

`\c__xtable_json_cell_regex` 用于解析 JSON 中的属性值的正则表达式。每个匹配生成四个项目，它们分别是所有匹配，名称，非字符值（数值或真假），字符值。

```

900 \regex_const:Nn \c__xtable_json_cell_regex
901 {
902   "((?:[^\\"\\\]|\\.)+)"
903   \s*:\s*
904   (?:
905     ([\+|-]?(?:\d*\.)?\d+|true|false) |
906     "((?:[^\\"\\\]|\\.)+)"
907   )
908   \s*[,\]?
909 }

```

(End of definition for \c__xtable_json_cell_regex.)

```

\c__xtable_row_ht_regex 用于判定某项内容是否为行高样式/列宽样式。
\c__xtable_col_wd_regex 910 \regex_const:Nn \c__xtable_row_ht_regex
911 {
912   \A(?:.*=)?(?:
913     (auto|same|a|s)|
914     ([0-9.]+(?:pt|ex|em|mm|cm|in|cc|dd|pc))
915   )\Z
916 }
917 \regex_const:Nn \c__xtable_col_wd_regex
918 {
919   \A(?:.*=)?(?:
920     (auto|same|fill|samefill|a|s|f|sf)|
921     ([0-9.]+(?:pt|ex|em|mm|cm|in|cc|dd|pc))
922   )\Z
923 }
(End of definition for \c__xtable_row_ht_regex and \c__xtable_col_wd_regex.)

```

6.1.2 状态参数

```

\l__xtable_input_bool 当前是否处于输入环境。
924 \bool_new:N \l__xtable_input_bool
925 \bool_set_false:N \l__xtable_input_bool
(End of definition for \l__xtable_input_bool.)

\_xtable_check_input_env:n 检测是否处理输入环境。如果不是，则报错。
926 \cs_new:Nn \_xtable_check_input_env:n
927 {
928   \bool_if:NF \l__xtable_input_bool
929     { \msg_error:nnn {xtable} {outside_table} {#1} }
930 }
(End of definition for \_xtable_check_input_env:n.)

\l__xtable_parse_status_int 用于保存解析过程中的状态。
\l__xtable_esc_status_bool 931 \int_new:N \l__xtable_parse_status_int
\l__xtable_quote_status_bool 932 \bool_new:N \l__xtable_esc_status_bool
933 \bool_new:N \l__xtable_quote_status_bool
(End of definition for \l__xtable_parse_status_int, \l__xtable_esc_status_bool, and \l__xtable_quote_
status_bool.)

```

6.1.3 局部变量

`\l__xtable_row_input_seq` 用于保存原始的输入数据。

```
\l__xtable_cell_input_seq 934 \seq_new:N \l__xtable_row_input_seq
935 \seq_new:N \l__xtable_cell_input_seq
```

(End of definition for `\l__xtable_row_input_seq` and `\l__xtable_cell_input_seq`.)

6.1.4 局部选项

`\l__xtable_input_format_tl` 输入选项变量。

```
\l__xtable_input_sep_tl 936 \tl_new:N \l__xtable_input_format_tl
\l__xtable_input_has_header_bool 937 \tl_new:N \l__xtable_input_sep_tl
938 \bool_new:N \l__xtable_input_has_header_bool
```

(End of definition for `\l__xtable_input_format_tl`, `\l__xtable_input_sep_tl`, and `\l__xtable_input_has_header_bool`.)

`xtable/input` 输入选项。

```
939 \keys_define:nn { xtable / input }
940 {
941   format .choices:nn =
942     { inner, csv, json }
943     { \tl_set_eq:NN \l__xtable_input_format_tl \l_keys_choice_tl },
944   format .initial:n = {inner},
945   title .bool_set:N = \l__xtable_input_has_header_bool,
946   header .bool_set:N = \l__xtable_input_has_header_bool,
947   header .initial:n = {true},
948   sep .tl_set:N = \l__xtable_input_sep_tl,
949   sep .initial:n = {,},
950   loc .code:n = {\__xtable_parse_coord:n {#1}},
951   loc .initial:n = {1,1}
952 }
```

(End of definition for `xtable/input`. This function is documented on page ??.)

6.2 数据解析

6.2.1 标准格式

`__xtable_parse_input:n` 解析标准输入的表格数据 (#1)，并将其添加到指定位置。起始行由 `\l__xtable_row_loc_int` 指定；起始列由 `\l__xtable_col_loc_int` 指定。

```
953 \cs_new:Nn \__xtable_parse_input:n
954 {
```

```

955 \seq_set_split:Nnn \l__xtable_row_input_seq {\} {#1}
956 \int_set_eq:NN \l__xtable_tmpa_int \l__xtable_row_loc_int
957 \seq_map_inline:Nn \l__xtable_row_input_seq
958 {
959   \tl_set:Nn \l__xtable_tmpa_tl {##1}
960   \tl_replace_all:Nnn \l__xtable_tmpa_tl {\newline} {}
961   \int_set_eq:NN \l__xtable_tmpb_int \l__xtable_col_loc_int
962   \seq_set_split:NVV \l__xtable_cell_input_seq
963     \l__xtable_input_sep_tl \l__xtable_tmpa_tl
964   \bool_if:NTF \l__xtable_input_has_header_bool
965     { % 填充标题
966       \seq_map_inline:Nn \l__xtable_cell_input_seq
967         {
968           \__xtable_set_col_name:nN {###1} \l__xtable_tmpb_int
969           \int_incr:N \l__xtable_tmpb_int
970         }
971       \bool_set_false:N \l__xtable_input_has_header_bool
972     }
973     { % 填充数据
974       \seq_map_inline:Nn \l__xtable_cell_input_seq
975         {
976           \__xtable_set_cell:NNn \l__xtable_tmpa_int \l__xtable_tmpb_int {###1}
977           \int_incr:N \l__xtable_tmpb_int
978         }
979       \int_incr:N \l__xtable_tmpa_int
980     }
981   }
982 }

```

(End of definition for `__xtable_parse_input:n`.)

6.2.2 CSV 数据

解析 CSV 数据使用的思路是，先用正则表达式解析出行，再用正则表达式解析每个字段。

`__xtable_parse_csv:n` 解析 CSV 数据 (#1)，并将其添加到指定位置。起始行由 `\l__xtable_row_loc_int` 指定；起始列由 `\l__xtable_col_loc_int` 指定。

```

983 \cs_new:Nn \__xtable_parse_csv:n
984 {
985   \seq_clear:N \l__xtable_row_input_seq
986   \regex_extract_all:NnN
987     \c__xtable_csv_row_regex {#1\} \l__xtable_row_input_seq

```

```

988 \int_set_eq:NN \l__xtable_tmpa_int \l__xtable_row_loc_int
989 \seq_map_inline:Nn \l__xtable_row_input_seq
990 {
991   \int_set_eq:NN \l__xtable_tmpb_int \l__xtable_col_loc_int
992   \seq_clear:N \l__xtable_tmpa_seq
993   \seq_clear:N \l__xtable_cell_input_seq
994   \regex_extract_all:NnN
995   \c__xtable_csv_cell_regex {##1} \l__xtable_tmpa_seq
996   \int_step_inline:nnnn {2} {3} {\seq_count:N \l__xtable_tmpa_seq}
997   {
998     \tl_set:Ne \l__xtable_tmpa_tl
999     { \seq_item:Nn \l__xtable_tmpa_seq {####1} }
1000    \tl_set:Ne \l__xtable_tmpb_tl
1001    { \seq_item:Nn \l__xtable_tmpa_seq {####1+1} }
1002    \tl_put_right:NV \l__xtable_tmpa_tl \l__xtable_tmpb_tl
1003    \seq_put_right:NV \l__xtable_cell_input_seq \l__xtable_tmpa_tl
1004  }
1005  \bool_if:NTF \l__xtable_input_has_header_bool
1006  { % 填充表头 (设置列名)
1007    \seq_map_inline:Nn \l__xtable_cell_input_seq
1008    {
1009      \__xtable_set_col_name:nN {####1} \l__xtable_tmpb_int
1010      \int_incr:N \l__xtable_tmpb_int
1011    }
1012    \bool_set_false:N \l__xtable_input_has_header_bool
1013  }
1014  { % 填充数据
1015    \seq_map_inline:Nn \l__xtable_cell_input_seq
1016    {
1017      \__xtable_set_cell:NNn \l__xtable_tmpa_int \l__xtable_tmpb_int {####1}
1018      \int_incr:N \l__xtable_tmpb_int
1019    }
1020    \int_incr:N \l__xtable_tmpa_int
1021  }
1022  }
1023 }

```

(End of definition for __xtable_parse_csv:n.)

6.2.3 JSON 数据

本宏包解析 JSON 数据的思路：先将行依次解析到一个中转序列（逐字符处理）再依次解析每一行数据（正则表达式）。

逐字符解析完整 JSON，并将解析状态分为：初始模式、行间模式、行内模式。整个解析过程都在这三个状态中切换。

`__xtable_parse_json_auxa:n` 拆分行时用到的辅助函数，处理初始模式。当前字符（#1）为 [，则表示数据开始，切换到行间模式；当前字符为 {，则表示一行数据开始，直接切换到行内模式。

```

1024 \cs_new:Nn \__xtable_parse_json_auxa:n
1025 {
1026   \str_case_e:nnF {#1}
1027   {
1028     {\c__xtable_space_str} {} % 空白，跳过
1029     {[ ]} % 数组开始
1030     { \int_set:Nn \l__xtable_parse_status_int {1} }
1031     {\c__xtable_lbrace_str} % 行开始
1032     {
1033       \int_set:Nn \l__xtable_parse_status_int {2}
1034       \str_clear:N \l__xtable_tmpa_str
1035     }
1036   }
1037   { \msg_error:nnn {xtable} {unknown_input} {#1} }
1038 }

```

(End of definition for `__xtable_parse_json_auxa:n`.)

`__xtable_parse_json_auxb:n` 拆分行时用到的辅助函数，处理行间模式。当前字符（#1）为]，则表示数据结束，切换回初始模式；当前字符为 {，则表示一行数据开始，切换到行内模式。忽略逗号与空白。

```

1039 \cs_new:Nn \__xtable_parse_json_auxb:n
1040 {
1041   \str_case_e:nnF {#1}
1042   {
1043     {\c__xtable_space_str} {} % 空白，跳过
1044     {,} {} % 行分隔，跳过
1045     [ ]} % 数组结束
1046     { \int_set:Nn \l__xtable_parse_status_int {0} }
1047     {\c__xtable_lbrace_str} % 行开始
1048     {
1049       \int_set:Nn \l__xtable_parse_status_int {2}
1050       \str_clear:N \l__xtable_tmpa_str
1051     }
1052   }
1053   { \msg_error:nnn {xtable} {unknown_input} {#1} }
1054 }

```

(End of definition for `_xtable_parse_json_auxb:n`.)

`_xtable_parse_json_auxc:n` 拆分行时用到的辅助函数，处理行内模式。`#1` 为当前字符。行内模式又有两个子状态：转义态与引用态。如果当前状态为转义态，则恢复为非转义态；否则，可以用 `\` 切换为转义态。如果当前状态为引用态（且非转义态），则可用 `"` 切换为非引用态；否则，可以使用 `"` 切换为引用态，或使用 `}` 结束当前行，并进入行间模式。

```

1055 \cs_new:Nn \_xtable_parse_json_auxc:n
1056 {
1057   \bool_if:NTF \l__xtable_esc_status_bool % 转义状态
1058     { \bool_set_false:N \l__xtable_esc_status_bool }
1059     {
1060       \str_if_eq:VnTF \c__xtable_escape_str {#1} % 即将转义
1061         { \bool_set_true:N \l__xtable_esc_status_bool }
1062         {
1063           \bool_if:NTF \l__xtable_quote_status_bool % 引用状态
1064             {
1065               \str_if_eq:nnT {"} {#1} % 结束引用
1066                 { \bool_set_false:N \l__xtable_quote_status_bool }
1067             }
1068             {
1069               \str_if_eq:nnTF {"} {#1} % 即将引用
1070                 { \bool_set_true:N \l__xtable_quote_status_bool }
1071                 {
1072                   \str_if_eq:VnT \c__xtable_rbrace_str {#1} % 行结束
1073                     {
1074                       \int_set:Nn \l__xtable_parse_status_int {1}
1075                       \seq_put_right:NV \l__xtable_row_input_seq \l__xtable_tmpa_str
1076                     }
1077                   }
1078                 }
1079             }
1080         }
1081       \str_put_right:Nn \l__xtable_tmpa_str {#1}
1082     }

```

(End of definition for `_xtable_parse_json_auxc:n`.)

`_xtable_parse_json_row:n` 解析 JSON 中的单行数据 (`#1`)，并将其添加表中。使用 `\l__xtable_row_loc_int` 与 `\l__xtable_col_loc_int` 来传递下一位置的坐标。使用缓存变量 `\l__xtable_cachea_tl` 与 `\l__xtable_cacheb_tl`。

```

1083 \cs_new:Nn \_xtable_parse_json_row:n
1084 {

```



```

1085 \regex_extract_all:NnN
1086 \c__xtable_json_cell_regex {#1} \l__xtable_cell_input_seq
1087 \bool_until_do:nn
1088 { \seq_if_empty_p:N \l__xtable_cell_input_seq }
1089 {
1090 \seq_pop_left:NN \l__xtable_cell_input_seq \l__xtable_tmpa_tl % 丢弃
1091 \seq_pop_left:NN \l__xtable_cell_input_seq \l__xtable_cachea_tl
1092 \seq_pop_left:NN \l__xtable_cell_input_seq \l__xtable_tmpa_tl
1093 \seq_pop_left:NN \l__xtable_cell_input_seq \l__xtable_tmpb_tl
1094 \tl_if_empty:NTF \l__xtable_tmpa_tl
1095 { \tl_set_eq:NN \l__xtable_cacheb_tl \l__xtable_tmpb_tl }
1096 { \tl_set_eq:NN \l__xtable_cacheb_tl \l__xtable_tmpa_tl }
1097 \int_set_eq:NN \l__xtable_cachea_int \l__xtable_col_loc_int
1098 \__xtable_parse_new_col_loc:V \l__xtable_cachea_tl
1099 \int_set_eq:NN \l__xtable_tmpa_int \l__xtable_col_loc_int
1100 \int_compare:nNnTF {\l__xtable_cachea_int} = {\l__xtable_col_loc_int}
1101 { \int_incr:N \l__xtable_col_loc_int }
1102 { \int_set_eq:NN \l__xtable_col_loc_int \l__xtable_cachea_int }
1103 \__xtable_set_cell:NNV
1104 \l__xtable_row_loc_int \l__xtable_tmpa_int \l__xtable_cacheb_tl
1105 }
1106 \int_incr:N \l__xtable_row_loc_int
1107 }

```

(End of definition for __xtable_parse_json_row:n.)

__xtable_parse_json:nnn 解析 JSON 数据 (#1)，并将其添加到指定位置。起始行由 \l__xtable_row_loc_int 指定；列位置则由列名指定。如果列名不存在，先依次将列名映射到指定位置 (\l__xtable_col_loc_int) 及其后。

```

1108 \cs_new:Nn \__xtable_parse_json:n
1109 {
1110 \int_zero:N \l__xtable_parse_status_int
1111 \bool_set_false:N \l__xtable_esc_status_bool
1112 \bool_set_false:N \l__xtable_quote_status_bool
1113 \str_clear:N \l__xtable_tmpa_str
1114 \seq_clear:N \l__xtable_row_input_seq
1115 \str_map_inline:nn {#1} % 解析 JSON 中的行
1116 {
1117 \int_case:nn
1118 { \l__xtable_parse_status_int }
1119 {
1120 {0} { \__xtable_parse_json_auxa:n {##1} } % 开始
1121 {1} { \__xtable_parse_json_auxb:n {##1} } % 行间状态

```

```

1122         {2} { \_xtable_parse_json_auxc:n {##1} } % 行内状态
1123     }
1124 }
1125 \seq_map_function:NN \l__xtable_row_input_seq \_xtable_parse_json_row:n
1126 }
1127 \cs_generate_variant:Nn \_xtable_parse_json:nnn {een}

(End of definition for \_xtable_parse_json:nnn.)

```

6.3 数据录入

`\loadtable` 保存与加载表格。

```

\loadtable 1128 \NewDocumentCommand {\loadtable} { m }
1129 {
1130     \_xtable_check_input_env:n {\loadtable}
1131     \_xtable_table_restore:n {#1}
1132 }
\loadtable 1133 \NewDocumentCommand {\savetable} { m }
1134 {
1135     \_xtable_check_input_env:n {\savetable}
1136     \_xtable_table_save:n {#1}
1137 }

```

(End of definition for `\loadtable` and `\savetable`. These functions are documented on page 6.)

`\excelcolname` 设置 Excel 风格列名。

```

1138 \NewDocumentCommand {\excelcolname} { O{26} }
1139 {
1140     \_xtable_check_input_env:n {\excelcolname}
1141     \_xtable_set_excel_col_names:n {#1}
1142 }

```

(End of definition for `\excelcolname`. This function is documented on page 4.)

`\rowname` 设置行名与列名。使用变量 `\l__xtable_row_loc_int` 与 `\l__xtable_col_loc_int`。
`\colname` `int`。

```

1143 \NewDocumentCommand {\rowname} { D(){} O{,} m }
1144 {
1145     \_xtable_check_input_env:n {\rowname}
1146     \str_if_empty:nTF {#1}
1147     { \int_set:Nn \l__xtable_row_loc_int {1} }
1148     { \int_set:Nn \l__xtable_row_loc_int {#1} }
1149     \seq_set_split:Nnn \l__xtable_cell_input_seq {#2} {#3}
1150     \seq_map_inline:Nn \l__xtable_cell_input_seq

```

```

1151     {
1152         \_xtable_set_row_name:nN {##1} \l__xtable_row_loc_int
1153         \int_incr:N \l__xtable_row_loc_int
1154     }
1155 }
1156 \NewDocumentCommand {\colname} { D(){} 0{,} m }
1157 {
1158     \_xtable_check_input_env:n {\colname}
1159     \str_if_empty:nTF {#1}
1160     { \int_set:Nn \l__xtable_col_loc_int {1} }
1161     { \int_set:Nn \l__xtable_col_loc_int {#1} }
1162     \seq_set_split:Nnn \l__xtable_cell_input_seq {#2} {#3}
1163     \seq_map_inline:Nn \l__xtable_cell_input_seq
1164     {
1165         \_xtable_set_col_name:nN {##1} \l__xtable_col_loc_int
1166         \int_incr:N \l__xtable_col_loc_int
1167     }
1168 }

```

(End of definition for `\rowname` and `\colname`. These functions are documented on page 4.)

`\row` 设置某一行数据。使用变量 `\l__xtable_row_loc_int` 与 `\l__xtable_col_loc_int`。
`_xtable_row:nnn` **int**。

```

1169 \NewDocumentCommand {\_xtable_row:nnn} { D(){} 0{,} m }
1170 {
1171     \str_if_empty:nTF {#1}
1172     {
1173         \int_set:Nn \l__xtable_row_loc_int { \g__xtable_row_count_int + 1}
1174         \int_set:Nn \l__xtable_col_loc_int {1}
1175     }
1176     {
1177         \str_if_in:nnTF {#1} {,}
1178         {
1179             \int_zero:N \l__xtable_row_loc_int
1180             \int_zero:N \l__xtable_col_loc_int
1181             \_xtable_parse_new_coord:n {#1}
1182         }
1183         {
1184             \int_zero:N \l__xtable_row_loc_int
1185             \int_set:Nn \l__xtable_col_loc_int {1}
1186             \_xtable_parse_new_row_loc:n {#1}
1187         }
1188     }

```

```

1189 \seq_set_split:Nnn \l__xtable_cell_input_seq {#2} {#3}
1190 \seq_map_inline:Nn \l__xtable_cell_input_seq
1191 {
1192   \__xtable_set_cell:n {##1}
1193   \int_incr:N \l__xtable_col_loc_int
1194 }
1195 }

```

(End of definition for `\row` and `__xtable_row:nnn`. This function is documented on page 6.)

`\col` 设置某一列数据。使用变量 `\l__xtable_row_loc_int` 与 `\l__xtable_col_loc_`
`__xtable_col:nnn` `int`。

```

1196 \NewDocumentCommand {\__xtable_col:nnn} { D(){} O{,} m }
1197 {
1198   \str_if_empty:nTF {#1}
1199   {
1200     \int_set:Nn \l__xtable_row_loc_int {1}
1201     \int_set:Nn \l__xtable_col_loc_int { \g__xtable_col_count_int + 1}
1202   }
1203   {
1204     \str_if_in:nnTF {#1} {,}
1205     {
1206       \int_zero:N \l__xtable_row_loc_int
1207       \int_zero:N \l__xtable_col_loc_int
1208       \__xtable_parse_new_coord:n {#1}
1209     }
1210     {
1211       \int_set:Nn \l__xtable_row_loc_int {1}
1212       \int_zero:N \l__xtable_col_loc_int
1213       \__xtable_parse_new_col_loc:n {#1}
1214     }
1215   }
1216   \seq_set_split:Nnn \l__xtable_cell_input_seq {#2} {#3}
1217   \seq_map_inline:Nn \l__xtable_cell_input_seq
1218   {
1219     \__xtable_set_cell:n {##1}
1220     \int_incr:N \l__xtable_row_loc_int
1221   }
1222 }

```

(End of definition for `\col` and `__xtable_col:nnn`. This function is documented on page 6.)

`\cell` 用于输入单元格的数据。使用变量 `\l__xtable_row_loc_int` 与 `\l__xtable_col_`
`__xtable_cell:nn` `loc_int`。

```

1223 \NewDocumentCommand {\__xtable_cell:nn} { r() m}
1224 {
1225   \int_zero:N \l__xtable_row_loc_int
1226   \int_zero:N \l__xtable_col_loc_int
1227   \__xtable_parse_new_coord:n {#1}
1228   \__xtable_set_cell:n {#2}
1229 }

```

(End of definition for `\cell` and `__xtable_cell:nn`. This function is documented on page 6.)

`data (env.)` 用于输入表格内容的环境。

```

@@_data: (env.) 1230 \NewDocumentEnvironment{\__xtable_data:}{0}{ +b}
1231 {
1232   \keys_set:nn { xtable/input } {#1}
1233   \str_case:VnF \l__xtable_input_format_tl
1234   {
1235     {inner} { \__xtable_parse_input:n {#2} }
1236     {csv}   { \__xtable_parse_csv:n   {#2} }
1237     {json}  { \__xtable_parse_json:n  {#2} }
1238   }
1239   { \msg_error:nnV {xtable} {unknown_format} \l__xtable_input_format_tl }
1240 }
1241 {}

```

`xtable (env.)` 用于输入表格的环境。

```

1242 \NewDocumentEnvironment{xtable}{0}{ }
1243 {
1244   \__xtable_table_init:
1245   \keys_set:nn { xtable/input } {#1}
1246   \bool_gset_eq:NN
1247     \g__xtable_has_header_bool \l__xtable_input_has_header_bool
1248   \cs_set_eq:NN \data \__xtable_data:
1249   \cs_set_eq:NN \enddata \end__xtable_data:
1250   \cs_set_eq:NN \row \__xtable_row:nnn
1251   \cs_set_eq:NN \col \__xtable_col:nnn
1252   \cs_set_eq:NN \cell \__xtable_cell:nn
1253   \bool_set_true:N \l__xtable_input_bool
1254 }
1255 {
1256   \bool_set_false:N \l__xtable_input_bool
1257   \__xtable_insure_style:
1258   \__xtable_process_header:
1259 }

```

6.4 格式设置

`\rowheight` 设置行高与列宽。使用变量 `\l__xtable_shared_seq`。

```

\colwidth 1260 \NewDocumentCommand {\rowheight} { O{auto} m }
1261 {
1262   \__xtable_check_input_env:n {\rowheight}
1263   \tl_if_empty:nTF {#2}
1264     { \seq_clear:N \l__xtable_shared_seq }
1265     { \seq_set_split:Nnn \l__xtable_shared_seq {,} {#2} }
1266   \regex_extract_all:NnNF
1267     \c__xtable_row_ht_regex {#1} \l__xtable_tmpa_seq
1268     { \msg_error:nnn {xtable} {unknown_format} {#1} }
1269   \seq_map_inline:Nn \l__xtable_shared_seq
1270     {
1271       \regex_extract_all:NnNF
1272         \c__xtable_row_ht_regex {##1} \l__xtable_tmpa_seq
1273         { \msg_error:nnn {xtable} {unknown_format} {##1} }
1274     }
1275   \__xtable_set_row_ht_style:Nn \l__xtable_shared_seq {#1}
1276 }

1277 \NewDocumentCommand {\colwidth} { O{auto} O{\textwidth} m }
1278 {
1279   \__xtable_check_input_env:n {\colwidth}
1280   \tl_if_empty:nTF {#3}
1281     { \seq_clear:N \l__xtable_shared_seq }
1282     { \seq_set_split:Nnn \l__xtable_shared_seq {,} {#3} }
1283   \regex_extract_all:NnNF
1284     \c__xtable_col_wd_regex {#1} \l__xtable_tmpa_seq
1285     { \msg_error:nnn {xtable} {unknown_format} {#1} }
1286   \seq_map_inline:Nn \l__xtable_shared_seq
1287     {
1288       \regex_extract_all:NnNF
1289         \c__xtable_col_wd_regex {##1} \l__xtable_tmpa_seq
1290         { \msg_error:nnn {xtable} {unknown_format} {##1} }
1291     }
1292   \__xtable_set_col_wd_style:Nnn \l__xtable_shared_seq {#1} {#2}
1293 }

```

(End of definition for `\rowheight` and `\colwidth`. These functions are documented on page 6.)

`\rowalign` 设置行与列的对齐方式。使用变量 `\l__xtable_shared_seq`。

```

\colalign 1294 \NewDocumentCommand {\rowalign} { O{b} m }
1295 {

```

```

1296 \l__xtable_check_input_env:n {\rowalign}
1297 \regex_extract_all:nnNF
1298   {^(?:(:?\.*)?[tmb],?)*$} {#1#2} \l__xtable_tmpa_seq
1299   { \msg_error:nnn {xtable} {unknown_format} {[#1],[#2]} }
1300 \str_if_in:nnTF {#2} {=}
1301   { \seq_set_split:Nnn \l__xtable_shared_seq {,} {#2} }
1302   {
1303     \tl_set:Nn \l__xtable_tmpa_tl {#2}
1304     \tl_replace_all:Nnn \l__xtable_tmpa_tl {,} {}
1305     \seq_set_split:NnV \l__xtable_shared_seq {} \l__xtable_tmpa_tl
1306   }
1307 \l__xtable_set_row_align:Nn \l__xtable_shared_seq {#1}
1308 }

1309 \NewDocumentCommand {\colalign} { O{c} m }
1310 {
1311   \l__xtable_check_input_env:n {\colalign}
1312   \regex_extract_all:nnNF
1313     {^(?:(:?\.*)?[lcr],?)*$} {#1#2} \l__xtable_tmpa_seq
1314     { \msg_error:nnn {xtable} {unknown_format} {[#1],[#2]} }
1315   \str_if_in:nnTF {#2} {=}
1316     { \seq_set_split:Nnn \l__xtable_shared_seq {,} {#2} }
1317     {
1318       \tl_set:Nn \l__xtable_tmpa_tl {#2}
1319       \tl_replace_all:Nnn \l__xtable_tmpa_tl {,} {}
1320       \seq_set_split:NnV \l__xtable_shared_seq {} \l__xtable_tmpa_tl
1321     }
1322   \l__xtable_set_col_align:Nn \l__xtable_shared_seq {#1}
1323 }

```

(End of definition for `\rowalign` and `\colalign`. These functions are documented on page 6.)

7 渲染输出

7.1 通用内容

7.1.1 局部变量

`\l__xtable_expand_start_dim` 两个扩展值变量。

```
\l__xtable_expand_end_dim 1324 \dim_new:N \l__xtable_expand_start_dim
```

```
1325 \dim_new:N \l__xtable_expand_end_dim
```

(End of definition for `\l__xtable_expand_start_dim` and `\l__xtable_expand_end_dim`.)

`\l__xtable_offset_x_dim` 两个偏移值变量。

```
\l__xtable_offset_y_dim 1326 \dim_new:N \l__xtable_offset_x_dim
1327 \dim_new:N \l__xtable_offset_y_dim

(End of definition for \l__xtable_offset_x_dim and \l__xtable_offset_y_dim.)
```

7.1.2 渲染函数

`__xtable_draw_hline:n` 绘制表格指定的水平线。使用 `\l__xtable_expand_start_dim` 与 `\l__xtable_expand_end_dim` 扩展边界。

```
1328 \cs_new:Nn \__xtable_draw_hline:n
1329 {
1330   \draw_path_moveto:n
1331   {
1332     \seq_item:Nn \g__xtable_col_loc_seq {1} - \l__xtable_expand_start_dim,
1333     \seq_item:Nn \g__xtable_row_loc_seq {#1}
1334   }
1335   \draw_path_lineto:n
1336   {
1337     \seq_item:Nn \g__xtable_col_loc_seq {-1} + \l__xtable_expand_end_dim,
1338     \seq_item:Nn \g__xtable_row_loc_seq {#1}
1339   }
1340 }

(End of definition for \__xtable_draw_hline:n.)
```

`__xtable_draw_vline:n` 绘制表格指定的垂直线。使用 `\l__xtable_expand_start_dim` 与 `\l__xtable_expand_end_dim` 扩展边界。

```
1341 \cs_new:Nn \__xtable_draw_vline:n
1342 {
1343   \draw_path_moveto:n
1344   {
1345     \seq_item:Nn \g__xtable_col_loc_seq {#1},
1346     \seq_item:Nn \g__xtable_row_loc_seq {1} + \l__xtable_expand_start_dim
1347   }
1348   \draw_path_lineto:n
1349   {
1350     \seq_item:Nn \g__xtable_col_loc_seq {#1},
1351     \seq_item:Nn \g__xtable_row_loc_seq {-1} - \l__xtable_expand_end_dim
1352   }
1353 }

(End of definition for \__xtable_draw_vline:n.)
```


7.2 打印表格

`_xtable_print_data:nnn` 使用垂直盒子打印输出指定数据。#1: 垂直对齐方式, #2: 水平对齐方式, #3: 内容。使用变量 `\l__xtable_wd_dim`、`\l__xtable_ht_dim` 和 `\l__xtable_dp_dim` 的值。

```

1354 \cs_new:Nn \_xtable_print_data:nnn
1355 {
1356   \str_if_in:nnTF {tmb} {#1}
1357   { \tl_set:Nn \l__xtable_tmpb_tl {#1} }
1358   { \tl_set:Nn \l__xtable_tmpb_tl {b} }
1359   \str_case:nnF {#2}
1360   {
1361     {l} { \tl_set:Nn \l__xtable_tmpa_tl {\raggedright} }
1362     {c} { \tl_set:Nn \l__xtable_tmpa_tl {\centering} }
1363     {r} { \tl_set:Nn \l__xtable_tmpa_tl {\raggedleft} }
1364   }
1365   { \tl_set:Nn \l__xtable_tmpa_tl {} }
1366   \tl_if_empty:nT {#3}
1367   { \tl_put_right:Nn \l__xtable_tmpa_tl {\rule{1pt}{0pt}} }
1368   \str_case:Vn \l__xtable_tmpb_tl
1369   {
1370     {t}
1371     {
1372       \vbox_set_top:Nn \l__xtable_tmpa_box
1373       {
1374         \baselineskip=\g__xtable_cell_lineskip_dim
1375         \hsize=\l__xtable_wd_dim \parindent=0pt
1376         \l__xtable_tmpa_tl #3 \vfill \par
1377       }
1378     }

```

如果是居中对齐, 则额外使用 `\l__xtable_cachea_dim` 与 `\l__xtable_ufill_dim` 传递单元格高度及填充量。

```

1379     {m}
1380     {
1381       \dim_set:Nn \l__xtable_tmpa_dim
1382       { (\l__xtable_ht_dim - \l__xtable_cachea_dim)/2 + \l__xtable_ufill_dim }
1383       \vbox_set_to_ht:Nnn \l__xtable_tmpa_box {\l__xtable_ht_dim}
1384       {
1385         \baselineskip=\g__xtable_cell_lineskip_dim
1386         \hsize=\l__xtable_wd_dim \parindent=0pt
1387         \vskip \l__xtable_tmpa_dim
1388         \l__xtable_tmpa_tl #3 \vfill \par
1389       }

```

```

1390     }
1391     {b}
1392     {
1393         \dim_set:Nn \l__xtable_tmpa_dim
1394             { \l__xtable_ht_dim - \l__xtable_cachea_dim + \l__xtable_ufill_dim }
1395         \vbox_set:Nn \l__xtable_tmpa_box
1396             {
1397                 \baselineskip=\g__xtable_cell_lineskip_dim
1398                 \hsize=\l__xtable_wd_dim \parindent=0pt
1399                 \vskip \l__xtable_tmpa_dim
1400                 \l__xtable_tmpa_tl #3 \par
1401             }
1402     }
1403 }
1404 \dim_compare:nNnT {\box_dp:N \l__xtable_tmpa_box} < {\l__xtable_dp_dim}
1405     { \box_set_dp:Nn \l__xtable_tmpa_box {\l__xtable_dp_dim} }
1406 \box_use:N \l__xtable_tmpa_box
1407 }
1408 \cs_generate_variant:Nn \__xtable_print_data:nnn {VVV}

```

(End of definition for __xtable_print_data:nnn.)

`\printtable` 打印表格。

```

1409 \NewDocumentCommand \printtable { s O{0em} }
1410 {
1411     \__xtable_set_col_sep:ne {0pt} { \dim_use:N \g__xtable_col_sep_dim }
1412     \__xtable_calc_col_wd: \__xtable_calc_row_ht:
1413     \bool_if:NTF \g__xtable_has_header_bool
1414         { \int_set:Nn \l__xtable_cachea_int {1} }
1415         { \int_set:Nn \l__xtable_cachea_int {0} }
1416     \int_step_inline:nnn {1-\l__xtable_cachea_int} {\g__xtable_row_count_int}
1417     {
1418         \hbox:n
1419         {
1420             \dim_zero:N \l__xtable_tmpa_dim
1421             \int_step_inline:nn {\g__xtable_col_count_int}
1422             {
1423                 \int_compare:nNnTF {####1} = {1}
1424                 { \hspace{#2} }
1425                 { \hspace{\g__xtable_col_sep_dim} }
1426                 \__xtable_parse_cell:nn {##1} {####1}
1427                 \__xtable_parse_cell_fill:nn {##1} {####1}
1428                 \__xtable_parse_cell_ht:nn {##1} {####1}

```

```

1429         \dim_set_eq:Nn \l__xtable_cacha_dim \l__xtable_ht_dim
1430         \__xtable_parse_cell_size:nn {##1} {###1}
1431         \__xtable_parse_row_align:n {##1}
1432         \__xtable_parse_col_align:n {###1}
1433         \__xtable_print_data:VVV
1434         \l__xtable_row_align_tl \l__xtable_col_align_tl \l__xtable_data_tl
1435     }
1436 }
1437 \IfBooleanF {#1}
1438 { \int_compare:nNnT {##1} < {\g__xtable_row_count_int} {\} }
1439 }
1440 }

```

(End of definition for `\printtable`. This function is documented on page 7.)

7.3 渲染表格

7.3.1 渲染内容

`__xtable_render_data:nnn` 使用垂直盒子渲染指定数据。#1: 垂直对齐方式，#2: 水平对齐方式，#3: 内容。渲染好的盒子存储在 `\l__xtable_cell_box` 中。

```

1441 \cs_new:Nn \__xtable_render_data:nnn
1442 {
1443   \str_case:nnF{#2}
1444   {
1445     {l} { \tl_set:Nn \l__xtable_tmpa_tl {\raggedright} }
1446     {c} { \tl_set:Nn \l__xtable_tmpa_tl {\centering} }
1447     {r} { \tl_set:Nn \l__xtable_tmpa_tl {\raggedleft} }
1448   }
1449   { \tl_set:Nn \l__xtable_tmpa_tl {} }
1450   \tl_if_empty:nTF {#3}
1451   { \tl_set:Nn \l__xtable_tmpb_tl {\rule{1pt}{0pt}} }
1452   { \tl_set:Nn \l__xtable_tmpb_tl {#3} }
1453   \str_if_eq:nnTF {#1} {t}
1454   { \vbox_set_top:Nn } { \vbox_set:Nn }
1455   \l__xtable_cell_box
1456   {
1457     \baselineskip=3ex \hsize=\l__xtable_wd_dim \parindent=0pt
1458     \l__xtable_tmpa_tl \l__xtable_tmpb_tl \par
1459   }
1460 }
1461 \cs_generate_variant:Nn \__xtable_render_data:nnn {VVV}

```

(End of definition for `__xtable_render_data:nnn`.)

`_xtable_draw_cell:nn` 绘制指定单元格, #1#2 为数字坐标。使用 `\l_xtable_x_dim` 与 `\l_xtable_y_dim` 的坐标信息。

```

1462 \cs_new:Nn \_xtable_draw_cell:nn
1463 {
1464   \_xtable_parse_cell:nn      {#1} {#2}
1465   \_xtable_parse_cell_fill:nn {#1} {#2}
1466   \_xtable_parse_cell_size:nn {#1} {#2}
1467   \_xtable_parse_row_align:n {#1}
1468   \_xtable_parse_col_align:n {#2}
1469   \_xtable_render_data:VVV
1470   \l\_xtable_row_align_tl \l\_xtable_col_align_tl \l\_xtable_data_tl
1471   \tl_if_eq:NnTF \l\_xtable_row_align_tl {m}
1472   {
1473     \dim_set:Nn \l\_xtable_tmpb_dim
1474       { \box_ht:N \l\_xtable_cell_box + \l\_xtable_ufill_dim }
1475     \dim_set:Nn \l\_xtable_tmpa_dim
1476       { \l\_xtable_y_dim - \l\_xtable_tmpb_dim }
1477     \dim_sub:Nn \l\_xtable_tmpa_dim
1478       { (\l\_xtable_ht_dim - \l\_xtable_tmpb_dim) / 2 }
1479   }
1480   {
1481     \dim_set:Nn \l\_xtable_tmpa_dim
1482       { \l\_xtable_y_dim - \l\_xtable_ht_dim }
1483   }
1484   \draw_box_use:Nn \l\_xtable_cell_box {\l\_xtable_x_dim, \l\_xtable_tmpa_dim }
1485 }

```

(End of definition for _xtable_draw_cell:nn.)

`_xtable_draw_cells:` 绘制所有单元格, 使用 `\l_xtable_x_dim` 与 `\l_xtable_y_dim` 的坐标信息。

```

1486 \cs_new:Nn \_xtable_draw_cells:
1487 {
1488   \dim_set:Nn \l\_xtable_y_dim {-\c\_xtable_std_dp_dim / 3} % 补偿
1489   \bool_if:NTF \g\_xtable_has_header_bool
1490     { \int_set:Nn \l\_xtable_tmpa_int {0} }
1491     {
1492       \int_set:Nn \l\_xtable_tmpa_int {1}
1493       \dim_sub:Nn \l\_xtable_y_dim { \seq_item:Nn \g\_xtable_row_sep_seq {1} }
1494       \dim_add:Nn \l\_xtable_y_dim { \seq_item:Nn \g\_xtable_row_sep_seq {2} }
1495     }
1496   \int_step_inline:nnn {\l\_xtable_tmpa_int} {\g\_xtable_row_count_int}
1497     {
1498       \dim_sub:Nn \l\_xtable_y_dim { \seq_item:Nn \g\_xtable_row_sep_seq {##1+1} }

```

```

1499     \dim_zero:N \l__xtable_x_dim
1500     \int_step_inline:nn {\g__xtable_col_count_int}
1501     {
1502         \dim_add:Nn \l__xtable_x_dim { \seq_item:Nn \g__xtable_col_sep_seq {####1} }
1503         \__xtable_draw_cell:nn {##1} {####1}
1504         \dim_add:Nn \l__xtable_x_dim { \l__xtable_wd_dim }
1505     }
1506     \dim_sub:Nn \l__xtable_y_dim
1507     { \l__xtable_ht_dim + \l__xtable_dp_dim }
1508 }
1509 }

```

(End of definition for __xtable_draw_cells:.)

7.3.2 渲染边框-三线表

`__xtable_draw_booktabs:n` 绘制三线表的边框线。使用变量 `\l__xtable_cachea_int`。

```

1510 \cs_new:Nn \__xtable_draw_booktabs:n
1511 {
1512     \dim_zero:N \l__xtable_expand_start_dim
1513     \dim_zero:N \l__xtable_expand_end_dim
1514     \__xtable_set_line_style:n {toprule}
1515     \__xtable_draw_hline:n {1}
1516     \draw_path_use_clear:n { stroke }
1517     \bool_if:NTF \g__xtable_has_header_bool
1518     {
1519         \__xtable_set_line_style:n {midrule}
1520         \__xtable_draw_hline:n {2}
1521         \draw_path_use_clear:n { stroke }
1522         \int_set:Nn \l__xtable_cachea_int {2}
1523     }
1524     { \int_set:Nn \l__xtable_cachea_int {1} }
1525     \__xtable_set_line_style:n {bottomrule}
1526     \__xtable_draw_hline:n {-1}
1527     \draw_path_use_clear:n { stroke }
1528     \__xtable_set_line_style:n {cmidrule}
1529     \clist_map_inline:nn {#1}
1530     { \__xtable_draw_hline:n {##1+\l__xtable_cachea_int} }
1531     \draw_path_use_clear:n { stroke }
1532     \__xtable_rendertable_post:
1533 }

```

(End of definition for __xtable_draw_booktabs:n.)

7.3.3 渲染边框-网格线

`_xtable_row_line:nn` 设置网格线风格的行网格线。#1 为默认值，#2 为线列表。

```

1534 \NewDocumentCommand {\_xtable_row_line:nn} { 0{ } m }
1535 {
1536   \tl_set:Nn \l__xtable_style_tl {#1}
1537   \tl_if_empty:nTF {#2}
1538     { \seq_clear:N \l__xtable_shared_seq }
1539     {
1540       \seq_set_split:Nnn \l__xtable_shared_seq {,} {#2}
1541       \seq_get_left:NN \l__xtable_shared_seq \l__xtable_tmpa_tl
1542       \str_if_in:NnF \l__xtable_tmpa_tl {=}
1543       { \seq_pop_left:NN \l__xtable_shared_seq \l__xtable_style_tl }
1544     }
1545   \_xtable_set_row_style:NNn \g__xtable_row_border_seq
1546     \l__xtable_shared_seq {#1}
1547   \bool_if:NF \g__xtable_has_header_bool
1548     { \seq_gpop_left:NN \g__xtable_row_border_seq \l__xtable_tmpa_tl }
1549   \seq_gput_left:NV \g__xtable_row_border_seq \l__xtable_style_tl
1550 }

```

(End of definition for `_xtable_row_line:nn`.)

`_xtable_col_line:nn` 设置网格线风格的列网格线。#1 为默认值，#2 为线列表。

```

1551 \NewDocumentCommand {\_xtable_col_line:nn} { 0{ } m }
1552 {
1553   \tl_set:Nn \l__xtable_style_tl {#1}
1554   \tl_if_empty:nTF {#2}
1555     { \seq_clear:N \l__xtable_shared_seq }
1556     {
1557       \seq_set_split:Nnn \l__xtable_shared_seq {,} {#2}
1558       \seq_get_left:NN \l__xtable_shared_seq \l__xtable_tmpa_tl
1559       \str_if_in:NnF \l__xtable_tmpa_tl {=}
1560       { \seq_pop_left:NN \l__xtable_shared_seq \l__xtable_style_tl }
1561     }
1562   \_xtable_set_col_style:NNn \g__xtable_col_border_seq
1563     \l__xtable_shared_seq {#1}
1564   \seq_gput_left:NV \g__xtable_col_border_seq \l__xtable_style_tl
1565 }

```

(End of definition for `_xtable_col_line:nn`.)

`_xtable_calc_grid_expand:N` 计算网格线的扩展长度²⁰。

²⁰用于平衡线宽造成的转角过渡问题。

```

1566 \cs_new:Nn \__xtable_calc_grid_expand:N
1567 {
1568   \seq_if_empty:NTF #1
1569     { \seq_set_split:Nnn \l__xtable_tmpa_seq {,} {Opt} }
1570     { \seq_set_eq:NN \l__xtable_tmpa_seq #1 }
1571   \seq_get_left:NN \l__xtable_tmpa_seq \l__xtable_style_tl
1572   \tl_if_empty:NTF \l__xtable_style_tl
1573     { \dim_zero:N \l__xtable_expand_start_dim }
1574     {
1575       \__xtable_parse_line_style:V \l__xtable_style_tl
1576       \dim_set:Nn \l__xtable_expand_start_dim
1577         { \l__xtable_line_dim/2 }
1578     }
1579   \seq_get_right:NN \l__xtable_tmpa_seq \l__xtable_style_tl
1580   \tl_if_empty:NTF \l__xtable_style_tl
1581     { \dim_zero:N \l__xtable_expand_end_dim }
1582     {
1583       \__xtable_parse_line_style:V \l__xtable_style_tl
1584       \dim_set:Nn \l__xtable_expand_end_dim
1585         { \l__xtable_line_dim/2 }
1586     }
1587 }

```

(End of definition for __xtable_calc_grid_expand:N.)

`__xtable_draw_grid:nn` 绘制网格线型表格。

```

1588 \cs_new:Nn \__xtable_draw_grid:nn
1589 {
1590   \group_begin:
1591     \cs_set_eq:NN \rowborder \__xtable_row_line:nn
1592     \cs_set_eq:NN \colborder \__xtable_col_line:nn
1593     #2
1594   \group_end:
1595   \__xtable_calc_grid_expand:N \g__xtable_row_border_seq
1596   \int_step_inline:nn {\seq_count:N \g__xtable_col_loc_seq}
1597     {
1598       \tl_set:Ne \l__xtable_style_tl
1599         { \seq_item:Nn \g__xtable_col_border_seq {##1} }
1600       \tl_if_empty:NF \l__xtable_style_tl
1601         {
1602           \__xtable_set_line_style:V \l__xtable_style_tl
1603           \__xtable_draw_vline:n {##1}
1604           \draw_path_use_clear:n { stroke }

```

```

1605     }
1606   }
1607   \__xtable_calc_grid_expand:N \g__xtable_col_border_seq
1608   \int_step_inline:nn {\seq_count:N \g__xtable_row_loc_seq}
1609   {
1610     \tl_set:Nc \l__xtable_style_tl
1611       { \seq_item:Nn \g__xtable_row_border_seq {##1} }
1612     \tl_if_empty:NF \l__xtable_style_tl
1613     {
1614       \__xtable_set_line_style:V \l__xtable_style_tl
1615       \__xtable_draw_hline:n {##1}
1616       \draw_path_use_clear:n { stroke }
1617     }
1618   }
1619   \__xtable_rendertable_post:
1620 }

```

(End of definition for __xtable_draw_grid:nn.)

7.3.4 渲染边框-框架

__xtable_draw_cell_border:nn 绘制指定单元格边框。

```

1621 \cs_new:Nn \__xtable_draw_cell_border:nn
1622 {
1623   \__xtable_parse_cell_size:nn {#1} {#2}
1624   \dim_set:Nn \l__xtable_tmpa_dim
1625     {\l__xtable_wd_dim + \g__xtable_col_margin_dim * 2}
1626   \dim_set:Nn \l__xtable_tmpb_dim
1627     {\l__xtable_ht_dim + \l__xtable_dp_dim + \g__xtable_row_margin_dim * 2}
1628   \draw_path_rectangle:nn
1629     {\l__xtable_x_dim, \l__xtable_y_dim - \l__xtable_tmpb_dim }
1630     {\l__xtable_tmpa_dim, \l__xtable_tmpb_dim }
1631   \draw_path_use_clear:n { draw }
1632 }

```

(End of definition for __xtable_draw_cell_border:nn.)

__xtable_draw_cell_border: 绘制单元格边框。

```

1633 \cs_new:Nn \__xtable_draw_cell_borders:
1634 {
1635   \dim_zero:N \l__xtable_y_dim
1636   \int_step_inline:nnn
1637     { \bool_if:NTF \g__xtable_has_header_bool {0} {1} }

```



```

1638     {\g__xtable_row_count_int}
1639     {
1640     \dim_zero:N \l__xtable_x_dim
1641     \int_step_inline:nn {\g__xtable_col_count_int}
1642     {
1643     \__xtable_draw_cell_border:nn {##1} {####1}
1644     \dim_set:Nn \l__xtable_x_dim
1645     { \l__xtable_x_dim + \l__xtable_wd_dim + \g__xtable_col_margin_dim * 2 }
1646     }
1647     \dim_set:Nn \l__xtable_y_dim
1648     { \l__xtable_y_dim - \l__xtable_ht_dim - \l__xtable_dp_dim - \g__xtable_row_margin
1649     }
1650     \__xtable_rendertable_post:
1651     }

```

(End of definition for __xtable_draw_cell_border:.)

7.3.5 渲染表格

__xtable_set_table_sep:n 处理表格，计算各类表格尺寸数据。

```

1652 \cs_new:Nn \__xtable_set_table_sep:n
1653 {
1654     \__xtable_set_row_sep:ee
1655     { \dim_use:N \g__xtable_row_margin_dim }
1656     { \dim_eval:n { \g__xtable_row_margin_dim * 2 } }
1657     \__xtable_set_col_sep:ee
1658     {
1659     \bool_if:nTF {#1}
1660     { \dim_use:N \g__xtable_col_margin_dim }
1661     { Opt }
1662     }
1663     { \dim_eval:n { \g__xtable_col_margin_dim * 2 } }
1664     }

```

(End of definition for __xtable_set_table_sep:n.)

__xtable_rendertable_pre: 渲染表格的准备函数与善后的函数。其中，准备函数由 \rendertable 直接调用，而
 __xtable_rendertable_post: 善后函数则由所有绘制表格的函数调用。

```

1665 \cs_new:Nn \__xtable_rendertable_pre:
1666 {
1667     \__xtable_set_table_sep:n {\c_true_bool}
1668     \__xtable_calc_col_wd:
1669     \__xtable_calc_row_ht:

```

```

1670   \_xtable_calc_coord:
1671   \draw_begin:
1672   \vbox_set_to_ht:Nnn \l__xtable_tmpa_box
1673     {\g__xtable_above_space_dim} {}
1674   \draw_box_use:N \l__xtable_tmpa_box
1675   }
1676 \cs_new:Nn \_xtable_rendertable_post:
1677 {
1678   \_xtable_draw_cells:
1679   \draw_end:
1680   }

```

(End of definition for _xtable_rendertable_pre: and _xtable_rendertable_post:.)

`\rendertable` 绘制表格。

```

1681 \NewDocumentCommand \rendertable { 0{} 0{} }
1682 {
1683   \_xtable_rendertable_pre:
1684   \str_case:nnF {#1}
1685     {
1686       {booktabs} { \_xtable_draw_booktabs:n {#2} }
1687       {grid}     { \_xtable_draw_grid:nn  {#2} }
1688     }
1689     { \_xtable_draw_cell_borders: }
1690   }

```

(End of definition for \rendertable. This function is documented on page 8.)

```

1691 \endpackage

```

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols	
<code>@@_data:</code> (env.) 1230
<code>\l</code> 31
C	
<code>\cell</code> 6 , 1223 , 1252
<code>\col</code> 6 , 1196 , 1251
D	
<code>data</code> (env.) 1230
<code>\colalign</code> 6 , 1294
<code>\colborder</code> 8 , 1592
<code>\colname</code> 4 , 1143
<code>\colwidth</code> 6 , 1260

- `\data` 1248
- `data` 4
- E**
- end internal commands:
- `\end_xtable_data:` 1249
- `\enddata` 1249
- environments:
- `@@_data:` 1230
- `data` 1230
- `xtable` 1242
- `\excelcolname` 4, 1138
- F**
- `\file` 5
- L**
- `\linepatternset` 9, 112
- `\linestyleset` 9, 112
- `\linewidthset` 9, 112
- `\loadtable` 6, 1128
- `\logtable` 9, 73, 568
- M**
- `\margin` 10
- P**
- `\printtable` 7, 1409
- R**
- `\rendertable` 8, 65, 1681
- `\row` 6, 1169, 1250
- `\rowalign` 6, 1294
- `\rowborder` 8, 1591
- `\rowheight` 6, 1260
- `\rowname` 4, 1143
- `\rowsep` 9
- S**
- `\savetable` 6, 1128
- scan internal commands:
- `\s_xtable_mark`
 16, 20, 36, 123, 174, 389, 418
- `\showtable` 9, 551
- X**
- `xtable` (env.) 1242
- `xtable` 4
- xtable internal commands:
- `\g_xtable_above_space_dim`
 77, 99, 1673
- `\g_xtable_below_space_dim` . 77, 100
- `\l_xtable_cachea_dim`
 .. 57, 57, 663, 681, 694, 720, 727,
 774, 793, 812, 820, 1382, 1394, 1429
- `\l_xtable_cachea_int` 61, 55, 701,
 714, 718, 721, 1097, 1100, 1102,
 1414, 1415, 1416, 1522, 1524, 1530
- `\l_xtable_cachea_seq` ... 61, 665,
 680, 691, 702, 715, 723, 776, 792, 804
- `\l_xtable_cachea_tl` 48, 59, 1091, 1098
- `\l_xtable_cacheb_dim`
 . 57, 664, 686, 699, 775, 794, 812, 820
- `\l_xtable_cacheb_int` 55
- `\l_xtable_cacheb_seq` 61, 666, 685, 696
- `\l_xtable_cacheb_tl`
 48, 59, 1095, 1096, 1104
- `_xtable_calc_col_wd:`
 661, 661, 1412, 1668
- `_xtable_calc_coord:` . 849, 849, 1670
- `_xtable_calc_grid_expand:N` ...
 1566, 1566, 1595, 1607
- `_xtable_calc_row_ht:`
 772, 772, 1412, 1669
- `_xtable_cell:nn` .. 1223, 1223, 1252
- `\l_xtable_cell_box`
 59, 54, 1455, 1474, 1484
- `\c_xtable_cell_content_tl`
 208, 405, 421
- `\g_xtable_cell_data_prop` .. 212,
 317, 356, 376, 388, 398, 416, 553, 575
- `\g_xtable_cell_dfill_prop`
 231, 656, 669, 839
- `\c_xtable_cell_empty_tl` .. 208, 427
- `\c_xtable_cell_formula_tl` 208, 423
- `\g_xtable_cell_ht_prop`
 231, 764, 779, 846

- \l__xtable_cell_input_seq
 - [934](#), [962](#), [966](#), [974](#),
 - [993](#), [1003](#), [1007](#), [1015](#), [1086](#), [1088](#),
 - [1090](#), [1091](#), [1092](#), [1093](#), [1149](#), [1150](#),
 - [1162](#), [1163](#), [1189](#), [1190](#), [1216](#), [1217](#)
- \g__xtable_cell_lineskip_dim
 - [84](#), [105](#), [742](#), [1374](#), [1385](#), [1397](#)
- \c__xtable_cell_merged_tl . . [208](#), [425](#)
- \g__xtable_cell_ufill_prop
 - [231](#), [654](#), [668](#), [837](#)
- \g__xtable_cell_wd_min_dim
 - [83](#), [103](#), [620](#)
- __xtable_check_input_env:n
 - [926](#), [926](#), [1130](#), [1135](#), [1140](#),
 - [1145](#), [1158](#), [1262](#), [1279](#), [1296](#), [1311](#)
- __xtable_col:nmn . . [1196](#), [1196](#), [1251](#)
- \g__xtable_col_align_seq
 - [218](#), [323](#), [362](#), [382](#), [518](#), [528](#), [549](#), [581](#)
- \l__xtable_col_align_tl
 - [31](#), [49](#), [548](#), [1434](#), [1470](#)
- \g__xtable_col_border_seq
 - [236](#), [1562](#), [1564](#), [1599](#), [1607](#)
- \g__xtable_col_count_int
 - [212](#), [254](#), [294](#), [313](#), [352](#), [372](#), [391](#),
 - [471](#), [472](#), [561](#), [571](#), [592](#), [610](#), [670](#),
 - [707](#), [757](#), [856](#), [1201](#), [1421](#), [1500](#), [1641](#)
- \g__xtable_col_formula_prop
 - [216](#), [319](#), [358](#), [378](#), [577](#)
- \g__xtable_col_header_seq
 - [225](#), [434](#), [556](#), [589](#), [595](#), [596](#)
- __xtable_col_line:nm [1551](#), [1551](#), [1592](#)
- \l__xtable_col_loc_int [23](#), [24](#), [27](#),
 - [29](#), [44](#), [45](#), [48-52](#), [37](#), [275](#), [291](#), [293](#),
 - [294](#), [295](#), [411](#), [470](#), [483](#), [484](#), [961](#),
 - [991](#), [1097](#), [1099](#), [1100](#), [1101](#), [1102](#),
 - [1160](#), [1161](#), [1165](#), [1166](#), [1174](#), [1180](#),
 - [1185](#), [1193](#), [1201](#), [1207](#), [1212](#), [1226](#)
- \g__xtable_col_loc_seq
 - [234](#), [852](#), [853](#),
 - [864](#), [868](#), [1332](#), [1337](#), [1345](#), [1350](#), [1596](#)
- \g__xtable_col_margin_dim
 - [77](#), [94](#), [1625](#), [1645](#), [1660](#), [1663](#)
- \g__xtable_col_name_prop [206](#), [242](#),
 - [253](#), [275](#), [291](#), [316](#), [355](#), [375](#), [574](#), [590](#)
- \g__xtable_col_sep_dim
 - [77](#), [89](#), [1411](#), [1425](#)
- \g__xtable_col_sep_seq
 - [226](#), [609](#), [611](#),
 - [612](#), [613](#), [705](#), [855](#), [859](#), [863](#), [867](#), [1502](#)
- \c__xtable_col_wd_regex
 - [910](#), [1284](#), [1289](#)
- \g__xtable_col_wd_seq
 - [228](#), [667](#), [676](#), [689](#),
 - [693](#), [698](#), [710](#), [726](#), [728](#), [762](#), [829](#), [861](#)
- \g__xtable_col_wd_style_seq
 - [218](#), [321](#), [360](#), [380](#), [502](#),
 - [504](#), [507](#), [511](#), [513](#), [524](#), [539](#), [579](#), [704](#)
- \l__xtable_color_tl
 - [19](#), [49](#), [193](#), [201](#), [202](#)
- \c__xtable_csv_cell_regex . . [896](#), [995](#)
- \c__xtable_csv_row_regex . . [896](#), [987](#)
- __xtable_data: [1248](#)
- \l__xtable_data_tl
 - [27](#), [32](#), [35](#), [47](#), [422](#), [424](#), [426](#),
 - [428](#), [434](#), [435](#), [564](#), [652](#), [763](#), [1434](#), [1470](#)
- \l__xtable_dfill_dim [34](#), [35](#),
 - [37](#), [38](#), [40](#), [41](#), [636](#), [637](#), [657](#), [748](#), [842](#)
- __xtable_dim_set_max:NN . . . [154](#),
 - [154](#), [653](#), [681](#), [686](#), [766](#), [767](#), [793](#), [794](#)
- __xtable_dim_set_max:Nn
 - [154](#), [156](#), [628](#), [632](#), [637](#), [745](#), [747](#)
- \l__xtable_dp_dim [37](#),
 - [38](#), [40](#), [57](#), [41](#), [735](#), [747](#), [767](#), [770](#),
 - [789](#), [794](#), [799](#), [801](#), [811](#), [814](#), [817](#),
 - [820](#), [832](#), [1404](#), [1405](#), [1507](#), [1627](#), [1648](#)
- __xtable_draw_booktabs:n
 - [1510](#), [1510](#), [1686](#)
- __xtable_draw_cell:nm
 - [1462](#), [1462](#), [1503](#)
- __xtable_draw_cell_border: . . . [1633](#)
- __xtable_draw_cell_border:nm
 - [1621](#), [1621](#), [1643](#)
- __xtable_draw_cell_borders:
 - [1633](#), [1689](#)

- _xtable_draw_cells: [1486](#), [1486](#), [1678](#)
- _xtable_draw_grid:nn
..... [1588](#), [1588](#), [1687](#)
- _xtable_draw_hline:n [1328](#),
[1328](#), [1515](#), [1520](#), [1526](#), [1530](#), [1615](#)
- _xtable_draw_vline:n
..... [1341](#), [1341](#), [1603](#)
- \l_xtable_esc_status_bool
..... [931](#), [1057](#), [1058](#), [1061](#), [1111](#)
- \c_xtable_escape_str [30](#), [1060](#)
- \l_xtable_expand_end_dim
[56](#), [1324](#), [1337](#), [1351](#), [1513](#), [1581](#), [1584](#)
- \l_xtable_expand_start_dim
[56](#), [1324](#), [1332](#), [1346](#), [1512](#), [1573](#), [1576](#)
- _xtable_get_cell:nnN [438](#)
- _xtable_get_loc:NnNn
..... [258](#), [258](#), [270](#), [275](#), [282](#), [291](#)
- _xtable_ginit_seq:Nnn
..... [139](#), [144](#), [452](#), [472](#)
- \g_xtable_has_header_bool
[212](#), [328](#), [350](#), [370](#), [442](#), [584](#), [646](#),
[872](#), [1247](#), [1413](#), [1489](#), [1517](#), [1547](#), [1637](#)
- \c_xtable_ht_dim [34](#)
- \l_xtable_ht_dim . [37](#), [38](#), [40](#), [41](#),
[57](#), [41](#), [734](#), [745](#), [765](#), [766](#), [769](#), [787](#),
[793](#), [809](#), [812](#), [819](#), [822](#), [830](#), [847](#),
[869](#), [871](#), [875](#), [881](#), [884](#), [886](#), [888](#),
[890](#), [892](#), [894](#), [1382](#), [1383](#), [1394](#),
[1429](#), [1478](#), [1482](#), [1507](#), [1627](#), [1648](#)
- _xtable_init_seq:Nnn . [139](#), [139](#), [149](#)
- \l_xtable_input_bool
..... [924](#), [928](#), [1253](#), [1256](#)
- \l_xtable_input_format_tl
..... [936](#), [943](#), [1233](#), [1239](#)
- \l_xtable_input_has_header_bool
..... [936](#),
[945](#), [946](#), [964](#), [971](#), [1005](#), [1012](#), [1247](#)
- \l_xtable_input_sep_tl [936](#), [948](#), [963](#)
- _xtable_insure_style: [519](#), [519](#), [1257](#)
- _xtable_int_gset_max:Nn
[150](#), [152](#), [249](#), [254](#), [390](#), [391](#), [450](#), [471](#)
- _xtable_int_set_max:Nn .. [150](#), [150](#)
- \c_xtable_json_cell_regex [900](#), [1086](#)
- \c_xtable_lbrace_str . [30](#), [1031](#), [1047](#)
- \l_xtable_line_dim
..... [19](#), [46](#), [192](#), [197](#), [198](#), [1577](#), [1585](#)
- \g_xtable_line_pattern_prop ...
..... [109](#), [113](#), [118](#), [160](#)
- \g_xtable_line_style_prop
..... [109](#), [122](#), [172](#), [188](#)
- \g_xtable_line_wd_prop
..... [109](#), [115](#), [120](#), [166](#), [196](#)
- _xtable_measure_cell_ht:n [732](#)
- _xtable_measure_cell_ht:nnn ..
..... [732](#), [751](#), [761](#)
- _xtable_measure_cell_wd:n
..... [617](#), [617](#), [642](#), [652](#)
- _xtable_measure_col_wd:n
..... [643](#), [643](#), [672](#)
- _xtable_measure_row_ht:n
..... [752](#), [752](#), [783](#)
- \g_xtable_merge_count_int
..... [222](#), [314](#), [353](#), [373](#), [572](#)
- \g_xtable_merge_info_prop
..... [222](#), [324](#), [363](#), [383](#), [582](#)
- \g_xtable_merge_ref_prop
..... [222](#), [325](#), [364](#), [384](#), [583](#)
- \l_xtable_offset_x_dim [1326](#)
- \l_xtable_offset_y_dim [1326](#)
- _xtable_parse_cell:nn [414](#),
[414](#), [439](#), [563](#), [651](#), [759](#), [1426](#), [1464](#)
- _xtable_parse_cell_fill:nn ...
..... [760](#), [835](#), [835](#), [1427](#), [1465](#)
- _xtable_parse_cell_ht:nn
..... [844](#), [844](#), [1428](#)
- _xtable_parse_cell_size:nn ...
..... [826](#), [826](#), [1430](#), [1466](#), [1623](#)
- _xtable_parse_col_align:n
..... [541](#), [546](#), [1432](#), [1468](#)
- _xtable_parse_col_loc:n
..... [268](#), [273](#), [279](#), [303](#), [480](#)
- _xtable_parse_col_wd_style:n .
..... [531](#), [536](#), [673](#), [711](#)
- _xtable_parse_coord:n [300](#), [300](#), [950](#)

- _xtable_parse_csv:n . 983, 983, 1236
- _xtable_parse_input:n 953, 953, 1235
- _xtable_parse_json:n ... 1108, 1237
- _xtable_parse_json:nnn . 1108, 1127
- _xtable_parse_json_auxa:n
..... 1024, 1024, 1120
- _xtable_parse_json_auxb:n
..... 1039, 1039, 1121
- _xtable_parse_json_auxc:n
..... 1055, 1055, 1122
- _xtable_parse_json_row:n
..... 1083, 1083, 1125
- _xtable_parse_line_style:n ...
..... 186, 186, 205, 1575, 1583
- _xtable_parse_new_col_loc:n ..
..... 280, 289, 299, 308, 1098, 1213
- _xtable_parse_new_coord:n
..... 300, 305, 1181, 1208, 1227
- _xtable_parse_new_row_loc:n ..
..... 280, 280, 298, 307, 1186
- _xtable_parse_row_align:n
..... 541, 541, 756, 806, 1431, 1467
- _xtable_parse_row_ht_style:n .
..... 531, 531, 782
- _xtable_parse_row_loc:n
..... 268, 268, 278, 302, 460
- \l_xtable_parse_status_int
..... 931, 1030,
1033, 1046, 1049, 1074, 1110, 1118
- \l_xtable_pattern_tl
..... 19, 49, 191, 199, 200
- _xtable_print_data:nnn
..... 1354, 1354, 1408, 1433
- _xtable_process_header:
..... 586, 586, 1258
- \l_xtable_quote_status_bool ...
..... 931, 1063, 1066, 1070, 1112
- \c_xtable_rbrace_str 30, 1072
- _xtable_render_data:nnn
..... 1441, 1441, 1461, 1469
- _xtable_rendertable_post:
..... 1532, 1619, 1650, 1665, 1676
- _xtable_rendertable_pre:
..... 1665, 1665, 1683
- _xtable_row:nnn ... 1169, 1169, 1250
- \g_xtable_row_align_seq
218, 322, 361, 381, 516, 526, 544, 580
- \l_xtable_row_align_tl 31,
38, 49, 543, 761, 807, 1434, 1470, 1471
- \g_xtable_row_border_seq
... 236, 1545, 1548, 1549, 1595, 1611
- \g_xtable_row_count_int
.... 212, 249, 285, 312, 351, 371,
390, 450, 452, 558, 570, 602, 649,
780, 877, 1173, 1416, 1438, 1496, 1638
- \g_xtable_row_dp_seq
228, 778, 788, 800, 813, 818, 833, 887
- \c_xtable_row_ht_regex
..... 910, 1267, 1272
- \g_xtable_row_ht_seq
228, 777, 786, 798, 810, 821, 831, 885
- \g_xtable_row_ht_style_seq
..... 218, 320, 359,
379, 489, 491, 494, 497, 522, 534, 578
- \l_xtable_row_input_seq 934, 955,
957, 985, 987, 989, 1075, 1114, 1125
- _xtable_row_line:nn 1534, 1534, 1591
- \l_xtable_row_loc_int
. 23, 24, 27, 28, 44, 45, 48-52, 37,
270, 282, 284, 285, 286, 411, 445,
448, 464, 465, 956, 988, 1104, 1106,
1147, 1148, 1152, 1153, 1173, 1179,
1184, 1200, 1206, 1211, 1220, 1225
- \g_xtable_row_loc_seq
..... 234, 870, 871,
890, 894, 1333, 1338, 1346, 1351, 1608
- \g_xtable_row_margin_dim
..... 77, 93, 1627, 1648, 1655, 1656
- \g_xtable_row_name_prop 206, 248,
270, 282, 315, 326, 327, 354, 374, 573
- \g_xtable_row_sep_dim 77, 87
- \g_xtable_row_sep_seq
..... 226, 601, 603, 604, 605,
876, 882, 889, 893, 1493, 1494, 1498

- \g_xtable_row_summary_prop 1320, 1322, 1538, 1540, 1541, 1543, 1546, 1555, 1557, 1558, 1560, 1563
- __xtable_set_cell:n \c_xtable_space_str . . . 30, 1028, 1043
- 394, 408, 1192, 1219, 1228
- __xtable_set_cell:NNn \c_xtable_std_dp_dim
- 34, 638, 735, 1488
- __xtable_set_cell:NNn \c_xtable_std_ht_dim . . . 34, 633, 734
- . . . 394, 394, 410, 413, 976, 1017, 1103
- __xtable_set_cell:nnnn \l_xtable_style_tl
- 386, 386, 393, 403
- 31, 47, 533, 538, 674, 678, 683, 689, 712, 784, 790, 799, 1536, 1543, 1549, 1553, 1560, 1564, 1571, 1572, 1575, 1579, 1580, 1583, 1598, 1600, 1602, 1610, 1612, 1614
- __xtable_set_col_align:Nn __xtable_table_init: . . 310, 310, 1244
- 515, 517, 529, 1322
- __xtable_set_col_name:nN __xtable_table_restore:n
- 246, 251, 257, 295, 968, 1009, 1165
- 366, 366, 1131
- __xtable_set_col_sep:nn __xtable_table_save:n . 330, 330, 1136
- 599, 607, 616, 1411, 1657
- __xtable_set_col_style:NNn \l_xtable_tmpa_bool . . . 623, 630, 634
- 440, 468, 502, 518, 1562
- __xtable_set_col_wd_style:Nnn \l_xtable_tmpa_box . . . 75, 627, 629, 633, 638, 740, 746, 748, 1372, 1383, 1395, 1404, 1405, 1406, 1672, 1674
- 487, 500, 525, 1292
- __xtable_set_excel_col_names:n \l_xtable_tmpa_dim 35, 38, 65, 645, 653, 659, 754, 766, 769, 1381, 1387, 1393, 1399, 1420, 1475, 1477, 1481, 1484, 1624, 1630
- 238, 238, 1141
- __xtable_set_line_pattern:n \l_xtable_tmpa_int 63, 444, 447, 451, 647, 648, 649, 873, 874, 877, 879, 956, 976, 979, 988, 1017, 1020, 1099, 1104, 1490, 1492, 1496
- 158, 158, 180
- __xtable_set_line_style:n \l_xtable_tmpa_prop . 71, 588, 591, 594
- 158, 170, 185, 1514, 1519, 1525, 1528, 1602, 1614
- __xtable_set_line_wd:n . 158, 164, 181
- __xtable_set_row_align:Nn \l_xtable_tmpa_seq . . . 71, 174, 175, 176, 177, 190, 191, 192, 193, 418, 419, 422, 424, 457, 458, 459, 477, 478, 479, 624, 625, 992, 995, 996, 999, 1001, 1267, 1272, 1284, 1289, 1298, 1313, 1569, 1570, 1571, 1579
- 515, 515, 527, 1307
- __xtable_set_row_ht_style:Nn \l_xtable_tmpa_str
- 487, 487, 523, 1275
- 67, 1034, 1050, 1075, 1081, 1113
- __xtable_set_row_name:nN \l_xtable_tmpa_tl 67, 118, 119, 123, 160, 161, 166, 167, 172, 174, 188, 190, 196, 197, 263, 264, 400, 416, 418, 458, 460, 478, 480, 493, 495, 496, 497, 506, 508,
- 246, 246, 256, 286, 1152
- __xtable_set_row_sep:nn
- 599, 599, 615, 1654
- __xtable_set_row_style:NNn
- 440, 440, 489, 516, 1545
- __xtable_set_table_sep:n
- 1652, 1652, 1667
- \l_xtable_shared_seq 30, 54, 53, 521, 523, 525, 527, 529, 1264, 1265, 1269, 1275, 1281, 1282, 1286, 1292, 1301, 1305, 1307, 1316,

- 509, 510, 511, 594, 595, 837, 838,
841, 846, 847, 959, 960, 963, 998,
1002, 1003, 1090, 1092, 1094, 1096,
1303, 1304, 1305, 1318, 1319, 1320,
1361, 1362, 1363, 1365, 1367, 1376,
1388, 1400, 1445, 1446, 1447, 1449,
1458, 1541, 1542, 1548, 1558, 1559
- `\l__xtable_tmpb_box` [75](#)
- `\l__xtable_tmpb_dim`
..... [38](#), [65](#), [755](#), [767](#),
[770](#), [1473](#), [1476](#), [1478](#), [1626](#), [1629](#), [1630](#)
- `\l__xtable_tmpb_int`
..... [63](#), [961](#), [968](#), [969](#),
[976](#), [977](#), [991](#), [1009](#), [1010](#), [1017](#), [1018](#)
- `\l__xtable_tmpb_prop` [71](#)
- `\l__xtable_tmpb_seq` [71](#)
- `\l__xtable_tmpb_str` [67](#)
- `\l__xtable_tmpb_tl`
..... [67](#), [120](#), [121](#), [123](#),
[459](#), [462](#), [464](#), [479](#), [482](#), [483](#), [839](#),
[840](#), [842](#), [1000](#), [1002](#), [1093](#), [1095](#),
[1357](#), [1358](#), [1368](#), [1451](#), [1452](#), [1458](#)
- `\l__xtable_ufill_dim`
.. [34](#), [35](#), [37](#), [38](#), [40](#), [57](#), [41](#), [619](#),
[632](#), [655](#), [746](#), [841](#), [1382](#), [1394](#), [1474](#)
- `\l__xtable_wd_dim` [34](#), [35](#),
[40](#), [41](#), [57](#), [41](#), [620](#), [628](#), [653](#), [659](#),
[677](#), [681](#), [686](#), [703](#), [706](#), [709](#), [721](#),
[725](#), [727](#), [729](#), [828](#), [851](#), [853](#), [854](#),
[858](#), [860](#), [862](#), [864](#), [866](#), [868](#), [1375](#),
[1386](#), [1398](#), [1457](#), [1504](#), [1625](#), [1645](#)
- `\l__xtable_x_dim` [60](#), [39](#), [1484](#), [1499](#),
[1502](#), [1504](#), [1629](#), [1640](#), [1644](#), [1645](#)
- `\l__xtable_y_dim` [60](#),
[39](#), [1476](#), [1482](#), [1488](#), [1493](#), [1494](#),
[1498](#), [1506](#), [1629](#), [1635](#), [1647](#), [1648](#)
- `xtable/input` [939](#)
- `xtable/package` [85](#)
- `xtable/package/colsep` [8](#)
- `xtable/package/lineskip` [8](#)
- `xtable/package/margin` [8](#)
- `xtable/package/minwidth` [8](#)
- `xtable/package/rowsep` [8](#)
- `xtable/package/vspace` [8](#)

To do...

- 1 (p. 3): **Debug** 修复由于测量单元格自然尺寸导致的 `\footnotemark` 命令重复计数的问题
- 2 (p. 4): **方向** 思考应当支持什么选项
- 3 (p. 4): **功能** 添加独立的行列名设置功能
- 4 (p. 5): **功能** 处理 CSV 格式中的特殊字符
- 5 (p. 5): **功能** 处理 JSON 格式中的特殊字符
- 6 (p. 5): **功能** 实现文件导入及细节描述
- 7 (p. 6): **功能** 添加按行/列输入的命令 `\row` 、 `\col`
- 8 (p. 6): **功能** 添加合并单元格功能
- 9 (p. 8): **优化** 更新边框命令及说明
- 10 (p. 8): **优化**更新 `vspace` 的实现与说明
- 11 (p. 25): **Debug** 修复表格保存命令报错的 Bug
- 12 (p. 32): **优化** 显示表格的更多内容信息
- 13 (p. 32): **Debug** 修复 `\logtable` 实际调用成 `show` 函数的 Bug
- 14 (p. 35): **Debug** 修复宽度样式为填充时长宽计算异常的 Bug
- 15 (p. 38): **Debug** 修复多行顶对齐时，相同行高居中对齐失效的 Bug
- 16 (p. 45): **Debug** 修复 CSV 正则表达式与代码不匹配的 Bug
- 17 (p. 61): **功能** 处理无表头时的渲染情况
- 18 (p. 61): **功能** 三线表标题自动合并
- 19 (p. 64): **优化** 提供内外边框的设置接口
- 20 (p. 64): **优化** 适应合并单元格的框架
- 21 (p. 65): **功能** 添加并排输出功能