

tabular2*: 一个实验性的表格排版方案

Ms_yam (Ms_yam@163.com)

二零二六年三月十六日

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 + 盒子 + `l3draw`¹ 绘图，完全摆脱了对传统的底层表格命令的依赖²。

目录

1	用户文档	4
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	5
1.1.4	格式设置	7
1.1.5	合并单元格	7
1.2	渲染输出	8
1.3	选项参数	9
1.3.1	宏包选项	9
1.3.2	设置命令	10
1.4	调试	10
2	关键逻辑	10
2.1	单元格大小	10
2.1.1	间距	10
2.1.2	宽度	11
2.1.3	高度	11

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

¹此宏包当前仍被官方标注为“实验性”宏包，因此本宏包存在未及时响应 `l3draw` 宏包变更而带来异常的风险。

²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	12
3.1	通用变体	12
3.2	l3draw 函数	12
3.3	消息定义	13
3.4	宏包选项	13
3.5	通用变量	14
3.5.1	全局常量	14
3.5.2	专用变量	15
3.5.3	缓存变量	16
3.5.4	临时变量	17
3.6	通用函数	18
3.6.1	常规设置	18
3.6.2	脚注函数	19
3.6.3	临时函数	20
3.7	边框线样式	20
3.7.1	专用变量	20
3.7.2	样式设置	21
3.7.3	应用样式	22
4	表格核心	23
4.1	核心存储	23
4.1.1	行列属性	23
4.1.2	单元格内容	24
4.1.3	合并单元格	25
4.2	行列定位	25
4.2.1	行列名设置	25
4.2.2	行列名解析	26
4.3	内容处理	27
4.3.1	初始化与保存	27
4.3.2	内容设置	29
4.3.3	内容查询	31
4.4	修改表格	32
4.4.1	表格重组	32
4.4.2	移动表格	33
4.4.3	移动单元格	34
4.5	格式处理	35
4.5.1	样式设置	35
4.5.2	样式查询	37
4.6	合并单元格	38
4.6.1	合并设置	38
4.6.2	合并查询	39
4.7	表头填充	40
4.8	调试	41

目录	3
5 输入接口	42
5.1 变量与选项	42
5.1.1 正则常量	42
5.1.2 状态参数	43
5.1.3 局部变量	44
5.1.4 局部选项	44
5.2 数据解析	45
5.2.1 标准格式	45
5.2.2 CSV 数据	46
5.2.3 JSON 数据	47
5.3 数据录入	51
5.4 格式设置	54
5.5 合并单元格	55
6 中间计算	56
6.1 中间变量	56
6.2 行列尺寸	57
6.2.1 边距设置	57
6.2.2 宽度计算	58
6.2.3 高度计算	62
6.2.4 查询尺寸	65
6.3 合并单元格	66
6.3.1 尺寸计算	66
6.3.2 尺寸查询	69
6.4 计算坐标	70
7 渲染输出	71
7.1 通用内容	71
7.1.1 局部变量	71
7.1.2 渲染函数	72
7.2 打印表格	73
7.3 渲染表格	76
7.3.1 渲染内容	76
7.3.2 渲染边框-三线表	80
7.3.3 渲染边框-网格线	80
7.3.4 渲染边框-全边框	83
7.3.5 渲染选项	85
7.3.6 渲染表格	86
Change History	88
Index	88

1 用户文档

与传统表格不同，本模块将表格的数据录入与渲染输出彻底分开，前者用于设置表格数据（内容及内容样式）³，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 Header⁴）；行列名则可由用户指定的⁵，但不能是以下名称之一：`header`、`title`、`first`、`last`。

本宏包直接支持脚注命令 `\footnote` 与 `\footnotemark`，但不建议混合使用这两个命令⁶。

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，必须通过嵌套在本环境中的环境与命令来输入。

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。

³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。

⁴在设计上，表头的坐标为 0，且行列表头（如果有）的默认内容为行列名。

⁵当行列名为纯数字时，定位按数字坐标直接处理。即使第 1 列的列名为 3，3 也始终指向第 3 列。

⁶前者是通过使用 `\footnotemark + \footnotetext` 命令实现的，混合使用有乱序风险。

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⁷，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。

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⁸。由于 TeX 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。

⁷在标准输入中，如果单元格里也有换行，请使用 `\newline`。

⁸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。

```

\begin{xtable}
  % format 用于指定输入的格式，也可以直接使用 json 选项
  \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】

TeX 黑客笔记： 文件中的换行符无需处理，其他待确认。

```
\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⁹。

⁹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。

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，则默认为 `header`（有 `Header`）或 `1`（没有 `Header`）。

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¹⁰；`fill` 则会根据总宽度来补偿，即为“自然尺寸 + 补偿值”¹¹；`samefill` 等于 `same`¹² + `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。

1.1.5 合并单元格

`\mergcell` `\mergcell` [`<数据角标>`] `{<对角坐标 1>}` `{<对角坐标 2>}` [`<行列对齐方式>`]

合并〈对角坐标 1〉与〈对角坐标 2〉之间的单元格。合并的后的单元格内容可以采用这个区域中的任何一个角的单元格内容；〈数据角标〉用来指定实际使用哪个角，默认为 `ul`。合单单元格的对齐方式可由〈行列对齐方式〉定义，默认为 `mc`。

TeX 黑客笔记： 计算单元格排版尺寸时，会跳过合并单元格，即合并单元格的尺寸由其所在的行与列决定。默认常规单元格的对齐方式由其所在的行与列中指定，无法专门指定。如果确实需要单独指定，可以本命令合并一个 `1*1` 的单元格，再指定该合并单元格的对齐方式。

¹⁰同时兼容所有行/列的自然尺寸，即同组自然尺寸的最大值。

¹¹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。

¹²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, rule={⟨中间的线⟩}]
\hrule      \rendertable [grid, rule =
\vrule      {
              \hrule [⟨默认值⟩] {⟨边框列表⟩}
              \vrule [⟨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/en
en

```

使用英文错误消息。

```

xtable/package/cellsep
cellsep = 0.5em

```

设置表格的单元格之间的间距。

```

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

`\rulepatternset` `\rulepatternset` {<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, ...}。

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

将指定 <名称> 的线宽映射到指定 <尺寸>。

`\rulestyleset` `\rulestyleset` {<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：

<0> , title, Name, Sex, Age;

<1> , 1, , man

woman

unknown, 18~25;

<2> , 2, John, man, 18;

<3> , 3, Leon, man, 20;

<4> , 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 `cellsep` 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。

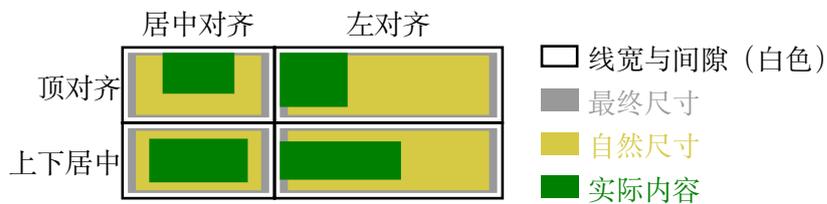


图 1: 表格渲染逻辑

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>。¹³

本节实现本宏包的初始化及选项定义，并定义一些公共的变量与函数。它们完全不依赖表格的设置，只为表格的实现提供功能。

```

1 <*package>
2 <@@=xtable>
3 \NeedsTeXFormat{LaTeX2e}[2023-11-01]
4 \ProvidesExplPackage{tabular2}{2026-03-16}{0.4}
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 \clist_item:nn { Vn }
10 \cs_generate_variant:Nn \seq_set_split:Nnn { NVn, NVV }
11 \cs_generate_variant:Nn \seq_set_item:Nnn { NnV, Nne }
12 \cs_generate_variant:Nn \seq_gset_item:Nnn { NnV, Nne }
13 \cs_generate_variant:Nn \prop_gput:Nnn { Nne, Nen, Nee, NeV, NVe, NVV }
14 \cs_generate_variant:Nn \prop_gpop:NnN { NeN }
15 \cs_generate_variant:Nn \color_stroke:n { e }
16 \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。

```

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

```

定义 l3draw 函数的变体。

```

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

```

¹³同时使用以下环境验证与打包：L^AT_EX2e 的版本为 <2025-11-01>，L3 编程层的版本为 <2026-01-19>。

3.3 消息定义

本小节定义一些常见的错误消息。

```

23 \msg_new:nnn {xtable} {unknown_row_name} {未知的行名称: <#1>}
24 \msg_new:nnn {xtable} {unknown_col_name} {未知的列名称: <#1>}
25 \msg_new:nnn {xtable} {unknown_input} {未知的输入: <#1>}
26 \msg_new:nnn {xtable} {unknown_format} {未知的格式: <#1>}
27 \msg_new:nnn {xtable} {unknown_cell} {未知的单元格数据 @<#1>}
28 \msg_new:nnn {xtable} {merged_cell} {已经合并的单元格 @<#1>}
29 \msg_new:nnn {xtable} {unmerged_cell} {未合并的单元格 @<#1>}
30 \msg_new:nnn {xtable} {outside_xtable} {<#1> 应当在 xtable 环境中使用}
31 \msg_new:nnn {xtable} {outside_render} {<#1> 应当在 \rendertable 命令中使用}
32 \msg_new:nnn {xtable} {unsaved_table} {未保存的表: <#1>}

```

添加英文支持。

```

33 \cs_new:Nn \__xtable_en_msg_support:
34 {
35   \msg_set:nnn {xtable} {unknown_row_name} {Unknown row name: <##1>}
36   \msg_set:nnn {xtable} {unknown_col_name} {Unknown column name: <##1>}
37   \msg_set:nnn {xtable} {unknown_input} {Unknown input: <##1>}
38   \msg_set:nnn {xtable} {unknown_format} {Unknown format: <##1>}
39   \msg_set:nnn {xtable} {unknown_cell} {Unknown cell data @<##1>}
40   \msg_set:nnn {xtable} {merged_cell} {Already merged cell @<##1>}
41   \msg_set:nnn {xtable} {unmerged_cell} {Unmerged cell @<##1>}
42   \msg_set:nnn {xtable} {outside_xtable} {<##1> should be used inside xtable environment}
43   \msg_set:nnn {xtable} {outside_render} {<##1> should be used inside \rendertable comma
44   \msg_set:nnn {xtable} {unsaved_table} {Unsaved table: <##1>}
45 }

```

3.4 宏包选项

本小节定义宏包的选项及对应的变量。选项是唯一修改这些变量的地方。

```

\g__xtable_cell_sep_dim 行列间距与边距。
\g__xtable_row_margin_dim 46 \dim_new:N \g__xtable_cell_sep_dim % 单元格间的水平间距 (用于打印表格)
\g__xtable_col_margin_dim 47 \dim_new:N \g__xtable_row_margin_dim % 行边距与列边路 (用于渲染表格)
48 \dim_new:N \g__xtable_col_margin_dim

(End of definition for \g__xtable_cell_sep_dim, \g__xtable_row_margin_dim, and \g__xtable_col_margin_dim.)

\g__xtable_above_space_dim 表格上下边空、单元格最小宽度及单元格内行间距。
\g__xtable_below_space_dim 49 \dim_new:N \g__xtable_above_space_dim % 表格上下边空
\g__xtable_cell_wd_min_dim
\g__xtable_cell_lineskip_dim

```

```

50 \dim_new:N \g__xtable_below_space_dim
51 \dim_new:N \g__xtable_cell_wd_min_dim    % 单元格的最小宽度
52 \dim_new:N \g__xtable_cell_lineskip_dim  % 多行单元格的行距

```

(End of definition for \g__xtable_above_space_dim and others.)

`xtable/package` 定义宏包选项。

```

53 \keys_define:nn { xtable / package }
54 {
55   en      .code:n    = { \__xtable_en_msg_support: },
56   cellsep .dim_gset:N = \g__xtable_cell_sep_dim,
57   cellsep .initial:n = { 0.5em },
58   margin  .code:n    =
59   {
60     \dim_gset:Nn \g__xtable_row_margin_dim { \clist_item:nn {#1} {2} }
61     \dim_gset:Nn \g__xtable_col_margin_dim { \clist_item:nn {#1} {1} }
62   },
63   margin  .initial:n = { 0.4em, 0.6ex },
64   vspace  .code:n    =
65   {
66     \dim_gset:Nn \g__xtable_above_space_dim { \clist_item:nn {#1} {1} }
67     \dim_gset:Nn \g__xtable_below_space_dim { \clist_item:nn {#1} {2} }
68   },
69   vspace  .initial:n = { 0.5ex, -1ex },
70   minwidth .dim_gset:N = \g__xtable_cell_wd_min_dim,
71   minwidth .initial:n = { 1.5em },
72   lineskip .dim_gset:N = \g__xtable_cell_lineskip_dim,
73   lineskip .initial:n = { 3ex }
74 }
75 \ProcessKeyOptions [ xtable / package ]

```

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

3.5 通用变量

本小节定义一些通用的变量，供整个宏包中使用。

3.5.1 全局常量

`\c__xtable_space_str` 定义一些不易直接输入的字符常量。

```

\c__xtable_escape_str 76 \str_const:Ne \c__xtable_space_str { \char_generate:nn {32} {10} } % <space>
\c__xtable_lbrace_str 77 \str_const:Ne \c__xtable_escape_str { \char_generate:nn {92} {12} } % \
\c__xtable_rbrace_str 78 \str_const:Ne \c__xtable_lbrace_str { \char_generate:nn {123} {12} } % {
79 \str_const:Ne \c__xtable_rbrace_str { \char_generate:nn {125} {12} } % }

```

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

`\s__xtable_mark` 定义内部标记。

```
80 \scan_new:N \s__xtable_mark
```

(End of definition for `\s__xtable_mark`.)

`\c__xtable_std_ht_dim` 单行文字的标准高度与深度。

```
\c__xtable_std_dp_dim 81 \dim_const:Nn \c__xtable_std_ht_dim {1.91ex} % 约等于汉字的高度,字母高度 {1.67ex}
```

```
82 \dim_const:Nn \c__xtable_std_dp_dim {0.48ex} % 约等于字母的深度
```

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

`\c__xtable_merge_loc_prop` 合并单元格设置映射表。

```
\c__xtable_merge_align_prop 83 \prop_const_from_keyval:Nn \c__xtable_merge_loc_prop
84 {
85   ul = ul, ur = ur, dl = dl, du = du,
86   lu = ul, ru = ur, ld = dl, ud = du
87 }
88 \prop_const_from_keyval:Nn \c__xtable_merge_align_prop
89 {
90   tl = {t, l}, tc = {t, c}, tr = {t, r},
91   ml = {m, l}, mc = {m, c}, mr = {m, r},
92   bl = {b, l}, bc = {b, c}, br = {b, r},
93   lt = {t, l}, ct = {t, c}, rt = {t, r},
94   lm = {m, l}, cm = {m, c}, rm = {m, r},
95   lb = {b, l}, cb = {b, c}, rb = {b, r},
96   l = {m, l}, c = {m, c}, r = {m, r},
97   t = {t, c}, m = {m, c}, b = {b, c}
98 }
```

(End of definition for `\c__xtable_merge_loc_prop` and `\c__xtable_merge_align_prop`.)

3.5.2 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。

`\l__xtable_row_loc_int` 用于存储（行/列的）数字坐标信息。

```
\l__xtable_col_loc_int 99 \int_new:N \l__xtable_row_loc_int
```

```
100 \int_new:N \l__xtable_col_loc_int
```

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

`\l__xtable_is_box_bool` 用于存储单元格内容及状态的变量。

```
\l__xtable_data_tl 101 \bool_new:N \l__xtable_is_box_bool
102 \tl_new:N \l__xtable_data_tl
```

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

`\l__xtable_style_tl` 用于存储通用样式与对齐样式专用的凭据表。

```
\l__xtable_row_align_tl 103 \tl_new:N \l__xtable_style_tl
\l__xtable_col_align_tl 104 \tl_new:N \l__xtable_row_align_tl
105 \tl_new:N \l__xtable_col_align_tl
```

(End of definition for \l__xtable_style_tl, \l__xtable_row_align_tl, and \l__xtable_col_align_tl.)

`\l__xtable_merge_id_tl` 用于存储合并单元格索引的凭据表和内容的序列。

```
\l__xtable_merge_seq 106 \tl_new:N \l__xtable_merge_id_tl
107 \seq_new:N \l__xtable_merge_seq
```

(End of definition for \l__xtable_merge_id_tl and \l__xtable_merge_seq.)

`\l__xtable_wd_dim` 用于存储宽高深等尺寸信息。

```
\l__xtable_ht_dim 108 \dim_new:N \l__xtable_wd_dim
\l__xtable_dp_dim 109 \dim_new:N \l__xtable_ht_dim
\l__xtable_ufill_dim 110 \dim_new:N \l__xtable_dp_dim
\l__xtable_dfill_dim 111 \dim_new:N \l__xtable_ufill_dim
112 \dim_new:N \l__xtable_dfill_dim
```

(End of definition for \l__xtable_wd_dim and others.)

3.5.3 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，但不应用于函数间传递数据。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 113 \int_new:N \l__xtable_cachea_int
114 \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 115 \dim_new:N \l__xtable_cachea_dim
116 \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` 117 `\tl_new:N \l__xtable_cachea_tl`
 118 `\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` 119 `\seq_new:N \l__xtable_cachea_seq`
 120 `\seq_new:N \l__xtable_cacheb_seq`
(End of definition for \l__xtable_cachea_seq and \l__xtable_cacheb_seq.)

3.5.4 临时变量

本小节定义一些临时变量¹⁴，以供函数内部使用¹⁵。可以随意使用这些变量而无需在文档中声明¹⁶。

`\l__xtable_tmpa_int` 整数变量。
`\l__xtable_tmpb_int` 121 `\int_new:N \l__xtable_tmpa_int`
 122 `\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` 123 `\dim_new:N \l__xtable_tmpa_dim`
 124 `\dim_new:N \l__xtable_tmpb_dim`
(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` 125 `\tl_new:N \l__xtable_tmpa_tl`
`\l__xtable_tmpa_str` 126 `\tl_new:N \l__xtable_tmpb_tl`
`\l__xtable_tmpb_str` 127 `\str_new:N \l__xtable_tmpa_str`
 128 `\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` 129 `\seq_new:N \l__xtable_tmpa_seq`
`\l__xtable_tmpa_prop` 130 `\seq_new:N \l__xtable_tmpb_seq`
`\l__xtable_tmpb_prop` 131 `\prop_new:N \l__xtable_tmpa_prop`
 132 `\prop_new:N \l__xtable_tmpb_prop`

¹⁴不使用系统提供的临时变量，以防止和其它宏包冲突。

¹⁵不应当使用这些变量在函数之间传递数据。

¹⁶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_seq and others.)

```
\l__xtable_tmpa_box 盒子对象。
\l__xtable_tmpb_box 133 \box_new:N \l__xtable_tmpa_box
134 \box_new:N \l__xtable_tmpb_box
```

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

3.6 通用函数

本小节定义一些通用的函数，供整个宏包中使用。本小节只对外提供功能，不提供变量，也依赖外部变量。

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 135 \cs_new:Nn \__xtable_init_seq:Nnn
136 {
137   \seq_clear:N #1
138   \prg_replicate:nn {#2} { \seq_put_right:Nn #1 {#3} }
139 }
140 \cs_new:Nn \__xtable_ginit_seq:Nnn
141 {
142   \seq_gclear:N #1
143   \prg_replicate:nn {#2} { \seq_gput_right:Nn #1 {#3} }
144 }
145 \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 146 \cs_new:Nn \__xtable_int_set_max:Nn
147 { \int_set:Nn #1 { \int_max:nn {#1} {#2} } }
148 \cs_new:Nn \__xtable_int_gset_max:Nn
149 { \int_gset:Nn #1 { \int_max:nn {#1} {#2} } }
150 \cs_generate_variant:Nn \__xtable_int_gset_max:Nn {cn}
(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 151 \cs_new:Nn \__xtable_dim_set_max:NN
152 { \dim_set:Nn #1 { \dim_max:nn {#1} {#2} } }
153 \cs_new:Nn \__xtable_dim_set_max:NN
154 { \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 脚注函数

`\g__xtable_footnote_prop` 表格脚注支持专用变量。

```
155 \prop_new:N \g__xtable_footnote_prop
```

(End of definition for \g__xtable_footnote_prop.)

`__xtable_origin_footnote:` 备份系统脚注函数。

```
\__xtable_origin_footnotemark: 156 \cs_gset_eq:NN \__xtable_origin_footnote: \footnote
157 \cs_gset_eq:NN \__xtable_origin_footnotemark: \footnotemark
```

(End of definition for __xtable_origin_footnote: and __xtable_origin_footnotemark:.)

`__xtable_footnote:` 只保留占位符的脚注函数。

```
\__xtable_footnotemark: 158 \NewDocumentCommand \__xtable_footnote: { 0{ } m }
159 { \__xtable_origin_footnotemark: [99] }
160 \NewDocumentCommand \__xtable_footnotemark: { 0{ } }
161 { \__xtable_origin_footnotemark: [99] }
```

(End of definition for __xtable_footnote: and __xtable_footnotemark:.)

`__xtable_footnote_support:` 脚注标签与内容拆分实现的脚注函数。

```
\__xtable_footnote_print: 162 \NewDocumentCommand \__xtable_footnote_support: { o m }
163 {
164   \IfNoValueTF {#1}
165   {
166     \__xtable_origin_footnotemark:
167     \prop_gput:Nen \g__xtable_footnote_prop {\thefootnote} {#2}
168   }
169   {
170     \__xtable_origin_footnotemark: [#1]
171     \prop_gput:Nnn \g__xtable_footnote_prop {#1} {#2}
172   }
173 }
174 \cs_new:Nn \__xtable_footnote_print:
175 {
176   \prop_map_inline:Nn \g__xtable_footnote_prop
177   { \footnotetext[##1] {##2} }
178 }
```

(End of definition for __xtable_footnote_support: and __xtable_footnote_print:.)

```

\__xtable_disable_sys_func: 切换脚注函数。
\__xtable_support_sys_func: 179 \cs_new:Nn \__xtable_disable_sys_func:
\__xtable_restore_sys_func: 180 {
181   \prop_gclear:N \g__xtable_footnote_prop
182   \cs_set_eq:NN \footnote \__xtable_footnote:
183   \cs_set_eq:NN \footnotemark \__xtable_footnotemark:
184 }
185 \cs_new:Nn \__xtable_support_sys_func:
186 {
187   \prop_gclear:N \g__xtable_footnote_prop
188   \cs_set_eq:NN \footnote \__xtable_footnote_support:
189 }
190 \cs_new:Nn \__xtable_restore_sys_func:
191 {
192   \__xtable_footnote_print:
193   \cs_set_eq:NN \footnote \__xtable_origin_footnote:
194   \cs_set_eq:NN \footnotemark \__xtable_origin_footnotemark:
195 }

```

(End of definition for __xtable_disable_sys_func:, __xtable_support_sys_func:, and __xtable_restore_sys_func:.)

3.6.3 临时函数

__xtable_tmpa_cs: 两个临时函数，用于临时保存函数的定义。

```

\__xtable_tmpb_cs: 196 \cs_new:Nn \__xtable_tmpa_cs: {}
197 \cs_new:Nn \__xtable_tmpb_cs: {}

```

(End of definition for __xtable_tmpa_cs: and __xtable_tmpb_cs:.)

3.7 边框线样式

本小节提供边框线的设置及应用功能。样式是全局设置，可以在表格渲染中应用样式。本小节函数不依赖外部变量，外面的函数也不应当修改本小节的变量。

3.7.1 专用变量

\g__xtable_rule_pattern_prop 用于全局存储表格线型、线宽与样式。样式的值为 {线型, 线宽, 颜色} (分隔符为 \g__xtable_rule_wd_prop \s__xtable_mark)。

```

\g__xtable_rule_style_prop 198 \prop_new:N \g__xtable_rule_pattern_prop
199 \prop_new:N \g__xtable_rule_wd_prop
200 \prop_new:N \g__xtable_rule_style_prop

```

(End of definition for `\g__xtable_rule_pattern_prop`, `\g__xtable_rule_wd_prop`, and `\g__xtable_rule_style_prop`.)

```

\l__xtable_rule_pattern_tl 用于临时存储表格线型、线宽与样式。
\l__xtable_rule_wd_tl      201 \tl_new:N \l__xtable_rule_pattern_tl
\l__xtable_rule_color_tl   202 \tl_new:N \l__xtable_rule_wd_tl
\l__xtable_rule_dim        203 \tl_new:N \l__xtable_rule_color_tl
\l__xtable_rule_style_tl   204 \tl_new:N \l__xtable_rule_style_tl
\l__xtable_rule_style_seq  205 \dim_new:N \l__xtable_rule_dim
                           206 \seq_new:N \l__xtable_rule_style_seq

```

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

3.7.2 样式设置

```

\rulepatternset 设置边框线的线宽、线型与样式。
\rulewdset      207 \NewDocumentCommand {\rulepatternset} { m m }
\rulestyleset   208 { \prop_gput:Nnn \g__xtable_rule_pattern_prop {#1} {#2} }
                209 \NewDocumentCommand {\rulewdset} { m m }
                210 { \prop_gput:Nnn \g__xtable_rule_wd_prop {#1} {#2} }
                211 \NewDocumentCommand {\rulestyleset} { m m m m }
                212 {
                213   \prop_get:NnNF \g__xtable_rule_pattern_prop {#2} \l__xtable_rule_pattern_tl
                214   { \tl_set:Nn \l__xtable_rule_pattern_tl {#2} }
                215   \prop_get:NnNF \g__xtable_rule_wd_prop {#3} \l__xtable_rule_wd_tl
                216   { \tl_set:Nn \l__xtable_rule_wd_tl {#3} }
                217   \prop_gput:Nne \g__xtable_rule_style_prop {#1}
                218   { \l__xtable_rule_pattern_tl \s__xtable_mark \l__xtable_rule_wd_tl \s__xtable_mark #4}
                219 }
                220 \rulepatternset {soild} {}
                221 \rulepatternset {dotted}{0.6em, 0.15em}
                222 \rulepatternset {dash} {0.6em, 0.15em, 0.05em, 0.15em}
                223 \rulewdset {normal} {1pt}
                224 \rulewdset {thick} {1.5pt}
                225 \rulewdset {thin} {0.75pt}
                226 \rulewdset {toprule} {0.08em}
                227 \rulewdset {midrule} {0.05em}
                228 \rulewdset {cmidrule} {0.03em}
                229 \rulewdset {bottomrule} {0.08em}
                230 \rulestyleset {normal} {soild} {normal} {black}
                231 \rulestyleset {dotted} {dotted} {normal} {black}
                232 \rulestyleset {dash} {dash} {normal} {black}
                233 \rulestyleset {bold} {soild} {thick} {black}

```

(End of definition for `\rulepatternset`, `\rulewdset`, and `\rulestyleset`. These functions are documented on page 10.)

3.7.3 应用样式

```

\__xtable_set_rule_pattern:n  设置当前线型、线宽及样式。
\__xtable_set_rule_wd:n      234 \cs_new:Nn \__xtable_set_rule_pattern:n
\__xtable_set_rule_style:n    235 {
                                236   \prop_get:NnNTF \g__xtable_rule_pattern_prop {#1} \l__xtable_rule_pattern_tl
                                237   { \draw_set_dash_pattern:Vn \l__xtable_rule_pattern_tl {0pt} }
                                238   { \draw_set_dash_pattern:nn {#1} {0pt} }
                                239 }
                                240 \cs_new:Nn \__xtable_set_rule_wd:n
                                241 {
                                242   \prop_get:NnNTF \g__xtable_rule_wd_prop {#1} \l__xtable_rule_wd_tl
                                243   { \draw_set_linewidth:V \l__xtable_rule_wd_tl }
                                244   { \draw_set_linewidth:n {#1} }
                                245 }
                                246 \cs_new:Nn \__xtable_set_rule_style:n
                                247 {
                                248   \prop_get:NnNTF \g__xtable_rule_style_prop {#1} \l__xtable_rule_style_tl
                                249   {
                                250     \seq_set_split:NnV \l__xtable_rule_style_seq
                                251     {\s__xtable_mark} \l__xtable_rule_style_tl
                                252     \draw_set_dash_pattern:en { \seq_item:Nn \l__xtable_rule_style_seq {1} } {0pt}
                                253     \draw_set_linewidth:e     { \seq_item:Nn \l__xtable_rule_style_seq {2} }
                                254     \color_stroke:e           { \seq_item:Nn \l__xtable_rule_style_seq {3} }
                                255   } {
                                256     \__xtable_set_rule_pattern:n {soild}
                                257     \__xtable_set_rule_wd:n {#1}
                                258     \color_stroke:e {black}
                                259   }
                                260 }
                                261 \cs_generate_variant:Nn \__xtable_set_rule_style:n {V}

```

(End of definition for `__xtable_set_rule_pattern:n`, `__xtable_set_rule_wd:n`, and `__xtable_set_rule_style:n`.)

```

\__xtable_parse_rule_style:n  解析边框线样式,并将结果分别保存在 \l__xtable_rule_pattern_tl、\l__xtable_
                                rule_wd_tl 与 \l__xtable_rule_color_tl 中。
                                262 \cs_new:Nn \__xtable_parse_rule_style:n
                                263 {
                                264   \prop_get:NnNTF \g__xtable_rule_style_prop {#1} \l__xtable_rule_style_tl

```

```

265     {
266         \seq_set_split:NnV \l__xtable_rule_style_seq
267         {\s__xtable_mark} \l__xtable_rule_style_tl
268         \tl_set:Ne \l__xtable_rule_pattern_tl { \seq_item:Nn \l__xtable_tmpa_seq {1} }
269         \tl_set:Ne \l__xtable_rule_wd_tl      { \seq_item:Nn \l__xtable_tmpa_seq {2} }
270         \tl_set:Ne \l__xtable_rule_color_tl   { \seq_item:Nn \l__xtable_tmpa_seq {3} }
271         \dim_set:Nn \l__xtable_rule_dim      { \l__xtable_rule_wd_tl }
272     }
273     {
274         \prop_get:NnNF \g__xtable_rule_wd_prop {#1} \l__xtable_rule_wd_tl
275         { \tl_set:Nn \l__xtable_rule_wd_tl {#1} }
276         \dim_set:Nn \l__xtable_rule_dim { \l__xtable_rule_wd_tl }
277         \tl_if_empty:NT \l__xtable_rule_pattern_tl
278         { \tl_set:Nn \l__xtable_rule_pattern_tl { soild } }
279         \tl_if_empty:NT \l__xtable_rule_color_tl
280         { \tl_set:Nn \l__xtable_rule_color_tl { black } }
281     }
282 }
283 \cs_generate_variant:Nn \__xtable_parse_rule_style:n {V}

```

(End of definition for __xtable_parse_rule_style:n.)

4 表格核心

本节定义了存储、设置及读取表格的位置、内容与格式的核心变量与函数。录入一个表格，应当先设置好内容，再设置格式，否则可能会报错。

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。在表格的内部存储中，统一使用数字坐标¹⁷为索引（键）；行名与列名仅用于输入接口。

4.1.1 行列属性

\g__xtable_row_header_bool 是否包含行列表头（Header）。

\g__xtable_col_header_bool 284 \bool_new:N \g__xtable_row_header_bool

285 \bool_new:N \g__xtable_col_header_bool

(End of definition for \g__xtable_row_header_bool and \g__xtable_col_header_bool.)

¹⁷数据从 1 开始，行/列表头的坐标为 0。

`\g__xtable_row_count_int` 最大行列数 (不统计 Header)。

`\g__xtable_col_count_int` 286 `\int_new:N \g__xtable_row_count_int`
287 `\int_new:N \g__xtable_col_count_int`
(*End of definition for \g__xtable_row_count_int and \g__xtable_col_count_int.*)

`\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
`\g__xtable_col_name_prop` 为索引的 (集合) 变量 (包含 Header)。
288 `\prop_new:N \g__xtable_row_name_prop`
289 `\prop_new:N \g__xtable_col_name_prop`
(*End of definition for \g__xtable_row_name_prop and \g__xtable_col_name_prop.*)

`\g__xtable_row_align_seq` 行列的对齐方式 (包含 Header)。

`\g__xtable_col_align_seq` 290 `\seq_new:N \g__xtable_row_align_seq`
291 `\seq_new:N \g__xtable_col_align_seq`
(*End of definition for \g__xtable_row_align_seq and \g__xtable_col_align_seq.*)

`\g__xtable_row_ht_style_seq` 行高与列宽样式 (包含 Header)。

`\g__xtable_col_wd_style_seq` 292 `\seq_new:N \g__xtable_row_ht_style_seq`
293 `\seq_new:N \g__xtable_col_wd_style_seq`
(*End of definition for \g__xtable_row_ht_style_seq and \g__xtable_col_wd_style_seq.*)

4.1.2 单元格内容

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

`\c__xtable_cell_formula_tl` 294 `\tl_const:Nn \c__xtable_cell_content_tl {content}`
`\c__xtable_cell_box_tl` 295 `\tl_const:Nn \c__xtable_cell_formula_tl {formula}`
296 `\tl_const:Nn \c__xtable_cell_box_tl {box}`
(*End of definition for \c__xtable_cell_content_tl, \c__xtable_cell_formula_tl, and \c__xtable_cell_box_tl.*)

`\g__xtable_cell_data_prop` 定义存储的单元格内容的属性表 (包含 Header)。
单元格内容: `row,col -> {type, content, formula}` (分隔符为 `\s__xtable-mark`)。
297 `\prop_new:N \g__xtable_cell_data_prop`
(*End of definition for \g__xtable_cell_data_prop.*)

4.1.3 合并单元格

`\g__xtable_merge_info_seq` 合并单元格引用映射表、信息表及对齐样式表。
`\g__xtable_merge_align_seq` 引用映射表: `row,col -> merge_id` (在信息表及样式表中的位置);
`\g__xtable_merge_ref_prop` 信息表: `{start_row, start_col, end_row, end_col, mapping_cell}`;
 对齐样式表: `{row_align, col_align}`。

```

298 \seq_new:N \g__xtable_merge_info_seq
299 \seq_new:N \g__xtable_merge_align_seq
300 \prop_new:N \g__xtable_merge_ref_prop

(End of definition for \g__xtable_merge_info_seq, \g__xtable_merge_align_seq, and \g__xtable_merge_ref_prop.)

```

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。

```

301 \cs_new:Nn \_xtable_set_excel_col_names:n
302 {
303   \int_step_inline:nn {#1}
304   {
305     \prop_gput:Nen \g__xtable_col_name_prop
306     { \int_to_Alph:n {##1} } {##1}
307   }
308 }

(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
309 \cs_new:Nn \_xtable_set_row_name:nN
310 {
311   \prop_gput:Nne \g__xtable_row_name_prop {#1} {\int_use:N #2}
312   \_xtable_int_gset_max:Nn \g__xtable_row_count_int {#2}
313 }
314 \cs_new:Nn \_xtable_set_col_name:nN
315 {
316   \prop_gput:Nne \g__xtable_col_name_prop {#1} {\int_use:N #2}
317   \_xtable_int_gset_max:Nn \g__xtable_col_count_int {#2}
318 }
319 \cs_generate_variant:Nn \_xtable_set_row_name:nN {VN}
320 \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_line_loc:NnNF` 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。

```

321 \prg_new_conditional:Nnn \_xtable_get_line_loc:NnN { T, F, TF }
322   {
323     \regex_match:nnTF {^-?[0-9]+$} {#2}
324     { \int_set:Nn #3 {#2} \prg_return_true: }
325     {
326       \prop_get:NnNTF #1 {#2} \l__xtable_tmpa_tl
327       { \int_set:Nn #3 { \l__xtable_tmpa_tl } \prg_return_true: }
328       { \prg_return_false: }
329     }
330   }

```

(End of definition for _xtable_get_line_loc:NnNF.)

`_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。

```

331 \cs_new:Nn \_xtable_parse_row_loc:n
332   {
333     \_xtable_get_line_loc:NnNF
334     \g__xtable_row_name_prop {#1} \l__xtable_row_loc_int
335     { \msg_error:nnn {xtable} {unknown_row_name} {#1} }
336   }
337 \cs_new:Nn \_xtable_parse_col_loc:n
338   {
339     \_xtable_get_line_loc:NnNF
340     \g__xtable_col_name_prop {#1} \l__xtable_col_loc_int
341     { \msg_error:nnn {xtable} {unknown_col_name} {#1} }
342   }
343 \cs_generate_variant:Nn \_xtable_parse_row_loc:n { e, V }
344 \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立映射, #1: 行名/列名。
`_xtable_parse_new_col_loc:n` 坐标保存在 `\l__xtable_row_loc_int` 或 `\l__xtable_col_loc_int` 中。

```

345 \cs_new:Nn \_xtable_parse_new_row_loc:n
346   {
347     \_xtable_get_line_loc:NnNF
348     \g__xtable_row_name_prop {#1} \l__xtable_row_loc_int
349     {

```

```

350     \int_if_zero:nT { \l__xtable_row_loc_int }
351     { \int_set:Nn \l__xtable_row_loc_int { \g__xtable_row_count_int + 1 } }
352     \__xtable_set_row_name:nN {#1} \l__xtable_row_loc_int
353   }
354 }
355 \cs_new:Nn \__xtable_parse_new_col_loc:n
356 {
357   \__xtable_get_line_loc:NnNF
358   \g__xtable_col_name_prop {#1} \l__xtable_col_loc_int
359   {
360     \int_if_zero:nT { \l__xtable_col_loc_int }
361     { \int_set:Nn \l__xtable_col_loc_int { \g__xtable_col_count_int + 1 } }
362     \__xtable_set_col_name:nN {#1} \l__xtable_col_loc_int
363   }
364 }
365 \cs_generate_variant:Nn \__xtable_parse_new_row_loc:n { e, V }
366 \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。

```

367 \cs_new:Nn \__xtable_parse_coord:n
368 {
369   \__xtable_parse_row_loc:e { \clist_item:nn {#1} {1} }
370   \__xtable_parse_col_loc:e { \clist_item:nn {#1} {2} }
371 }
372 \cs_new:Nn \__xtable_parse_new_coord:n
373 {
374   \__xtable_parse_new_row_loc:e { \clist_item:nn {#1} {1} }
375   \__xtable_parse_new_col_loc:e { \clist_item:nn {#1} {2} }
376 }

```

(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。

```

377 \cs_new:Nn \__xtable_table_init:
378 {

```

```

379 \int_gzero:N \g__xtable_row_count_int
380 \int_gzero:N \g__xtable_col_count_int
381 \seq_gclear:N \g__xtable_row_align_seq
382 \seq_gclear:N \g__xtable_col_align_seq
383 \seq_gclear:N \g__xtable_row_ht_style_seq
384 \seq_gclear:N \g__xtable_col_wd_style_seq
385 \seq_gclear:N \g__xtable_merge_info_seq
386 \seq_gclear:N \g__xtable_merge_align_seq
387 \prop_gclear:N \g__xtable_row_name_prop
388 \prop_gclear:N \g__xtable_col_name_prop
389 \prop_gclear:N \g__xtable_cell_data_prop
390 \prop_gclear:N \g__xtable_merge_ref_prop

391 \prop_gput:Nnn \g__xtable_row_name_prop {header} {0}
392 \prop_gput:Nnn \g__xtable_col_name_prop {title} {0}
393 \bool_gset_false:N \g__xtable_row_header_bool
394 \bool_gset_true:N \g__xtable_col_header_bool
395 }

```

(End of definition for __xtable_table_init:.)

__xtable_table_save:n 保存表格。

```

396 \cs_new:Nn \__xtable_table_save:n
397 {
398   \cs_if_exist:cF { g__xtable_row_count_#1_int }
399   {
400     \bool_new:c { g__xtable_row_header_#1_bool }
401     \bool_new:c { g__xtable_col_header_#1_bool }
402     \int_new:c { g__xtable_row_count_#1_int }
403     \int_new:c { g__xtable_col_count_#1_int }
404     \seq_new:c { g__xtable_row_align_#1_seq }
405     \seq_new:c { g__xtable_col_align_#1_seq }
406     \seq_new:c { g__xtable_row_ht_style_#1_seq }
407     \seq_new:c { g__xtable_col_wd_style_#1_seq }
408     \seq_new:c { g__xtable_merge_info_#1_seq }
409     \seq_new:c { g__xtable_merge_align_#1_seq }
410     \prop_new:c { g__xtable_row_name_#1_prop }
411     \prop_new:c { g__xtable_col_name_#1_prop }
412     \prop_new:c { g__xtable_cell_data_#1_prop }
413     \prop_new:c { g__xtable_merge_ref_#1_prop }
414   }
415   \bool_gset_eq:cN { g__xtable_row_header_#1_bool } \g__xtable_row_header_bool
416   \bool_gset_eq:cN { g__xtable_col_header_#1_bool } \g__xtable_col_header_bool
417   \int_gset_eq:cN { g__xtable_row_count_#1_int } \g__xtable_row_count_int

```

```

418 \int_gset_eq:cN { g__xtable_col_count_#1_int } \g__xtable_col_count_int
419 \seq_gset_eq:cN { g__xtable_row_align_#1_seq } \g__xtable_row_align_seq
420 \seq_gset_eq:cN { g__xtable_col_align_#1_seq } \g__xtable_col_align_seq
421 \seq_gset_eq:cN { g__xtable_row_ht_style_#1_seq } \g__xtable_row_ht_style_seq
422 \seq_gset_eq:cN { g__xtable_col_wd_style_#1_seq } \g__xtable_col_wd_style_seq
423 \seq_gset_eq:cN { g__xtable_merge_info_#1_seq } \g__xtable_merge_info_seq
424 \seq_gset_eq:cN { g__xtable_merge_align_#1_seq } \g__xtable_merge_align_seq
425 \prop_gset_eq:cN { g__xtable_row_name_#1_prop } \g__xtable_row_name_prop
426 \prop_gset_eq:cN { g__xtable_col_name_#1_prop } \g__xtable_col_name_prop
427 \prop_gset_eq:cN { g__xtable_cell_data_#1_prop } \g__xtable_cell_data_prop
428 \prop_gset_eq:cN { g__xtable_merge_ref_#1_prop } \g__xtable_merge_ref_prop
429 }

```

(End of definition for __xtable_table_save:n.)

__xtable_table_restore:n 恢复表格。

```

430 \cs_new:Nn \__xtable_table_restore:n
431 {
432   \cs_if_exist:cF { g__xtable_row_count_#1_int }
433     { \msg_error:nmn {xtable} {unsaved_table} {#1} }
434   \bool_gset_eq:Nc \g__xtable_row_header_bool { g__xtable_row_header_#1_bool }
435   \bool_gset_eq:Nc \g__xtable_col_header_bool { g__xtable_col_header_#1_bool }
436   \int_gset_eq:Nc \g__xtable_row_count_int { g__xtable_row_count_#1_int }
437   \int_gset_eq:Nc \g__xtable_col_count_int { g__xtable_col_count_#1_int }
438   \seq_gset_eq:Nc \g__xtable_row_align_seq { g__xtable_row_align_#1_seq }
439   \seq_gset_eq:Nc \g__xtable_col_align_seq { g__xtable_col_align_#1_seq }
440   \seq_gset_eq:Nc \g__xtable_row_ht_style_seq { g__xtable_row_ht_style_#1_seq }
441   \seq_gset_eq:Nc \g__xtable_col_wd_style_seq { g__xtable_col_wd_style_#1_seq }
442   \seq_gset_eq:Nc \g__xtable_merge_info_seq { g__xtable_merge_info_#1_seq }
443   \seq_gset_eq:Nc \g__xtable_merge_align_seq { g__xtable_merge_align_#1_seq }
444   \prop_gset_eq:Nc \g__xtable_row_name_prop { g__xtable_row_name_#1_prop }
445   \prop_gset_eq:Nc \g__xtable_col_name_prop { g__xtable_col_name_#1_prop }
446   \prop_gset_eq:Nc \g__xtable_cell_data_prop { g__xtable_cell_data_#1_prop }
447   \prop_gset_eq:Nc \g__xtable_merge_ref_prop { g__xtable_merge_ref_#1_prop }
448 }

```

(End of definition for __xtable_table_restore:n.)

4.3.2 内容设置

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

```

449 \cs_new:Nn \__xtable_set_cell:nmnn
450 {

```

```

451 \prop_gput:Nnn \g__xtable_cell_data_prop
452   {#1} {#2 \s__xtable_mark #3 \s__xtable_mark #4}
453 \__xtable_int_gset_max:Nn \g__xtable_row_count_int { \clist_item:nn{#1}{1} }
454 \__xtable_int_gset_max:Nn \g__xtable_col_count_int { \clist_item:nn{#1}{2} }
455 }
456 \cs_generate_variant:Nn \__xtable_set_cell:nnnn { nVnn, eVnn }

```

(End of definition for __xtable_set_cell:nnnn.)

__xtable_set_cell:nn 设置单元格内容。#1 数字坐标，#2 内容/公式/盒子。若内容为空，则直接清空。

```

\__xtable_set_cell_formula:nn 457 \cs_new:Nn \__xtable_set_cell:nn
\__xtable_set_cell_box:nn 458 {
459   \tl_if_empty:nTF {#2}
460     { \prop_gpop:NeN \g__xtable_cell_data_prop {#1} \l__xtable_tmpa_tl }
461     { \__xtable_set_cell:nVnn {#1} \c__xtable_cell_content_tl {#2} {} }
462 }
463 \cs_new:Nn \__xtable_set_cell_formula:nn
464   { \__xtable_set_cell:nVnn {#1} \c__xtable_cell_formula_tl {} {#2} }
465 \cs_new:Nn \__xtable_set_cell_box:nn
466   { \__xtable_set_cell:nVnn {#1} \c__xtable_cell_box_tl {#2} {} }
467 \cs_generate_variant:Nn \__xtable_set_cell:nn { en, nV, eV }
468 \cs_generate_variant:Nn \__xtable_set_cell_formula:nn { en }
469 \cs_generate_variant:Nn \__xtable_set_cell_box:nn { en }

```

(End of definition for __xtable_set_cell:nn, __xtable_set_cell_formula:nn, and __xtable_set_cell_box:nn.)

__xtable_set_cell:NNn 设置单元格内容。#1#2 数字坐标，#3 内容/公式/盒子。若内容为空，则直接清空。

```

\__xtable_set_cell_formula:NNn 470 \cs_new:Nn \__xtable_set_cell:NNn
\__xtable_set_cell_box:NNn 471 { \__xtable_set_cell:en { \int_use:N #1, \int_use:N #2 } {#3} }
472 \cs_new:Nn \__xtable_set_cell_formula:NNn
473   { \__xtable_set_cell_formula:en { \int_use:N #1, \int_use:N #2 } {#3} }
474 \cs_new:Nn \__xtable_set_cell_box:NNn
475   { \__xtable_set_cell_box:en { \int_use:N #1, \int_use:N #2 } {#3} }
476 \cs_generate_variant:Nn \__xtable_set_cell:NNn { NNV }
477 \cs_generate_variant:Nn \__xtable_set_cell_formula:NNn { NNV }
478 \cs_generate_variant:Nn \__xtable_set_cell_box:NNn { NNV }

```

(End of definition for __xtable_set_cell:NNn, __xtable_set_cell_formula:NNn, and __xtable_set_cell_box:NNn.)

__xtable_set_cell:n 设置单元格内容/公式/盒子。坐标使用 \l__xtable_row_loc_int 与 \l__xtable_col_loc_int。

```

\__xtable_set_cell_box:n 479 \cs_new:Nn \__xtable_set_cell:n

```

```

480 {
481   \_xtable_set_cell:NNn
482     \l__xtable_row_loc_int \l__xtable_col_loc_int {#1}
483 }
484 \cs_new:Nn \_xtable_set_cell_formula:n
485 {
486   \_xtable_set_cell_formula:NNn
487     \l__xtable_row_loc_int \l__xtable_col_loc_int {#1}
488 }
489 \cs_new:Nn \_xtable_set_cell_box:n
490 {
491   \_xtable_set_cell_box:NNn
492     \l__xtable_row_loc_int \l__xtable_col_loc_int {#1}
493 }

```

(End of definition for _xtable_set_cell:n, _xtable_set_cell_formula:n, and _xtable_set_cell_box:n.)

4.3.3 内容查询

_xtable_parse_cell:n 获取指定坐标的单元格的内容。结果保存在 \l__xtable_data_tl (内容) 与 \l__xtable_cell_is_box_bool (是否盒子) 中。

```

494 \cs_new:Nn \_xtable_parse_cell:n
495 {
496   \bool_set_false:N \l__xtable_is_box_bool
497   \prop_get:NnNTF \g__xtable_cell_data_prop {#1} \l__xtable_tmpa_tl
498   {
499     \seq_set_split:NnV \l__xtable_tmpa_seq {\s__xtable_mark} \l__xtable_tmpa_tl
500     \tl_set:Ne \l__xtable_tmpa_tl { \seq_item:Nn \l__xtable_tmpa_seq {1} }
501     \tl_set:Ne \l__xtable_data_tl { \seq_item:Nn \l__xtable_tmpa_seq {2} }
502     \str_if_eq:VVT \l__xtable_tmpa_tl \c__xtable_cell_box_tl
503       { \bool_set_true:N \l__xtable_is_box_bool }
504   }
505   { \tl_clear:N \l__xtable_data_tl } % 空单元格
506 }
507 \cs_generate_variant:Nn {\_xtable_parse_cell:n} {V, e}

```

(End of definition for _xtable_parse_cell:n.)

4.4 修改表格

4.4.1 表格重组

`_xtable_table_restructure:nn` 重新选择数据替换原数据表。#1: 新的行/列逗号分隔列表, #2: 行还是列。

```
508 \cs_new:Nn \_xtable\_table\_restructure:nn
509 {
510   \group\_begin:
511   \prop\_clear:N \l\_xtable\_tmpa\_prop % new -> old
512   \prop\_clear:N \l\_xtable\_tmpb\_prop % old -> new
513   \prop\_clear:N \l\_xtable\_cachea\_prop % new name prop
514   \prop\_clear:N \l\_xtable\_cacheb\_prop % new data prop
515   \seq\_set\_split:Nnn \l\_xtable\_tmpa\_seq {,} {#1} % new col list
```

确认映射表与新的行/列数。

```
516   \int\_gset:cn {g\_xtable\_#2\_count\_int}
517   { \seq\_count:N \l\_xtable\_tmpa\_seq }
518   \prop\_gput:Nnn \l\_xtable\_tmpa\_prop {0} {0}
519   \prop\_gput:Nnn \l\_xtable\_tmpb\_prop {0} {0}
520   \int\_step\_inline:nn { \seq\_count:N \l\_xtable\_tmpa\_seq }
521   {
522     \tl\_set:Ne \l\_xtable\_tmpa\_tl
523     { \seq\_item:Nn \l\_xtable\_tmpa\_seq {##1} }
524     \prop\_gput:NnV \l\_xtable\_tmpa\_prop {##1} \l\_xtable\_tmpa\_tl
525     \prop\_gput:NVn \l\_xtable\_tmpb\_prop \l\_xtable\_tmpa\_tl {##1}
526   }
```

更新名称属性表。

```
527   \prop\_map\_inline:cn { g\_xtable\_#2\_name\_prop }
528   {
529     \prop\_get:NnTF \l\_xtable\_tmpb\_prop {##2} \l\_xtable\_tmpa\_tl
530     { \prop\_put:NnV \l\_xtable\_cachea\_prop {##1} \l\_xtable\_tmpa\_tl }
531     {
532       \int\_compare:nNnT {##2} = {0}
533       { \prop\_put:Nnn \l\_xtable\_cachea\_prop {##1} {##2} }
534     }
535   }
536   \prop\_gset\_eq:cN { g\_xtable\_#2\_name\_prop } \l\_xtable\_cachea\_prop
```

更新实际数据。

```
537   \str\_if\_eq:nnTF {#2} {row}
538   { \cs\_set\_eq:NN \_xtable\_tmpa\_cs: \l\_xtable\_tmpa\_str }
539   { \cs\_set\_eq:NN \_xtable\_tmpa\_cs: \l\_xtable\_tmpb\_str }
540   \prop\_map\_inline:nn \g\_xtable\_cell\_data\_prop
```

```

541     {
542         \str_set:Nc \l__xtable_tmpa_str { \clist_item:nn {##1} {1} }
543         \str_set:Nc \l__xtable_tmpb_str { \clist_item:nn {##1} {2} }
544         \prop_get:NnNT \l__xtable_tmpb_prop \__xtable_tmpa_cs: \__xtable_tmpa_cs:
545         {
546             \prop_put:Nen \l__xtable_cacheb_prop
547             { \l__xtable_tmpa_str, \l__xtable_tmpb_str } {##2}
548         }
549     }
550     \prop_gset_eq:NN \g__xtable_cell_data_prop \l__xtable_cacheb_prop
551     \group_end:
552 }

```

(End of definition for __xtable_table_restructure:nn.)

4.4.2 移动表格

__xtable_data_move:nnn 整行或整列移动单元格。#1: 起始位置，#2: 移动数量，#3: 行还是列。

```

553 \cs_new:Nn \__xtable_data_move:nnn
554 {
555     \prop_clear:N \l__xtable_tmpa_seq      % new data prop
556     \str_if_eq:nnTF {#3} {row}
557     {
558         \cs_set_eq:NN \__xtable_tmpa_cs: \l__xtable_tmpa_int      % 临时行索引
559         \cs_set_eq:NN \__xtable_tmpb_cs: \l__xtable_row_count_int % 行数
560     } {
561         \cs_set_eq:NN \__xtable_tmpa_cs: \l__xtable_tmpb_int      % 临时列索引
562         \cs_set_eq:NN \__xtable_tmpb_cs: \l__xtable_col_count_int % 列数
563     }

```

更新实际数据。

```

564     \prop_map_inline:nn \g__xtable_cell_data_prop
565     {
566         \int_set:Nn \l__xtable_tmpa_int { \clist_item:nn {##1} {1} }
567         \int_set:Nn \l__xtable_tmpb_int { \clist_item:nn {##1} {2} }
568         \int_compare:nNnTF {\__xtable_tmpa_cs:} < {#1}
569         % 如果有数据，说明其它单元格移动到此位置
570         { \prop_put_if_new:Nnn \l__xtable_tmpa_seq {##1} {##2} }
571         {
572             \int_add:Nn \__xtable_tmpa_cs: {#2}
573             \__xtable_int_gset_max:Nn \__xtable_tmpb_cs: { \__xtable_tmpa_cs: }
574             \prop_put:Nen \l__xtable_tmpa_seq
575             { \int_use:N \l__xtable_tmpa_int, \int_use:N \l__xtable_tmpb_int } {##2}

```

```

576     }
577   }
578   \prop_gset_eq:Nn \g__xtable_cell_data_prop \l__xtable_tmpa_seq
579 }

```

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

4.4.3 移动单元格

`__xtable_cell_move_by_row:nnn` 移动单元格。#1: 起始行，#2: 起始列，#3: 移动数量。

```

\__xtable_cell_move_by_col:nnn
580 \cs_new:Nn \__xtable_cell_move_by_row:nnn
581 {
582   \int_compare:nNnTF {#3} < {0}
583     { \int_step_inline:nnn {#1} {\g__xtable_row_count_int} }
584     { \int_step_inline:nnnn {\g__xtable_row_count_int} {#1} {-1} }
585     { % 此内容为映射函数的最后一个参数
586       \prop_gpop:NnNT \g__xtable_cell_data_prop {##1,#2} \l__xtable_tmpa_tl
587       {
588         \int_set:Nn \l__xtable_tmpa_int {##1 + #3}
589         \prop_gput:NeV \g__xtable_cell_data_prop
590           { \int_use:N \l__xtable_tmpa_int, #2 } \l__xtable_tmpa_tl
591         \__xtable_int_gset_max:Nn \l__xtable_row_count_int { \l__xtable_tmpa_int }
592       }
593     }
594 }

595 \cs_new:Nn \__xtable_cell_move_by_col:nnn
596 {
597   \int_compare:nNnTF {#3} < {0}
598     { \int_step_inline:nnn {#2} {\g__xtable_col_count_int} }
599     { \int_step_inline:nnnn {\g__xtable_col_count_int} {#2} {-1} }
600     { % 此内容为映射函数的最后一个参数
601       \prop_gpop:NnNT \g__xtable_cell_data_prop {#1,##1} \l__xtable_tmpa_tl
602       {
603         \int_set:Nn \l__xtable_tmpa_int {##1 + #3}
604         \prop_gput:NeV \g__xtable_cell_data_prop
605           { #1, \int_use:N \l__xtable_tmpa_int } \l__xtable_tmpa_tl
606         \__xtable_int_gset_max:Nn \l__xtable_col_count_int { \l__xtable_tmpa_int }
607       }
608     }
609 }

```

(End of definition for `__xtable_cell_move_by_row:nnn` and `__xtable_cell_move_by_col:nnn`.)

4.5 格式处理

4.5.1 样式设置

`_xtable_set_line_style:NNNn` 设置行/列样式。#1: 行样式名, #2: 样式内容, #3: 默认值, #4: 行或列。本命令会修改变量 `\l_xtable_cachea_int`、`\l_xtable_row_loc_int` 及 `\l_xtable_col_loc_int` 的值。

```

610 \cs_new:Nn \_xtable_set_line_style:NNNn
611 {
612   \bool_if:cTF
613     {
614       \str_if_eq:nnTF {#4} {row}
615         { g_xtable_col_header_bool } { g_xtable_row_header_bool }
616     }
617   { \int_set:Nn \l_xtable_tmpa_int {0} }
618   { \int_set:Nn \l_xtable_tmpa_int {1} }
619   \_xtable_int_gset_max:cn { g_xtable_#4_count_int }
620   { \seq_count:N #1 - 1 + \l_xtable_tmpa_int }
621   \_xtable_ginit_seq:Nnn #1 { \int_use:c { g_xtable_#4_count_int } + 1 } {#3}
622   \cs_set_eq:Nc \_xtable_tmpa_cs: { \_xtable_parse_#4_loc:V }
623   \seq_map_inline:Nn #2
624     {
625       \str_if_in:nnTF {##1} {=}
626       {
627         \seq_set_split:Nnn \l_xtable_tmpa_seq {=} {##1}
628         \seq_get_left:NN \l_xtable_tmpa_seq \l_xtable_tmpa_tl
629         \seq_get_right:NN \l_xtable_tmpa_seq \l_xtable_tmpb_tl
630         \_xtable_tmpa_cs: \l_xtable_tmpa_tl
631         \int_set_eq:Nc \l_xtable_tmpa_int { \l_xtable_#4_loc_int }
632       }
633       { \tl_set:Nn \l_xtable_tmpb_tl {##1} }
634       \seq_gset_item:NnV #1
635         { \l_xtable_tmpa_int + 1 } \l_xtable_tmpb_tl
636       \int_incr:N \l_xtable_tmpa_int
637     }
638 }

```

(End of definition for _xtable_set_line_style:NNNn.)

`_xtable_set_row_align:Nn` 设置行高样式。#1: 样式内容, #2: 默认值。

```

\__xtable_set_col_align:Nn
639 \cs_new:Nn \_xtable_set_row_align:Nn
640 { \_xtable_set_line_style:NNNn |g_xtable_row_align_seq #1 {#2} {row} }
641 \cs_new:Nn \_xtable_set_col_align:Nn

```

```
642 { \_xtable_set_line_style:NNnn \g__xtable_col_align_seq #1 {#2} {col} }
```

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

_xtable_set_row_ht_style:Nnn 设置行高的样式。#1: 样式内容, #2: 默认值, #3: 备用。

```
\_xtable_set_col_wd_style:Nnn 643 \cs_new:Nn \_xtable_set_row_ht_style:Nnn
644 {
645   \_xtable_set_line_style:NNnn \g__xtable_row_ht_style_seq #1 {#2} {row}
646   \int_step_inline:nn
647     { \seq_count:N \g__xtable_row_ht_style_seq }
648     {
649       \tl_set:Ne \l__xtable_tmpa_tl
650         { \seq_item:Nn \g__xtable_row_ht_style_seq {##1} }
651       \tl_replace_all:Nnn \l__xtable_tmpa_tl {auto} {a}
652       \tl_replace_all:Nnn \l__xtable_tmpa_tl {same} {s}
653       \seq_gset_item:NnV \g__xtable_row_ht_style_seq {##1} \l__xtable_tmpa_tl
654     }
655   \seq_gput_right:Nn \g__xtable_row_ht_style_seq {#3}
656 }
```

设置列宽的样式。#1: 样式内容, #2: 默认值, #3: 总宽。

```
657 \cs_new:Nn \_xtable_set_col_wd_style:Nnn
658 {
659   \_xtable_set_line_style:NNnn \g__xtable_col_wd_style_seq #1 {#2} {col}
660   \int_step_inline:nn
661     { \seq_count:N \g__xtable_col_wd_style_seq }
662     {
663       \tl_set:Ne \l__xtable_tmpa_tl
664         { \seq_item:Nn \g__xtable_col_wd_style_seq {##1} }
665       \tl_replace_all:Nnn \l__xtable_tmpa_tl {auto} {a}
666       \tl_replace_all:Nnn \l__xtable_tmpa_tl {same} {s}
667       \tl_replace_all:Nnn \l__xtable_tmpa_tl {fill} {f}
668       \seq_gset_item:NnV \g__xtable_col_wd_style_seq {##1} \l__xtable_tmpa_tl
669     }
670   \seq_gput_right:Nn \g__xtable_col_wd_style_seq {#3}
671 }
```

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

_xtable_insure_style: 确保样式有设置好。

```
672 \cs_new:Nn \_xtable_insure_style:
673 {
674   \seq_if_empty:NT \g__xtable_row_align_seq
675     { \_xtable_set_row_align:Nn \c_empty_seq {b} }
```

```

676 \seq_if_empty:NT \g__xtable_col_align_seq
677   { \__xtable_set_col_align:Nn \c_empty_seq {c} }
678 \seq_if_empty:NT \g__xtable_row_ht_style_seq
679   { \__xtable_set_row_ht_style:Nnn \c_empty_seq {a} {} }
680 \seq_if_empty:NT \g__xtable_col_wd_style_seq
681   { \__xtable_set_col_wd_style:Nnn \c_empty_seq {a} {\textwidth} }
682 }

```

(End of definition for __xtable_insure_style:.)

4.5.2 样式查询

__xtable_parse_row_align:n 解析行列的对齐样式。#1: 数字坐标，结果保存在 \l__xtable_row_align_tl 与 \l__xtable_parse_col_align:n \l__xtable_col_align_tl 中。

```

683 \cs_new:Nn \__xtable_parse_row_align:n
684   {
685     \tl_set:Ne \l__xtable_row_align_tl
686       { \seq_item:Nn \g__xtable_row_align_seq {#1 + 1} }
687   }
688 \cs_new:Nn \__xtable_parse_col_align:n
689   {
690     \tl_set:Ne \l__xtable_col_align_tl
691       { \seq_item:Nn \g__xtable_col_align_seq {#1 + 1} }
692   }

```

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

__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 693 \cs_new:Nn \__xtable_parse_row_ht_style:n
694   {
695     \tl_set:Ne \l__xtable_style_tl
696       { \seq_item:Nn \g__xtable_row_ht_style_seq {#1 + 1} }
697   }
698 \cs_new:Nn \__xtable_parse_col_wd_style:n
699   {
700     \tl_set:Ne \l__xtable_style_tl
701       { \seq_item:Nn \g__xtable_col_wd_style_seq {#1 + 1} }
702   }

```

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

4.6 合并单元格

4.6.1 合并设置

`_xtable_check_merge_range:nmmn` 检查指定区域内的所有单元格是否都未被合并。#1: 起始行, #2: 起始列, #3: 结束行, #4: 结束列。如果有合并, 则报错。

```

703 \cs_new:Nn \_xtable\_check\_merge\_range:nmmn
704 {
705   \int_step_inline:nnn {#1} {#3}
706   {
707     \int_step_inline:nnn {#2} {#4}
708     {
709       \prop_if_in:NnT \g\_xtable\_merge\_ref\_prop {##1,###1}
710       { \msg_error:nnn {xtable} {merged\_cell} {##1,###1} }
711     }
712   }
713 }
```

(End of definition for _xtable_check_merge_range:nmmn.)

`_xtable_set_merge_cell:nmmmm` 合并单元格。#1: 起始行, #2: 起始列, #3: 结束行, #4: 结束列, #5: 合并后取哪个单元格的內容。

```

714 \cs_new:Nn \_xtable\_set\_merge\_cell:nmmmm
715 {
716   \_xtable\_check\_merge\_range:nmmn {#1} {#2} {#3} {#4}
717   \int_set:Nn \l\_xtable\_tmpa\_int
718   { \seq_count:N \g\_xtable\_merge\_info\_seq + 1 }
719   \prop_get:NnNF \c\_xtable\_merge\_align\_prop {#5} \l\_xtable\_tmpa\_tl
720   { \tl_set:Nn \l\_xtable\_tmpa\_tl {} }
721   \seq_gput_right:Ne \g\_xtable\_merge\_info\_seq
722   {
723     #1 \s\_xtable\_mark #2 \s\_xtable\_mark
724     #3 \s\_xtable\_mark #4 \s\_xtable\_mark
725     \str_case:VnF \l\_xtable\_tmpa\_tl
726     {
727       {ul} { #1, #2 }
728       {ur} { #3, #2 }
729       {dl} { #1, #4 }
730       {dr} { #3, #4 }
731     }
732     { #1, #2 }
733   }
734   \int_step_inline:nnn {#1} {#3}
```

```

735     {
736       \int_step_inline:nnn {#2} {#4}
737       {
738         \prop_gput:Nne \g__xtable_merge_ref_prop
739           {##1,###1} { \int_use:N \l__xtable_tmpa_int }
740       }
741     }
742   }
743 \cs_generate_variant:Nn \__xtable_set_merge_cell:nnnnn { VVVVn }

```

(End of definition for __xtable_set_merge_cell:nnnnn.)

__xtable_set_merge_align:n 指定当前合并单元格的对齐格式。

```

744 \cs_new:Nn \__xtable_set_merge_align:n
745   {
746     \int_set:Nn \l__xtable_tmpa_int
747       { \seq_count:N \g__xtable_merge_align_seq + 1 }
748     \int_set:Nn \l__xtable_tmpb_int
749       { \seq_count:N \g__xtable_merge_info_seq - 1 }
750     \int_step_inline:nnn { \l__xtable_tmpa_int } { \l__xtable_tmpb_int }
751       { \seq_gput_right:Nn \g__xtable_merge_align_seq {m, c} }
752     \prop_get:NnNF \c__xtable_merge_align_prop {#1} \l__xtable_tmpa_tl
753       { \tl_set:Nn \l__xtable_tmpa_tl {m, c} }
754     \seq_gput_right:NV \g__xtable_merge_align_seq \l__xtable_tmpa_tl
755   }

```

(End of definition for __xtable_set_merge_align:n.)

4.6.2 合并查询

__xtable_merge_if:n 确认指定单元格是否已合并。#1: 数字坐标。

```

756 \prg_new_conditional:Nnn \__xtable_merge_if:n { p, T, F, TF }
757   {
758     \prop_if_in:NnTF \g__xtable_merge_ref_prop {#1}
759       { \prg_return_true: } { \prg_return_false: }
760   }

```

(End of definition for __xtable_merge_if:n.)

__xtable_parse_merge_count: 解析当前合并单元格数量，结果直接留在输入流中。

```

761 \cs_new:Nn \__xtable_parse_merge_count:
762   { \seq_count:N \g__xtable_merge_info_seq }

```

(End of definition for __xtable_parse_merge_count:.)

`_xtable_parse_merge_id:n` 解析指定坐标所处的合并单元格的索引。修改变量 `\l__xtable_merge_id_tl`。

```
763 \cs_new:Nn \_xtable_parse_merge_id:n
764 {
765   \prop_get:NnNF \g__xtable_merge_ref_prop {#1} \l__xtable_merge_id_tl
766   { \msg_error:nnn {xtable} {unmerged_cell} {#1} }
767 }
```

(End of definition for _xtable_parse_merge_id:n.)

`_xtable_parse_merge:n` 解析指定索引的合并单元格的内容。修改变量 `\l__xtable_merge_seq`、`\l__xtable_data_tl` 及 `\l__xtable_is_box_bool`。

```
768 \cs_new:Nn \_xtable_parse_merge:n
769 {
770   \tl_set:Ne \l__xtable_tmpa_tl
771   { \seq_item:Nn \g__xtable_merge_info_seq {#1} }
772   \seq_set_split:NnV \l__xtable_merge_seq {\s__xtable_mark} \l__xtable_tmpa_tl
773   \_xtable_parse_cell:e { \seq_item:Nn \l__xtable_merge_seq {5} }
774 }
775 \cs_generate_variant:Nn \_xtable_parse_merge:n { V }
```

(End of definition for _xtable_parse_merge:n.)

`_xtable_parse_merge_align:n` 解析指定索引的合并单元格的内容。`\l__xtable_row_align_tl` 与 `\l__xtable_col_align_tl` 中。

```
776 \cs_new:Nn \_xtable_parse_merge_align:n
777 {
778   \tl_set:Ne \l__xtable_tmpa_tl
779   { \seq_item:Nn \g__xtable_merge_align_seq {#1} }
780   \tl_set:Ne \l__xtable_row_align_tl
781   { \clist_item:Vn \l__xtable_tmpa_tl {1} }
782   \tl_set:Ne \l__xtable_col_align_tl
783   { \clist_item:Vn \l__xtable_tmpa_tl {2} }
784 }
```

(End of definition for _xtable_parse_merge_align:n.)

4.7 表头填充

`_xtable_header_fill:` 当表格的表头为空时，尝试用行列名替代。

```
785 \cs_new:Nn \_xtable_header_fill:
786 {
787   \prop_clear:N \l__xtable_tmpa_prop
788   \prop_map_inline:Nn \g__xtable_row_name_prop
```

```

789     { \prop_put:Nnn \l__xtable_tmpa_prop {##2, 0} {##1} }
790 \prop_map_inline:Nn \g__xtable_col_name_prop
791     { \prop_put:Nnn \l__xtable_tmpa_prop {0, ##2} {##1} }
792 \prop_map_inline:Nn \l__xtable_tmpa_prop
793     {
794     \tl_set:Nn \l__xtable_tmpa_tl { \s__xtable_mark ##2 \s__xtable_mark }
795     \tl_put_left:NV \l__xtable_tmpa_tl \c__xtable_cell_content_tl
796     \prop_gput_if_new:NnV \g__xtable_cell_data_prop {##1} \l__xtable_tmpa_tl
797     }
798 \int_step_inline:nn { \g__xtable_row_count_int }
799     {
800     \tl_set:No \l__xtable_tmpa_tl
801     { \c__xtable_cell_content_tl \s__xtable_mark ##1 \s__xtable_mark }
802     \prop_gput_if_new:NnV \g__xtable_cell_data_prop {##1, 0} \l__xtable_tmpa_tl
803     }
804 }

```

(End of definition for `__xtable_header_fill:`.)

4.8 调试

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

```

805 \NewDocumentCommand \showtable {}
806 {

```

输出列标题。

```

807     \prop_if_empty:NTF \g__xtable_cell_data_prop
808     {当前表格内容为空。}
809     {当前表格内容如下：}

```

输出内容。

```

810     \int_step_inline:nnn {0} { \g__xtable_row_count_int }
811     {
812     \<##1>~
813     \int_step_inline:nnn {0} { \g__xtable_col_count_int }
814     {
815     \__xtable_parse_cell:n {##1, #####1}
816     ,~\l__xtable_data_tl
817     }
818     ;
819     }
820 }

```

(End of definition for `\showtable`. This function is documented on page 10.)

`\logtable` 在日志中显示表格的核心数据。

```

821 \NewDocumentCommand \logtable {}
822 {
823   \int_log:N \g__xtable_row_count_int
824   \int_log:N \g__xtable_col_count_int
825   \seq_log:N \g__xtable_row_align_seq
826   \seq_log:N \g__xtable_col_align_seq
827   \seq_log:N \g__xtable_row_ht_style_seq
828   \seq_log:N \g__xtable_col_wd_style_seq
829   \seq_log:N \g__xtable_merge_info_seq
830   \seq_log:N \g__xtable_merge_align_seq
831   \prop_log:N \g__xtable_row_name_prop
832   \prop_log:N \g__xtable_col_name_prop
833   \prop_log:N \g__xtable_cell_data_prop
834   \prop_log:N \g__xtable_merge_ref_prop
835   \bool_log:N \g__xtable_row_header_bool
836   \bool_log:N \g__xtable_col_header_bool
837 }

```

(End of definition for `\logtable`. This function is documented on page 10.)

5 输入接口

本节作为数据输入的对外接口，负责将表格内容从用户层转到存储层，但不涉及渲染相关计算。

5.1 变量与选项

5.1.1 正则常量

`\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。

```

838 \regex_const:Nn \c__xtable_csv_row_regex
839 { (?:(?:[^\c{\}\}]+|"(?:"|"|"[^"])+")+?(?:\c{\}\}) }
840 \regex_const:Nn \c__xtable_csv_cell_regex
841 { (?:([^\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。

```

842 \regex_const:Nn \c__xtable_json_cell_regex

```

```

843 {
844   "((?:[~"\\]|\\.)+)"
845   \s*:\s*
846   (?:
847     ([\+~]?(?:\d*\.)?\d+|true|false) |
848     "((?:[~"\\]|\\.)+)"
849   )
850   \s*[,]?
851 }

```

(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 852 \regex_const:Nn \c__xtable_row_ht_regex
853 {
854   \A(?:.*=)?(?:
855     (auto|same|a|s)|
856     ([0-9.]+(?:pt|ex|em|mm|cm|in|cc|dd|pc))
857   )\Z
858 }
859 \regex_const:Nn \c__xtable_col_wd_regex
860 {
861   \A(?:.*=)?(?:
862     (auto|same|fill|samefill|a|s|f|sf)|
863     ([0-9.]+(?:pt|ex|em|mm|cm|in|cc|dd|pc))
864   )\Z
865 }

```

(End of definition for `\c__xtable_row_ht_regex` and `\c__xtable_col_wd_regex`.)

5.1.2 状态参数

`\l__xtable_input_bool` 当前是否处于输入环境。

```

866 \bool_new:N \l__xtable_input_bool
867 \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。

```

868 \cs_new:Nn \__xtable_check_input_env:n
869 {
870   \bool_if:NF \l__xtable_input_bool
871     { \msg_error:nnn {xtable} {outside_xtable} {#1} }
872 }

```

(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   873 \int_new:N \l__xtable_parse_status_int
\l__xtable_quote_status_bool  874 \bool_new:N \l__xtable_esc_status_bool
                               875 \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`.)

5.1.3 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  876 \seq_new:N \l__xtable_row_input_seq
                               877 \seq_new:N \l__xtable_cell_input_seq

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

5.1.4 局部选项

```
\l__xtable_input_format_tl  输入选项变量。
\l__xtable_input_sep_tl     878 \tl_new:N \l__xtable_input_format_tl
\l__xtable_input_row_header_bool  879 \tl_new:N \l__xtable_input_sep_tl
\l__xtable_input_col_header_bool  880 \bool_new:N \l__xtable_input_row_header_bool
                               881 \bool_new:N \l__xtable_input_col_header_bool

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

`xtable/input` 输入选项。

```
882 \keys_define:nn { xtable / input }
883 {
884   format    .choices:nn = { inner, csv, json }
885             { \tl_set_eq:NN \l__xtable_input_format_tl \l_keys_choice_tl },
886   format    .initial:n   = {inner},
887   csv       .code:n      = { \tl_set:Nn \l__xtable_input_format_tl { csv } },
888   json      .code:n      = { \tl_set:Nn \l__xtable_input_format_tl { json } },
889   title     .bool_set:N  = \l__xtable_input_col_header_bool,
890   header    .bool_set:N  = \l__xtable_input_col_header_bool,
891   header    .initial:n   = {true},
892   rowheader .bool_set:N  = \l__xtable_input_row_header_bool,
893   rowheader .initial:n   = {false},
894   sep       .tl_set:N    = \l__xtable_input_sep_tl,
895   sep       .initial:n   = {,},
896   loc       .code:n      = { \__xtable_parse_coord:n {#1} },
```

```

897     loc      .initial:n = {1,1}
898   }

```

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

5.2 数据解析

5.2.1 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。

```

899 \cs_new:Nn \_xtable_parse_input:n
900   {

```

准备阶段。

```

901     \seq_set_split:Nnn \l__xtable_row_input_seq {\} {#1}
902     \cs_set:Nn \_xtable_tmpa_cs: % 更新单元格序列
903     {
904         \tl_replace_all:Nnn \l__xtable_tmpa_tl {\newline} {}
905         \seq_set_split:NVV \l__xtable_cell_input_seq
906         \l__xtable_input_sep_tl \l__xtable_tmpa_tl
907     }
908     \cs_set:Nn \_xtable_tmpb_cs: % 填充一行数据
909     {
910         \int_set_eq:NN \l__xtable_tmpb_int \l__xtable_col_loc_int
911         \seq_map_inline:Nn \l__xtable_cell_input_seq
912         {
913             \_xtable_set_cell:NNn \l__xtable_tmpa_int \l__xtable_tmpb_int {###1}
914             \int_incr:N \l__xtable_tmpb_int
915         }
916     }

```

处理标题行。

```

917     \bool_if:NT \l__xtable_input_col_header_bool
918     {
919         \int_zero:N \l__xtable_tmpa_int
920         \seq_pop_left:NN \l__xtable_row_input_seq \l__xtable_tmpa_tl
921         \_xtable_tmpa_cs:
922         \bool_if:NT \l__xtable_input_row_header_bool
923         {
924             \seq_pop_left:NNT \l__xtable_cell_input_seq \l__xtable_tmpb_tl
925             { \_xtable_set_cell:nV {0, 0} \l__xtable_tmpb_tl }
926         }
927         \_xtable_tmpb_cs:

```

```

928     \int_set_eq:NN \l__xtable_tmpb_int \l__xtable_col_loc_int
929     \seq_map_inline:Nn \l__xtable_cell_input_seq
930     {
931         \__xtable_set_col_name:nN {##1} \l__xtable_tmpb_int
932         \int_incr:N \l__xtable_tmpb_int
933     }
934 }

```

处理正常数据行。

```

935     \int_set_eq:NN \l__xtable_tmpa_int \l__xtable_row_loc_int
936     \seq_map_inline:Nn \l__xtable_row_input_seq
937     {
938         \tl_set:Nn \l__xtable_tmpa_tl {##1}
939         \__xtable_tmpa_cs:
940         \bool_if:NT \l__xtable_input_row_header_bool
941         {
942             \seq_pop_left:NNT \l__xtable_cell_input_seq \l__xtable_tmpb_tl
943             { \__xtable_set_cell:eV {\l__xtable_tmpa_int, 0} \l__xtable_tmpb_tl }
944         }
945         \__xtable_tmpb_cs:
946         \int_incr:N \l__xtable_tmpa_int
947     }
948 }

```

(End of definition for __xtable_parse_input:n.)

5.2.2 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。

```

949 \cs_new:Nn \__xtable_parse_csv:n
950 {
951     \seq_clear:N \l__xtable_row_input_seq
952     \regex_extract_all:NnN
953     \c__xtable_csv_row_regex {##1\\} \l__xtable_row_input_seq
954     \int_set_eq:NN \l__xtable_tmpa_int \l__xtable_row_loc_int
955     \seq_map_inline:Nn \l__xtable_row_input_seq
956     {
957         \int_set_eq:NN \l__xtable_tmpb_int \l__xtable_col_loc_int
958         \seq_clear:N \l__xtable_tmpa_seq

```

```

959     \seq_clear:N \l__xtable_cell_input_seq
960     \regex_extract_all:NnN
961     \c__xtable_csv_cell_regex {##1} \l__xtable_tmpa_seq
962     \int_step_inline:nnnn {2} {3} {\seq_count:N \l__xtable_tmpa_seq}
963     {
964       \tl_set:Ne \l__xtable_tmpa_tl
965       { \seq_item:Nn \l__xtable_tmpa_seq {####1} }
966       \tl_set:Ne \l__xtable_tmpb_tl
967       { \seq_item:Nn \l__xtable_tmpa_seq {####1+1} }
968       \tl_put_right:NV \l__xtable_tmpa_tl \l__xtable_tmpb_tl
969       \seq_put_right:NV \l__xtable_cell_input_seq \l__xtable_tmpa_tl
970     }
971     \bool_if:NTF \l__xtable_input_col_header_bool
972     { % 填充表头（设置列名）
973       \seq_map_inline:Nn \l__xtable_cell_input_seq
974       {
975         \__xtable_set_col_name:nN {####1} \l__xtable_tmpb_int
976         \int_incr:N \l__xtable_tmpb_int
977       }
978       \bool_set_false:N \l__xtable_input_col_header_bool
979     }
980     { % 填充数据
981       \seq_map_inline:Nn \l__xtable_cell_input_seq
982       {
983         \__xtable_set_cell:NNn \l__xtable_tmpa_int \l__xtable_tmpb_int {####1}
984         \int_incr:N \l__xtable_tmpb_int
985       }
986       \int_incr:N \l__xtable_tmpa_int
987     }
988   }
989 }

```

(End of definition for __xtable_parse_csv:n.)

5.2.3 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。

```

990 \cs_new:Nn \__xtable_parse_json_auxa:n
991 {
992   \str_case_e:nnF {#1}
993   {
994     {\c__xtable_space_str} {} % 空白, 跳过
995     {[} % 数组开始
996     { \int_set:Nn \l__xtable_parse_status_int {1} }
997     {\c__xtable_lbrace_str} % 行开始
998     {
999       \int_set:Nn \l__xtable_parse_status_int {2}
1000       \str_clear:N \l__xtable_tmpa_str
1001     }
1002   }
1003   { \msg_error:nnn {xtable} {unknown_input} {#1} }
1004 }

```

(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。

```

1005 \cs_new:Nn \__xtable_parse_json_auxb:n
1006 {
1007   \str_case_e:nnF {#1}
1008   {
1009     {\c__xtable_space_str} {} % 空白, 跳过
1010     {,} {} % 行分隔, 跳过
1011     []} % 数组结束
1012     { \int_set:Nn \l__xtable_parse_status_int {0} }
1013     {\c__xtable_lbrace_str} % 行开始
1014     {
1015       \int_set:Nn \l__xtable_parse_status_int {2}
1016       \str_clear:N \l__xtable_tmpa_str
1017     }
1018   }
1019   { \msg_error:nnn {xtable} {unknown_input} {#1} }
1020 }

```

(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。

```

1021 \cs_new:Nn \__xtable_parse_json_auxc:n
1022 {
1023   \bool_if:NTF \l__xtable_esc_status_bool % 转义状态
1024   { \bool_set_false:N \l__xtable_esc_status_bool }
1025   {
1026     \str_if_eq:VnTF \c__xtable_escape_str {#1} % 即将转义
1027     { \bool_set_true:N \l__xtable_esc_status_bool }
1028     {
1029       \bool_if:NTF \l__xtable_quote_status_bool % 引用状态
1030       {
1031         \str_if_eq:nnT {"} {#1} % 结束引用
1032         { \bool_set_false:N \l__xtable_quote_status_bool }
1033       }
1034       {
1035         \str_if_eq:nnTF {"} {#1} % 即将引用
1036         { \bool_set_true:N \l__xtable_quote_status_bool }
1037         {
1038           \str_if_eq:VnT \c__xtable_rbrace_str {#1} % 行结束
1039           {
1040             \int_set:Nn \l__xtable_parse_status_int {1}
1041             \seq_put_right:NV \l__xtable_row_input_seq \l__xtable_tmpa_str
1042           }
1043         }
1044       }
1045     }
1046   }
1047   \str_put_right:Nn \l__xtable_tmpa_str {#1}
1048 }

```

(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。

```

1049 \cs_new:Nn \__xtable_parse_json_row:n
1050 {
1051   \regex_extract_all:NnN
1052   \c__xtable_json_cell_regex {#1} \l__xtable_cell_input_seq
1053   \bool_until_do:nn
1054   { \seq_if_empty_p:N \l__xtable_cell_input_seq }

```

```

1055     {
1056         \seq_pop_left:NN \l__xtable_cell_input_seq \l__xtable_tmpa_tl % 丢弃
1057         \seq_pop_left:NN \l__xtable_cell_input_seq \l__xtable_cachea_tl
1058         \seq_pop_left:NN \l__xtable_cell_input_seq \l__xtable_tmpa_tl
1059         \seq_pop_left:NN \l__xtable_cell_input_seq \l__xtable_tmpb_tl
1060         \tl_if_empty:NTF \l__xtable_tmpa_tl
1061             { \tl_set_eq:NN \l__xtable_cacheb_tl \l__xtable_tmpb_tl }
1062             { \tl_set_eq:NN \l__xtable_cacheb_tl \l__xtable_tmpa_tl }
1063         \int_set_eq:NN \l__xtable_cachea_int \l__xtable_col_loc_int
1064         \__xtable_parse_new_col_loc:V \l__xtable_cachea_tl
1065         \int_set_eq:NN \l__xtable_tmpa_int \l__xtable_col_loc_int
1066         \int_compare:nNnTF {\l__xtable_cachea_int} = {\l__xtable_col_loc_int}
1067             { \int_incr:N \l__xtable_col_loc_int }
1068             { \int_set_eq:NN \l__xtable_col_loc_int \l__xtable_cachea_int }
1069         \__xtable_set_cell:NNV
1070             \l__xtable_row_loc_int \l__xtable_tmpa_int \l__xtable_cacheb_tl
1071     }
1072     \int_incr:N \l__xtable_row_loc_int
1073 }

```

(End of definition for __xtable_parse_json_row:n.)

__xtable_parse_json:n 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。

```

1074 \cs_new:Nn \__xtable_parse_json:n
1075     {
1076         \int_zero:N \l__xtable_parse_status_int
1077         \bool_set_false:N \l__xtable_esc_status_bool
1078         \bool_set_false:N \l__xtable_quote_status_bool
1079         \str_clear:N \l__xtable_tmpa_str
1080         \seq_clear:N \l__xtable_row_input_seq
1081         \str_map_inline:nn {#1} % 解析 JSON 中的行
1082         {
1083             \int_case:nn
1084                 { \l__xtable_parse_status_int }
1085                 {
1086                     {0} { \__xtable_parse_json_auxa:n {##1} } % 开始
1087                     {1} { \__xtable_parse_json_auxb:n {##1} } % 行间状态
1088                     {2} { \__xtable_parse_json_auxc:n {##1} } % 行内状态
1089                 }
1090         }
1091         \seq_map_function:NN \l__xtable_row_input_seq \__xtable_parse_json_row:n

```

```
1092 }
```

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

5.3 数据录入

`\loadtable` 保存与加载表格。

```
\savetable 1093 \NewDocumentCommand {\loadtable} { m }
1094 {
1095   \_xtable_check_input_env:n {\loadtable}
1096   \_xtable_table_restore:n {#1}
1097 }
1098 \NewDocumentCommand {\savetable} { m }
1099 {
1100   \_xtable_check_input_env:n {\savetable}
1101   \_xtable_table_save:n {#1}
1102 }
```

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

`\excelcolname` 设置 Excel 风格列名。

```
1103 \NewDocumentCommand {\excelcolname} { O{26} }
1104 {
1105   \_xtable_check_input_env:n {\excelcolname}
1106   \_xtable_set_excel_col_names:n {#1}
1107 }
```

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

`_xtable_line_name:Nnnn` 设置行名与列名。

```
\rowname 1108 \cs_new:Nn \_xtable_line_name:Nnnn
\colname 1109 {
1110   \int_set:Nn \l__xtable_tmpa_int {#2}
1111   \seq_set_split:Nnn \l__xtable_cell_input_seq {#3} {#4}
1112   \seq_map_inline:Nn \l__xtable_cell_input_seq
1113     {
1114       #1 {##1} \l__xtable_tmpa_int
1115       \int_incr:N \l__xtable_tmpa_int
1116     }
1117 }
```

#1 位置, #2 分隔符, #3 名称列表。

```
1118 \NewDocumentCommand {\rowname} { D(){} 0{,} m }
1119 {
```

```

1120   \_xtable_check_input_env:n {\rowname}
1121   \_xtable_line_name:Nnnn \_xtable_set_row_name:nN {#1} {#2} {#3}
1122 }
1123 \NewDocumentCommand {\colname} { D() {1} 0{,} m }
1124 {
1125   \_xtable_check_input_env:n {\colname}
1126   \_xtable_line_name:Nnnn \_xtable_set_col_name:nN {#1} {#2} {#3}
1127 }

```

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

`_xtable_line:nnnn` 设置某一行/列数据。使用变量 `\l__xtable_row_loc_int` 与 `\l__xtable_col_loc_int`。

```

1128 \cs_new:Nn \_xtable_line:nnnn
1129 {

```

解析位置。

```

1130   \int_set:Nn \l__xtable_row_loc_int {1}
1131   \int_set:Nn \l__xtable_col_loc_int {1}
1132   \str_if_empty:nTF {#1}
1133   {
1134     \int_set_eq:Nc \l__xtable_tmpa_int { g__xtable_#4_count_int }
1135     \int_add:cn { l__xtable_#4_loc_int } { \l__xtable_tmpa_int }
1136   }
1137   {
1138     \str_if_in:nnTF {#1} {,}
1139     { \_xtable_parse_new_coord:n {#1} }
1140     { \use:c { __xtable_parse_new_#4_loc:n } {#1} }
1141   }

```

遍历添加数据。

```

1142   \str_if_eq:nnTF {#4} {row}
1143   { \cs_set_eq:NN \_xtable_tmpa_cs: \l__xtable_col_loc_int }
1144   { \cs_set_eq:NN \_xtable_tmpa_cs: \l__xtable_row_loc_int }
1145   \seq_set_split:Nnn \l__xtable_cell_input_seq {#2} {#3}
1146   \seq_map_inline:Nn \l__xtable_cell_input_seq
1147   {
1148     \_xtable_set_cell:n {##1}
1149     \int_incr:N \_xtable_tmpa_cs:
1150   }
1151 }

```

定义内部名称。#1 位置，#2 分隔符，#3 名称列表。

```

1152 \NewDocumentCommand {\__xtable_row:nnn} { D(){} 0{,} m }
1153 { \__xtable_line:nnnn {#1} {#2} {#3} {row} }
1154 \NewDocumentCommand {\__xtable_col:nnn} { D(){} 0{,} m }
1155 { \__xtable_line:nnnn {#1} {#2} {#3} {col} }

```

(End of definition for __xtable_line:nnnn, __xtable_row:nnn, and __xtable_col:nnn.)

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

```

1156 \NewDocumentCommand {\__xtable_cell:nn} { r() m }
1157 {
1158   \int_zero:N \l__xtable_row_loc_int
1159   \int_zero:N \l__xtable_col_loc_int
1160   \__xtable_parse_new_coord:n {#1}
1161   \__xtable_set_cell:n {#2}
1162 }

```

(End of definition for __xtable_cell:nn.)

`@@_data:` (*env.*) 用于输入表格内容的环境。

```

1163 \NewDocumentEnvironment{\__xtable_data:}{0{} +b}
1164 {
1165   \keys_set:nn { xtable/input } {#1}
1166   \str_case:VnF \l__xtable_input_format_tl
1167   {
1168     {inner} { \__xtable_parse_input:n {#2} }
1169     {csv} { \__xtable_parse_csv:n {#2} }
1170     {json} { \__xtable_parse_json:n {#2} }
1171   }
1172   { \msg_error:nnV {xtable} {unknown_format} \l__xtable_input_format_tl }
1173 }
1174 {}

```

`xtable` (*env.*) 用于输入表格的环境。

```

data (env.) 1175 \NewDocumentEnvironment{xtable}{ 0{} }
\row 1176 {
\col 1177   \__xtable_table_init:
\cell 1178   \keys_set:nn { xtable/input } {#1}
1179   \bool_gset_eq:NN
1180     \g__xtable_row_header_bool \l__xtable_input_row_header_bool
1181   \bool_gset_eq:NN
1182     \g__xtable_col_header_bool \l__xtable_input_col_header_bool
1183   \cs_set_eq:NN \data \__xtable_data:

```

```

1184     \cs_set_eq:NN \enddata \end__xtable_data:
1185     \cs_set_eq:NN \row \__xtable_row:nnn
1186     \cs_set_eq:NN \col \__xtable_col:nnn
1187     \cs_set_eq:NN \cell \__xtable_cell:nn
1188     \bool_set_true:N \l__xtable_input_bool
1189   }
1190   {
1191     \bool_set_false:N \l__xtable_input_bool
1192     \__xtable_insure_style:
1193     \__xtable_header_fill:
1194   }

```

(End of definition for . and others. These functions are documented on page ??.)

5.4 格式设置

`__xtable_line_align:nnnn` 设置行与列的对齐方式。#1 默认值，#2 格式列表，#3 正则表达式，#4 行/列。

```

\rowalign 1195 \cs_new:Nn \__xtable_line_align:nnnn
\colalign 1196   {
1197     \regex_extract_all:nnNF {#3} {#1#2} \l__xtable_tmpa_seq
1198     { \msg_error:nnn {xtable} {unknown_format} {[#1],[#2]} }
1199     \str_if_in:nnTF {#2} {=}
1200     { \seq_set_split:Nnn \l__xtable_cell_input_seq {,} {#2} }
1201     {
1202       \tl_set:Nn \l__xtable_tmpa_tl {#2}
1203       \tl_replace_all:Nnn \l__xtable_tmpa_tl {,} {}
1204       \seq_set_split:NnV \l__xtable_cell_input_seq {} \l__xtable_tmpa_tl
1205     }
1206     \use:c { __xtable_set_#4_align:Nn } \l__xtable_cell_input_seq {#1}
1207   }
1208 \NewDocumentCommand {\rowalign} { 0{b} m }
1209   {
1210     \__xtable_check_input_env:n {\rowalign}
1211     \__xtable_line_align:nnnn {#1} {#2} {^(?:(:?\.*)?[tmb],?)*$} {row}
1212   }
1213 \NewDocumentCommand {\colalign} { 0{c} m }
1214   {
1215     \__xtable_check_input_env:n {\colalign}
1216     \__xtable_line_align:nnnn {#1} {#2} {^(?:(:?\.*)?[lcr],?)*$} {col}
1217   }

```

(End of definition for `__xtable_line_align:nnnn`, `\rowalign`, and `\colalign`. These functions are documented on page 7.)

```

\__xtable_line_size:nmmm 设置行高与列宽。#1 默认值，#2 总尺寸，#3 格式列表，#4 行高/列宽。
\rowheight 1218 \cs_new:Nn \__xtable_line_size:nmmm
\colwidth 1219 {
1220   \seq_clear:N \l__xtable_cell_input_seq
1221   \seq_set_split:Nnn \l__xtable_cell_input_seq {,} {#3}
1222   \cs_set_eq:Nc \__xtable_tmpa_cs: { c__xtable_#4_regex }
1223   \regex_extract_all:NnNF
1224     \__xtable_tmpa_cs: {#1} \l__xtable_tmpa_seq
1225     { \msg_error:nnn {xtable} {unknown_format} {#1} }
1226   \seq_map_inline:Nn \l__xtable_cell_input_seq
1227     {
1228       \regex_extract_all:NnNF
1229         \__xtable_tmpa_cs: {##1} \l__xtable_tmpa_seq
1230         { \msg_error:nnn {xtable} {unknown_format} {##1} }
1231     }
1232   \use:c { __xtable_set_#4_style:Nnn } \l__xtable_cell_input_seq {#1} {#2}
1233 }

1234 \NewDocumentCommand {\rowheight} { 0{auto} 0{\textheight} m }
1235 {
1236   \__xtable_check_input_env:n {\rowheight}
1237   \__xtable_line_size:nmmm {#1} {#2} {#3} { row_ht }
1238 }

1239 \NewDocumentCommand {\colwidth} { 0{auto} 0{\textwidth} m }
1240 {
1241   \__xtable_check_input_env:n {\colwidth}
1242   \__xtable_line_size:nmmm {#1} {#2} {#3} { col_wd }
1243 }

```

(End of definition for `__xtable_line_size:nmmm`, `\rowheight`, and `\colwidth`. These functions are documented on page 7.)

5.5 合并单元格

`\mergecell` 合并单元格。

```

1244 \NewDocumentCommand {\mergecell} { 0{ul} r() r() 0{mc} }
1245 {
1246   \__xtable_check_input_env:n {\mergecell}
1247   \cs_new:Npn \__xtable_tmpa_cs: ##1##2
1248     {
1249       \int_set_eq:NN \l__xtable_tmpa_int ##1
1250       \int_set_eq:NN ##1 ##2
1251       \int_set_eq:NN ##2 \l__xtable_tmpa_int

```

```

1252     }
1253     \__xtable_check_input_env:n {\mergecell}
1254     \__xtable_parse_coord:n {#3}
1255     \int_set_eq:NN \l__xtable_cachea_int \l__xtable_row_loc_int
1256     \int_set_eq:NN \l__xtable_cacheb_int \l__xtable_col_loc_int
1257     \__xtable_parse_coord:n {#2}
1258     \int_compare:nNnT {\l__xtable_row_loc_int} > {\l__xtable_cachea_int}
1259     { \__xtable_tmapa_cs: \l__xtable_row_loc_int \l__xtable_cachea_int }
1260     \int_compare:nNnT {\l__xtable_col_loc_int} > {\l__xtable_cacheb_int}
1261     { \__xtable_tmapa_cs: \l__xtable_col_loc_int \l__xtable_cacheb_int }
1262     \__xtable_set_merge_cell:VVVn \l__xtable_row_loc_int \l__xtable_col_loc_int
1263     \l__xtable_cachea_int \l__xtable_cacheb_int {#1}
1264     \__xtable_set_merge_align:n {#4}
1265 }

```

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

6 中间计算

本节内容为渲染输出准备，它计算一些表格相关的数据。

6.1 中间变量

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

```

\g__xtable_col_sep_seq 1266 \seq_new:N \g__xtable_row_sep_seq
1267 \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、深及宽信息（包含Header），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。

```

\g__xtable_col_wd_seq 1268 \seq_new:N \g__xtable_row_ht_seq
1269 \seq_new:N \g__xtable_row_dp_seq
1270 \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¹⁸，在渲染前需要刷新此变量的值。

```

\g__xtable_cell_ufill_prop 1271 \prop_new:N \g__xtable_cell_ht_prop
1272 \prop_new:N \g__xtable_cell_ufill_prop
1273 \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_merge_wd_seq` 合并单元格的净宽、净高、净深及上下填充值，在渲染前需要刷新此变量的值。

```
\g__xtable_merge_ht_seq 1274 \seq_new:N \g__xtable_merge_wd_seq
\g__xtable_merge_dp_seq 1275 \seq_new:N \g__xtable_merge_ht_seq
1276 \seq_new:N \g__xtable_merge_dp_seq
1277 \seq_new:N \g__xtable_merge_ufill_seq
1278 \seq_new:N \g__xtable_merge_dfill_seq
```

(End of definition for `\g__xtable_merge_wd_seq`, `\g__xtable_merge_ht_seq`, and `\g__xtable_merge_dp_seq`.)

`\g_xtable_merge_range_wd_seq` 合并单元格的排版尺寸，在渲染前需要刷新此变量的值。

```
\g_xtable_merge_range_ht_seq 1279 \seq_new:N \g__xtable_merge_range_wd_seq
\g_xtable_merge_range_dp_seq 1280 \seq_new:N \g__xtable_merge_range_ht_seq
1281 \seq_new:N \g__xtable_merge_range_dp_seq
```

(End of definition for `\g__xtable_merge_range_wd_seq`, `\g__xtable_merge_range_ht_seq`, and `\g__xtable_merge_range_dp_seq`.)

`\g__xtable_row_loc_seq` 用于存储表格的行列坐标（图形坐标）信息（包含 Header）。需要在渲染条件确认后
`\g__xtable_col_loc_seq` 再刷新。

```
1282 \seq_new:N \g__xtable_row_loc_seq
1283 \seq_new:N \g__xtable_col_loc_seq
```

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

6.2 行列尺寸

6.2.1 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 1284 \cs_new:Nn \__xtable_set_row_sep:nn
1285 {
1286   \seq_gclear:N \g__xtable_row_sep_seq
1287   \prg_replicate:nn {\g__xtable_row_count_int}
1288     { \seq_gput_right:Nn \g__xtable_row_sep_seq {#2} }
1289   \seq_gput_left:Nn \g__xtable_row_sep_seq {#1}
1290   \seq_gput_right:Nn \g__xtable_row_sep_seq {#1}
1291   \bool_if:NF \g__xtable_col_header_bool
1292   {
1293     \seq_gset_item:Nnn \g__xtable_row_sep_seq {1} { Opt }
1294     \seq_gset_item:Nnn \g__xtable_row_sep_seq {2} { #1 }
```

¹⁸此高度可能比行高要小，因为行高是该行中所有单元格高度中最大的值。

```

1295     }
1296   }
1297   \cs_new:Nn \__xtable_set_col_sep:nn
1298     {
1299     \seq_gclear:N \g__xtable_col_sep_seq
1300     \prg_replicate:nn {\g__xtable_col_count_int}
1301       { \seq_gput_right:Nn \g__xtable_col_sep_seq {#2} }
1302     \seq_gput_left:Nn \g__xtable_col_sep_seq {#1}
1303     \seq_gput_right:Nn \g__xtable_col_sep_seq {#1}
1304     \bool_if:NF \g__xtable_row_header_bool
1305       {
1306         \seq_gset_item:Nnn \g__xtable_col_sep_seq {1} { Opt }
1307         \seq_gset_item:Nnn \g__xtable_col_sep_seq {2} { #1 }
1308       }
1309     }
1310   \cs_generate_variant:Nn \__xtable_set_row_sep:nn { ne, en, ee }
1311   \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.)

```

6.2.2 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。顺便会更新首行的高度补偿¹⁹ 与未行的深度补偿²⁰，结果保存在 `\l__xtable_ufill_dim` `\l__xtable_dfill_dim` 中。

```

1312 \cs_new:Nn \__xtable_measure_cell_wd:n
1313   {
1314     \dim_zero:N \l__xtable_ufill_dim
1315     \dim_zero:N \l__xtable_dfill_dim
1316     \dim_set_eq:NN \l__xtable_wd_dim \g__xtable_cell_wd_min_dim
1317     \tl_if_empty:nF {#1}
1318       {
1319         \bool_set_true:N \l__xtable_tmpa_bool % top line mark
1320         \seq_set_split:Nnn \l__xtable_tmpa_seq {\} {#1}
1321         \seq_map_inline:Nn \l__xtable_tmpa_seq % each line
1322           {
1323             \hbox_set:Nn \l__xtable_tmpa_box {##1}
1324             \__xtable_dim_set_max:Nn \l__xtable_wd_dim
1325               { \box_wd:N \l__xtable_tmpa_box }
1326             \bool_if:NT \l__xtable_tmpa_bool

```

¹⁹高度与 `\c__xtable_std_ht_dim` 的差距值 (为正)，如果高度更大，则为零。

²⁰深度与 `\c__xtable_std_dp_dim` 的差距值 (为正)，如果深度更大，则为零。

```

1327         {
1328             \__xtable_dim_set_max:Nn \l__xtable_ufill_dim
1329             { \c__xtable_std_ht_dim - \box_ht:N \l__xtable_tmpa_box }
1330             \bool_set_false:N \l__xtable_tmpa_bool
1331         }
1332         \dim_zero:N \l__xtable_dfill_dim % clear above line
1333         \__xtable_dim_set_max:Nn \l__xtable_dfill_dim
1334         { \c__xtable_std_dp_dim - \box_dp:N \l__xtable_tmpa_box }
1335     }
1336 }
1337 }
1338 \cs_generate_variant:Nn \__xtable_measure_cell_wd:n { V, e }

```

(End of definition for __xtable_measure_cell_wd:n.)

__xtable_measure_cell_box_wd:n 测量单元格盒子 (#1) 的宽度，结果保存在 \l__xtable_wd_dim 中。同样会更新首行的高度补偿与未行的深度补偿，结果保存在 \l__xtable_ufill_dim \l__xtable_dfill_dim 中。

```

1339 \cs_new:Nn \__xtable_measure_cell_box_wd:n
1340 {
1341     \dim_zero:N \l__xtable_ufill_dim
1342     \dim_zero:N \l__xtable_dfill_dim
1343     \dim_set_eq:NN \l__xtable_wd_dim \g__xtable_cell_wd_min_dim
1344     \tl_if_empty:nF {#1}
1345     {
1346         \hbox_set:Nn \l__xtable_tmpa_box {#1}
1347         \__xtable_dim_set_max:Nn \l__xtable_wd_dim
1348         { \box_wd:N \l__xtable_tmpa_box }
1349         \__xtable_dim_set_max:Nn \l__xtable_ufill_dim
1350         { \c__xtable_std_ht_dim - \box_ht:N \l__xtable_tmpa_box }
1351         \__xtable_dim_set_max:Nn \l__xtable_dfill_dim
1352         { \c__xtable_std_dp_dim - \box_dp:N \l__xtable_tmpa_box }
1353     }
1354 }
1355 \cs_generate_variant:Nn \__xtable_measure_cell_box_wd:n { V, e }

```

(End of definition for __xtable_measure_cell_box_wd:n.)

__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。²¹

²¹本函数需要确认 \l__xtable_tmpa_dim 不会被调用的函数修改。

```

1356 \cs_new:Nn \__xtable_measure_col_wd:n
1357 {
1358   \dim_zero:N \l__xtable_tmpa_dim % max width
1359   \bool_if:NTF \g__xtable_col_header_bool
1360     { \int_set:Nn \l__xtable_tmpa_int {0} }
1361     { \int_set:Nn \l__xtable_tmpa_int {1} }
1362   \int_step_inline:nnn {\l__xtable_tmpa_int} {\g__xtable_row_count_int}
1363   {
1364     \__xtable_merge_if:nF {##1, #1}
1365     {
1366       \__xtable_parse_cell:n {##1, #1}
1367       \bool_if:NTF \l__xtable_is_box_bool
1368         { \__xtable_measure_cell_box_wd:V \l__xtable_data_tl }
1369         { \__xtable_measure_cell_wd:V \l__xtable_data_tl }
1370       \__xtable_dim_set_max:NN \l__xtable_tmpa_dim \l__xtable_wd_dim
1371       \prop_gput:Nne \g__xtable_cell_ufill_prop
1372         {##1,#1} {\dim_use:N \l__xtable_ufill_dim}
1373       \prop_gput:Nne \g__xtable_cell_dfill_prop
1374         {##1,#1} {\dim_use:N \l__xtable_dfill_dim}
1375     }
1376   }
1377   \dim_set_eq:NN \l__xtable_wd_dim \l__xtable_tmpa_dim
1378 }

```

(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。

```

1379 \cs_new:Nn \__xtable_calc_col_wd:
1380 {

```

初始化。

```

1381   \dim_zero:N \l__xtable_cachea_dim % 相同宽度的列的最大列宽
1382   \dim_zero:N \l__xtable_cacheb_dim % 相同宽度且填充的列的最大列宽
1383   \seq_clear:N \l__xtable_cachea_seq % 相同宽度的列的索引序列
1384   \seq_clear:N \l__xtable_cacheb_seq % 相同宽度且填充的列的索引序列
1385   \seq_gclear:N \g__xtable_col_wd_seq % 列宽序列
1386   \prop_gclear:N \g__xtable_cell_ufill_prop
1387   \prop_gclear:N \g__xtable_cell_dfill_prop

```

自然宽度循环，并统计相同宽度的列的索引及最大宽度。

```

1388   \__xtable_disable_sys_func:
1389   \bool_if:NTF \g__xtable_row_header_bool
1390     { \int_step_inline:nnn { 0 } { \g__xtable_col_count_int } }

```

```

1391     {
1392       \seq_gput_right:Nn \g__xtable_col_wd_seq { Opt }
1393       \int_step_inline:nnn { 1 } { \g__xtable_col_count_int }
1394     }
1395     { % \int_step_inline:nnn 的参数三
1396       \__xtable_measure_col_wd:n {##1}
1397       \__xtable_parse_col_wd_style:n {##1}
1398       \clist_if_in:nVTF {a,s,f,sf} \l__xtable_style_tl
1399       {
1400         \seq_gput_right:Ne \g__xtable_col_wd_seq
1401         { \dim_use:N \l__xtable_wd_dim }
1402         \tl_if_eq:VnT \l__xtable_style_tl {s}
1403         {
1404           \seq_put_right:Nn \l__xtable_cachea_seq {##1}
1405           \__xtable_dim_set_max:NN \l__xtable_cachea_dim \l__xtable_wd_dim
1406         }
1407         \tl_if_eq:VnT \l__xtable_style_tl {sf}
1408         {
1409           \seq_put_right:Nn \l__xtable_cacheb_seq {##1}
1410           \__xtable_dim_set_max:NN \l__xtable_cacheb_dim \l__xtable_wd_dim
1411         }
1412       }
1413       { \seq_gput_right:NV \g__xtable_col_wd_seq \l__xtable_style_tl }
1414     }
1415     \__xtable_restore_sys_func:

```

相同宽度循环：根据索引直接使用最大宽度替换。

```

1416     \seq_map_inline:Nn \l__xtable_cachea_seq
1417     {
1418       \seq_gset_item:Nne \g__xtable_col_wd_seq
1419       {##1+1} { \dim_use:N \l__xtable_cachea_dim }
1420     }
1421     \seq_map_inline:Nn \l__xtable_cacheb_seq
1422     {
1423       \seq_gset_item:Nne \g__xtable_col_wd_seq
1424       {##1+1} { \dim_use:N \l__xtable_cacheb_dim }
1425     }

```

填充循环。可填充长度 = 总宽 - 单元格间距 - 单元格所需宽度。

```

1426     \int_zero:N \l__xtable_cachea_int % 填充列的个数
1427     \seq_clear:N \l__xtable_cachea_seq % 填充列的索引序列
1428     \dim_set:Nn \l__xtable_wd_dim
1429     { \seq_item:Nn \g__xtable_col_wd_style_seq {-1} }

```

```

1430 \seq_map_inline:Nn \g__xtable_col_sep_seq
1431   { \dim_sub:Nn \l__xtable_wd_dim {##1} }
1432 \int_step_inline:nnn { 0 } { \g__xtable_col_count_int }
1433   {
1434     \dim_sub:Nn \l__xtable_wd_dim
1435       { \seq_item:Nn \g__xtable_col_wd_seq {##1+1} }
1436     \__xtable_parse_col_wd_style:n {##1}
1437     \tl_if_in:NnT \l__xtable_style_tl {f}
1438     {
1439       \int_incr:N \l__xtable_cachea_int
1440       \seq_put_right:Nn \l__xtable_cachea_seq {##1}
1441     }
1442   }
1443 \int_if_zero:nF { \l__xtable_cachea_int }
1444   {
1445     \dim_set:Nn \l__xtable_cachea_dim
1446       { \l__xtable_wd_dim / \l__xtable_cachea_int }
1447   }
1448 \seq_map_inline:Nn \l__xtable_cachea_seq
1449   {
1450     \dim_sub:Nn \l__xtable_wd_dim
1451       { \seq_item:Nn \g__xtable_col_wd_seq {##1+1} }
1452     \dim_add:Nn \l__xtable_wd_dim { \l__xtable_cachea_dim }
1453     \seq_gset_item:Nne \g__xtable_col_wd_seq
1454       {##1+1} { \dim_use:N \l__xtable_wd_dim }
1455   }
1456 }

```

(End of definition for __xtable_calc_col_wd:.)

6.2.3 高度计算

__xtable_measure_cell_ht:nnn 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。

```

1457 \cs_new:Nn \__xtable_measure_cell_ht:nnn
1458   {
1459     \dim_set_eq:NN \l__xtable_ht_dim \c__xtable_std_ht_dim
1460     \dim_set_eq:NN \l__xtable_dp_dim \c__xtable_std_dp_dim
1461     \tl_if_empty:nF {#3}
1462     {
1463       \str_if_eq:nnTF {#1} {t}
1464       { \vbox_set_top:Nn } { \vbox_set:Nn }

```

```

1465     \l__xtable_tmpa_box
1466     {
1467         \baselineskip=\g__xtable_cell_lineskip_dim
1468         \hsize=#2 \parindent=0pt \noindent #3
1469     }
1470     \__xtable_dim_set_max:Nn \l__xtable_ht_dim
1471     { \box_ht:N \l__xtable_tmpa_box + \l__xtable_ufill_dim }
1472     \__xtable_dim_set_max:Nn \l__xtable_dp_dim
1473     { \box_dp:N \l__xtable_tmpa_box + \l__xtable_dfill_dim }
1474     }
1475 }
1476 \cs_generate_variant:Nn \__xtable_measure_cell_ht:nnn { VeV }

```

(End of definition for __xtable_measure_cell_ht:nnn.)

__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。²²

```

1477 \cs_new:Nn \__xtable_measure_row_ht:n
1478 {
1479     \dim_zero:N \l__xtable_tmpa_dim
1480     \dim_zero:N \l__xtable_tmpb_dim
1481     \__xtable_parse_row_align:n {#1}
1482     \bool_if:NTF \g__xtable_row_header_bool
1483     { \int_set:Nn \l__xtable_tmpa_int {0} }
1484     { \int_set:Nn \l__xtable_tmpa_int {1} }
1485     \int_step_inline:nnn { \l__xtable_tmpa_int } { \g__xtable_col_count_int }
1486     {
1487         \__xtable_merge_if:nF {#1, ##1}
1488         {
1489             \__xtable_parse_cell:n {#1, ##1}
1490             \__xtable_parse_cell_fill:n {#1, ##1}
1491             \__xtable_measure_cell_ht:VeV \l__xtable_row_align_tl
1492             { \seq_item:Nn \g__xtable_col_wd_seq { ##1 + 1 } }
1493             \l__xtable_data_tl
1494             \prop_gput:Nne \g__xtable_cell_ht_prop {#1,##1}
1495             { \dim_use:N \l__xtable_ht_dim }
1496             \__xtable_dim_set_max:NN \l__xtable_tmpa_dim \l__xtable_ht_dim
1497             \__xtable_dim_set_max:NN \l__xtable_tmpb_dim \l__xtable_dp_dim
1498         }
1499     }

```

²²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。

```

1500     \dim_set_eq:NN \l__xtable_ht_dim \l__xtable_tmpa_dim
1501     \dim_set_eq:NN \l__xtable_dp_dim \l__xtable_tmpb_dim
1502   }

```

(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。

```

1503 \cs_new:Nn \__xtable_calc_row_ht:
1504 {
1505     \dim_zero:N     \l__xtable_cachea_dim
1506     \dim_zero:N     \l__xtable_cacheb_dim
1507     \seq_clear:N    \l__xtable_cachea_seq
1508     \seq_gclear:N   \g__xtable_row_ht_seq
1509     \seq_gclear:N   \g__xtable_row_dp_seq
1510     \prop_gclear:N \g__xtable_cell_ht_prop

```

自然高度循环，并统计相同高度的列的索引及最大高度。

```

1511     \__xtable_disable_sys_func:
1512     \bool_if:NTF \g__xtable_col_header_bool
1513       { \int_step_inline:nnn { 0 } { \g__xtable_row_count_int } }
1514       {
1515         \seq_gput_right:Nn \g__xtable_row_ht_seq { Opt }
1516         \seq_gput_right:Nn \g__xtable_row_dp_seq { Opt }
1517         \int_step_inline:nnn { 1 } { \g__xtable_row_count_int }
1518       }
1519     { % \int_step_inline:nnn 的参数三
1520       \__xtable_measure_row_ht:n {##1}
1521       \__xtable_parse_row_ht_style:n {##1}
1522       \clist_if_in:nVTF {a,s} \l__xtable_style_tl
1523       {
1524         \seq_gput_right:Ne \g__xtable_row_ht_seq
1525           { \dim_use:N \l__xtable_ht_dim }
1526         \seq_gput_right:Ne \g__xtable_row_dp_seq
1527           { \dim_use:N \l__xtable_dp_dim }
1528         \tl_if_eq:VnT \l__xtable_style_tl {s}
1529         {
1530           \seq_put_right:Nn \l__xtable_cachea_seq {##1}
1531           \__xtable_dim_set_max:NN \l__xtable_cachea_dim \l__xtable_ht_dim
1532           \__xtable_dim_set_max:NN \l__xtable_cacheb_dim \l__xtable_dp_dim
1533         }
1534       }
1535     {
1536       \seq_gput_right:Ne \g__xtable_row_ht_seq

```

```

1537         { \dim_eval:n { \l__xtable_style_tl - \l__xtable_dp_dim } }
1538     \seq_gput_right:Ne \g__xtable_row_dp_seq
1539         { \dim_use:N \l__xtable_dp_dim }
1540     }
1541 }
1542 \__xtable_restore_sys_func:

```

相同高度循环：根据索引直接使用最大高度替换。

```

1543 \seq_map_inline:Nn \l__xtable_cachea_seq
1544 {
1545     \__xtable_parse_row_align:n {##1}
1546     \str_if_eq:VnTF \l__xtable_row_align_tl {t}
1547     {
1548         \dim_set:Nn \l__xtable_ht_dim
1549             { \seq_item:Nn \g__xtable_row_ht_seq {##1+1} }
1550         \dim_set:Nn \l__xtable_dp_dim
1551             { \l__xtable_cachea_dim + \l__xtable_cacheb_dim - \l__xtable_ht_dim }
1552         \seq_gset_item:Nne \g__xtable_row_dp_seq
1553             {##1+1} { \dim_use:N \l__xtable_dp_dim }
1554     }
1555     {
1556         \dim_set:Nn \l__xtable_dp_dim
1557             { \seq_item:Nn \g__xtable_row_dp_seq {##1+1} }
1558         \dim_set:Nn \l__xtable_ht_dim
1559             { \l__xtable_cachea_dim + \l__xtable_cacheb_dim - \l__xtable_dp_dim }
1560         \seq_gset_item:Nne \g__xtable_row_ht_seq
1561             {##1+1} { \dim_use:N \l__xtable_ht_dim }
1562     }
1563 }
1564 }

```

(End of definition for __xtable_calc_row_ht:.)

6.2.4 查询尺寸

__xtable_parse_cell_size:n 查询单元格的排版尺寸。#1 为数字坐标。结果保存在 \l__xtable_wd_dim、\l__xtable_ht_dim 及 \l__xtable_dp_dim 中。

```

1565 \cs_new:Nn \__xtable_parse_cell_size:n
1566 {
1567     \int_set:Nn \l__xtable_tmpa_int { \clist_item:nn {#1} {1} + 1 }
1568     \int_set:Nn \l__xtable_tmpb_int { \clist_item:nn {#1} {2} + 1 }
1569     \dim_set:Nn \l__xtable_wd_dim
1570         { \seq_item:Nn \g__xtable_col_wd_seq { \l__xtable_tmpb_int } }

```

```

1571 \dim_set:Nn \l__xtable_ht_dim
1572 { \seq_item:Nn \g__xtable_row_ht_seq { \l__xtable_tmpa_int } }
1573 \dim_set:Nn \l__xtable_dp_dim
1574 { \seq_item:Nn \g__xtable_row_dp_seq { \l__xtable_tmpa_int } }
1575 }

```

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

`__xtable_parse_cell_fill:n` 查询单元格的首行高度与深度填充值。#1 为数字坐标。结果保存在 `\l__xtable_ufill_dim` 与 `\l__xtable_dfill_dim` 中。

```

1576 \cs_new:Nn \__xtable_parse_cell_fill:n
1577 {
1578 \prop_get:NnNF \g__xtable_cell_ufill_prop {#1} \l__xtable_tmpa_tl
1579 { \tl_set:Nn \l__xtable_tmpa_tl {Opt} }
1580 \prop_get:NnNF \g__xtable_cell_dfill_prop {#1} \l__xtable_tmpb_tl
1581 { \tl_set:Nn \l__xtable_tmpb_tl {Opt} }
1582 \dim_set:Nn \l__xtable_ufill_dim {\l__xtable_tmpa_tl}
1583 \dim_set:Nn \l__xtable_dfill_dim {\l__xtable_tmpb_tl}
1584 }

```

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

`__xtable_parse_cell_ht:n` 查询单元格的实际高度。#1 为数字坐标。结果保存在 `\l__xtable_ht_dim` 中。

```

1585 \cs_new:Nn \__xtable_parse_cell_ht:n
1586 {
1587 \prop_get:NnN \g__xtable_cell_ht_prop {#1} \l__xtable_tmpa_tl
1588 \dim_set:Nn \l__xtable_ht_dim {\l__xtable_tmpa_tl}
1589 }

```

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

6.3 合并单元格

6.3.1 尺寸计算

`__xtable_measure_merge_size:` 查询单元格的实际高度。使用单元格格内容及高宽深及填充值等变量。

```

1590 \cs_new:Nn \__xtable_measure_merge_size:
1591 {
1592 \seq_gclear:N \g__xtable_merge_wd_seq
1593 \seq_gclear:N \g__xtable_merge_ht_seq
1594 \seq_gclear:N \g__xtable_merge_dp_seq
1595 \seq_gclear:N \g__xtable_merge_dfill_seq
1596 \seq_gclear:N \g__xtable_merge_ufill_seq
1597 \int_step_inline:nn

```

```

1598     { \_xtable_parse_merge_count: }
1599     {
1600     \_xtable_parse_merge:n {##1}
1601     % 计算宽度
1602     \bool_if:NTF \l_xtable_is_box_bool
1603     { \_xtable_measure_cell_box_wd:V \l_xtable_data_tl }
1604     { \_xtable_measure_cell_wd:V \l_xtable_data_tl }
1605     \seq_gput_right:Ne \g_xtable_merge_wd_seq
1606     { \dim_use:N \l_xtable_wd_dim }
1607     \seq_gput_right:Ne \g_xtable_merge_dfill_seq
1608     { \dim_use:N \l_xtable_dfill_dim }
1609     \seq_gput_right:Ne \g_xtable_merge_ufill_seq
1610     { \dim_use:N \l_xtable_ufill_dim }
1611     % 计算高度
1612     \_xtable_parse_merge_align:n {##1}
1613     \_xtable_measure_cell_ht:VeV \l_xtable_row_align_tl
1614     { \dim_use:N \l_xtable_wd_dim }
1615     \l_xtable_data_tl
1616     \seq_gput_right:Ne \g_xtable_merge_ht_seq
1617     { \dim_use:N \l_xtable_ht_dim }
1618     \seq_gput_right:Ne \g_xtable_merge_dp_seq
1619     { \dim_use:N \l_xtable_dp_dim }
1620     }
1621   }

```

(End of definition for _xtable_measure_merge_size:.)

_xtable_calc_merge_size: 计算单元格的排版尺寸。使用单元格内容及高宽深及填充值等变量。

```

1622 \cs_new:Nn \_xtable_calc_merge_size:
1623 {
1624   \seq_gclear:N \g_xtable_merge_range_wd_seq
1625   \seq_gclear:N \g_xtable_merge_range_ht_seq
1626   \seq_gclear:N \g_xtable_merge_range_dp_seq
1627   \_xtable_disable_sys_func:
1628   \_xtable_measure_merge_size:
1629   \_xtable_restore_sys_func:
1630   \int_step_inline:nn
1631     { \_xtable_parse_merge_count: }
1632     {
1633     \_xtable_parse_merge:n {##1}
1634     \_xtable_parse_merge_align:n {##1}

```

计算宽度。

```

1635     \int_zero:N \l__xtable_tmpa_int
1636     \int_step_inline:nnn
1637     { \seq_item:Nn \l__xtable_merge_seq {2} }
1638     { \seq_item:Nn \l__xtable_merge_seq {4} }
1639     {
1640         \dim_add:Nn \l__xtable_wd_dim
1641         { \seq_item:Nn \g__xtable_col_sep_seq { #####1 + 1 } }
1642         \int_if_zero:nT { \l__xtable_tmpa_int }
1643         { % 置后清零, 以排除第一列左边的间隙
1644             \dim_zero:N \l__xtable_wd_dim
1645             \int_incr:N \l__xtable_tmpa_int
1646         }
1647         \dim_add:Nn \l__xtable_wd_dim
1648         { \seq_item:Nn \g__xtable_col_wd_seq { #####1 + 1 } }
1649     }
1650     \seq_gput_right:Ne \g__xtable_merge_range_wd_seq
1651     { \dim_use:N \l__xtable_wd_dim }

```

计算高度与深度。

```

1652     \int_zero:N \l__xtable_tmpa_int
1653     \int_step_inline:nnn
1654     { \seq_item:Nn \l__xtable_merge_seq {1} }
1655     { \seq_item:Nn \l__xtable_merge_seq {3} }
1656     {
1657         \dim_add:Nn \l__xtable_tmpa_dim
1658         { \seq_item:Nn \g__xtable_row_sep_seq { #####1 + 1 } }
1659         \int_if_zero:nT { \l__xtable_tmpa_int }
1660         { % 置后清零, 以排除第一行上边的间隙
1661             \dim_zero:N \l__xtable_tmpa_dim
1662             \int_incr:N \l__xtable_tmpa_int
1663             \dim_set:Nn \l__xtable_ht_dim % 首行
1664             { \seq_item:Nn \g__xtable_row_ht_seq { #####1 + 1 } }
1665         }
1666         \dim_set:Nn \l__xtable_dp_dim % 未列
1667         { \seq_item:Nn \g__xtable_row_dp_seq { #####1 + 1 } }
1668         \dim_add:Nn \l__xtable_tmpa_dim
1669         { \seq_item:Nn \g__xtable_row_ht_seq { #####1 + 1 } }
1670         \dim_add:Nn \l__xtable_tmpa_dim
1671         { \seq_item:Nn \g__xtable_row_dp_seq { #####1 + 1 } }
1672     }
1673     \str_if_eq:VnTF \l__xtable_row_align_tl {t}
1674     {
1675         \seq_gput_right:Ne \g__xtable_merge_range_ht_seq

```

```

1676         { \dim_use:N \l__xtable_ht_dim }
1677         \seq_gput_right:Ne \g__xtable_merge_range_dp_seq
1678         { \dim_eval:n { \l__xtable_tmpa_dim - \l__xtable_ht_dim } }
1679     } {
1680         \seq_gput_right:Ne \g__xtable_merge_range_dp_seq
1681         { \dim_use:N \l__xtable_dp_dim }
1682         \seq_gput_right:Ne \g__xtable_merge_range_ht_seq
1683         { \dim_eval:n { \l__xtable_tmpa_dim - \l__xtable_dp_dim } }
1684     }
1685 }
1686 }

```

(End of definition for `__xtable_calc_merge_size:`)

6.3.2 尺寸查询

`__xtable_parse_merge_size:n` 查询合并单元格的排版尺寸，#1 为索引。结果保存在 `\l__xtable_wd_dim`、`\l__xtable_ht_dim` 与 `\l__xtable_dp_dim` 中。

```

1687 \cs_new:Nn \__xtable_parse_merge_size:n
1688 {
1689     \dim_set:Nn \l__xtable_wd_dim
1690     { \seq_item:Nn \g__xtable_merge_range_wd_seq {#1} }
1691     \dim_set:Nn \l__xtable_ht_dim
1692     { \seq_item:Nn \g__xtable_merge_range_ht_seq {#1} }
1693     \dim_set:Nn \l__xtable_dp_dim
1694     { \seq_item:Nn \g__xtable_merge_range_dp_seq {#1} }
1695 }

```

(End of definition for `__xtable_parse_merge_size:n`)

`__xtable_parse_merge_fill:n` 查询单元格的首行高度与深度填充值，#1 为索引。结果保存在 `\l__xtable_ufill_dim` 与 `\l__xtable_dfill_dim` 中。

```

1696 \cs_new:Nn \__xtable_parse_merge_fill:n
1697 {
1698     \dim_set:Nn \l__xtable_dfill_dim
1699     { \seq_item:Nn \g__xtable_merge_dfill_seq {#1} }
1700     \dim_set:Nn \l__xtable_ufill_dim
1701     { \seq_item:Nn \g__xtable_merge_ufill_seq {#1} }
1702 }

```

(End of definition for `__xtable_parse_merge_fill:n`)

`__xtable_parse_merge_ht:n` 查询单元格的首行高度与深度填充值，#1 为索引。结果保存在 `\l__xtable_ht_dim` 中。

```

1703 \cs_new:Nn \__xtable_parse_merge_ht:n
1704 {
1705   \dim_set:Nn \l__xtable_ht_dim
1706     { \seq_item:Nn \g__xtable_merge_ht_seq {#1} }
1707 }

```

(End of definition for __xtable_parse_merge_ht:n.)

6.4 计算坐标

__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。

```

1708 \cs_new:Nn \__xtable_calc_coord:
1709 {
1710   \dim_zero:N \l__xtable_wd_dim
1711   \seq_gclear:N \g__xtable_col_loc_seq
1712   \seq_gput_right:NV \g__xtable_col_loc_seq \l__xtable_wd_dim
1713   \bool_if:NTF \g__xtable_row_header_bool
1714     { \int_set:Nn \l__xtable_tmpa_int {0} }
1715     {
1716       \int_set:Nn \l__xtable_tmpa_int {1}
1717       \seq_gput_right:NV \g__xtable_col_loc_seq \l__xtable_wd_dim
1718     }
1719   \dim_add:Nn \l__xtable_wd_dim
1720     { ( \seq_item:Nn \g__xtable_col_sep_seq { \l__xtable_tmpa_int + 1 } ) / 2 }
1721   \int_step_inline:nnn
1722     { \l__xtable_tmpa_int } { \g__xtable_col_count_int }
1723     {
1724       \dim_add:Nn \l__xtable_wd_dim
1725         { ( \seq_item:Nn \g__xtable_col_sep_seq { ##1 + 1 } ) / 2 }
1726       \dim_add:Nn \l__xtable_wd_dim
1727         { \seq_item:Nn \g__xtable_col_wd_seq { ##1 + 1 } }
1728       \dim_add:Nn \l__xtable_wd_dim
1729         { ( \seq_item:Nn \g__xtable_col_sep_seq { ##1 + 2 } ) / 2 }
1730       \seq_gput_right:NV \g__xtable_col_loc_seq \l__xtable_wd_dim
1731     }
1732   \dim_add:Nn \l__xtable_wd_dim
1733     { ( \seq_item:Nn \g__xtable_col_sep_seq {-1} ) / 2 }
1734   \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。

```

1735   \dim_zero:N \l__xtable_ht_dim
1736   \seq_gclear:N \g__xtable_row_loc_seq

```

```

1737 \seq_gput_right:NV \g__xtable_row_loc_seq \l__xtable_ht_dim
1738 \bool_if:NTF \g__xtable_col_header_bool
1739 { \int_set:Nn \l__xtable_tmpa_int {0} }
1740 {
1741 \int_set:Nn \l__xtable_tmpa_int {1}
1742 \seq_gput_right:NV \g__xtable_row_loc_seq \l__xtable_ht_dim
1743 }
1744 \dim_sub:Nn \l__xtable_ht_dim
1745 { ( \seq_item:Nn \g__xtable_row_sep_seq { \l__xtable_tmpa_int + 1 } ) / 2 }
1746 \int_step_inline:nnn
1747 { \l__xtable_tmpa_int } { \g__xtable_row_count_int }
1748 {
1749 \dim_sub:Nn \l__xtable_ht_dim
1750 { ( \seq_item:Nn \g__xtable_row_sep_seq { ##1+1 } ) / 2 }
1751 \dim_sub:Nn \l__xtable_ht_dim
1752 { \seq_item:Nn \g__xtable_row_ht_seq { ##1+1 } }
1753 \dim_sub:Nn \l__xtable_ht_dim
1754 { \seq_item:Nn \g__xtable_row_dp_seq { ##1+1 } }
1755 \dim_sub:Nn \l__xtable_ht_dim
1756 { ( \seq_item:Nn \g__xtable_row_sep_seq { ##1+2 } ) / 2 }
1757 \seq_gput_right:NV \g__xtable_row_loc_seq \l__xtable_ht_dim
1758 }
1759 \dim_sub:Nn \l__xtable_ht_dim
1760 { ( \seq_item:Nn \g__xtable_row_sep_seq {-1} ) / 2 }
1761 \seq_gset_item:NnV \g__xtable_row_loc_seq {-1} \l__xtable_ht_dim
1762 }

```

(End of definition for __xtable_calc_coord:.)

7 渲染输出

7.1 通用内容

7.1.1 局部变量

\l__xtable_x_dim 用于存储绘图坐标信息。

```

\l__xtable_y_dim 1763 \dim_new:N \l__xtable_x_dim
1764 \dim_new:N \l__xtable_y_dim

```

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

\l__xtable_offset_x_dim 两个偏移值变量。

```

\l__xtable_offset_y_dim 1765 \dim_new:N \l__xtable_offset_x_dim
1766 \dim_new:N \l__xtable_offset_y_dim

```

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

`\l__xtable_expand_start_dim` 两个扩展值变量。

```
\l__xtable_expand_end_dim 1767 \dim_new:N \l__xtable_expand_start_dim
1768 \dim_new:N \l__xtable_expand_end_dim
```

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

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

```
\g__xtable_col_rule_seq 1769 \seq_new:N \g__xtable_row_rule_seq
1770 \seq_new:N \g__xtable_col_rule_seq
```

(End of definition for `\g__xtable_row_rule_seq` and `\g__xtable_col_rule_seq`.)

`\l__xtable_style_seq` 用于存储样式序列。

```
1771 \seq_new:N \l__xtable_style_seq
```

(End of definition for `\l__xtable_style_seq`.)

`\l__xtable_cell_box` 用于存储单元格内容的盒子。

```
1772 \box_new:N \l__xtable_cell_box
```

(End of definition for `\l__xtable_cell_box`.)

7.1.2 渲染函数

`__xtable_draw_hrule: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。

```
1773 \cs_new:Nn \__xtable_draw_hrule:n
1774 {
1775   \draw_path_moveto:n
1776     {
1777       \seq_item:Nn \g__xtable_col_loc_seq {1} - \l__xtable_expand_start_dim,
1778       \seq_item:Nn \g__xtable_row_loc_seq {#1}
1779     }
1780   \draw_path_lineto:n
1781     {
1782       \seq_item:Nn \g__xtable_col_loc_seq {-1} + \l__xtable_expand_end_dim,
1783       \seq_item:Nn \g__xtable_row_loc_seq {#1}
1784     }
1785 }
```

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

`_xtable_draw_vrule: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。

```

1786 \cs_new:Nn \_xtable_draw_vrule:n
1787 {
1788   \draw_path_moveto:n
1789   {
1790     \seq_item:Nn \g__xtable_col_loc_seq {#1},
1791     \seq_item:Nn \g__xtable_row_loc_seq {1} + \l__xtable_expand_start_dim
1792   }
1793   \draw_path_lineto:n
1794   {
1795     \seq_item:Nn \g__xtable_col_loc_seq {#1},
1796     \seq_item:Nn \g__xtable_row_loc_seq {-1} - \l__xtable_expand_end_dim
1797   }
1798 }

```

(End of definition for _xtable_draw_vrule: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。

```

1799 \cs_new:Nn \_xtable_print_data:nnn
1800 {
1801   \str_if_in:nnTF {tmb} {#1}
1802   { \tl_set:Nn \l__xtable_tmpb_tl {#1} }
1803   { \tl_set:Nn \l__xtable_tmpb_tl {b} }
1804   \str_case:nnF {#2}
1805   {
1806     {l} { \tl_set:Nn \l__xtable_tmpa_tl {\raggedright} }
1807     {c} { \tl_set:Nn \l__xtable_tmpa_tl {\centering} }
1808     {r} { \tl_set:Nn \l__xtable_tmpa_tl {\raggedleft} }
1809   }
1810   { \tl_set:Nn \l__xtable_tmpa_tl {} }
1811   \tl_if_empty:nT {#3}
1812   { \tl_put_right:Nn \l__xtable_tmpa_tl {\rule{1pt}{0pt}} }
1813   \str_case:Vn \l__xtable_tmpb_tl
1814   {
1815     {t}
1816     {
1817       \vbox_set_top:Nn \l__xtable_tmpa_box
1818       {

```

```

1819         \baselineskip=\g__xtable_cell_lineskip_dim
1820         \hsize=\l__xtable_wd_dim \parindent=0pt
1821         \l__xtable_tmpa_tl #3 \vfill \par
1822     }
1823 }

```

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。

```

1824     {m}
1825     {
1826         \dim_set:Nn \l__xtable_tmpa_dim
1827         { (\l__xtable_ht_dim - \l__xtable_cachea_dim)/2 + \l__xtable_ufill_dim }
1828         \vbox_set_to_ht:Nnn \l__xtable_tmpa_box {\l__xtable_ht_dim}
1829         {
1830             \baselineskip=\g__xtable_cell_lineskip_dim
1831             \hsize=\l__xtable_wd_dim \parindent=0pt
1832             \vskip \l__xtable_tmpa_dim
1833             \l__xtable_tmpa_tl #3 \vfill \par
1834         }
1835     }
1836     {b}
1837     {
1838         \dim_set:Nn \l__xtable_tmpa_dim
1839         { \l__xtable_ht_dim - \l__xtable_cachea_dim + \l__xtable_ufill_dim }
1840         \vbox_set:Nn \l__xtable_tmpa_box
1841         {
1842             \baselineskip=\g__xtable_cell_lineskip_dim
1843             \hsize=\l__xtable_wd_dim \parindent=0pt
1844             \vskip \l__xtable_tmpa_dim
1845             \l__xtable_tmpa_tl #3 \par
1846         }
1847     }
1848 }
1849 \dim_compare:nNnT {\box_dp:N \l__xtable_tmpa_box} < {\l__xtable_dp_dim}
1850   { \box_set_dp:Nn \l__xtable_tmpa_box {\l__xtable_dp_dim} }
1851 \box_use:N \l__xtable_tmpa_box
1852 }
1853 \cs_generate_variant:Nn \__xtable_print_data:nnn {VVV}

```

(End of definition for __xtable_print_data:nnn.)

`__xtable_print_merge_data:nn` 打印合并单元格，修改变量 `\l__xtable_merge_seq`。

```

1854 \cs_new:Nn \__xtable_print_merge_data:nn

```

```

1855 {
1856   \__xtable_parse_merge_id:n {#1, #2}
1857   \__xtable_parse_merge:V \l__xtable_merge_id_tl
1858   \tl_set:Ne \l__xtable_tmpa_tl
1859   { \seq_item:Nn \l__xtable_merge_seq {2} }
1860   \str_if_eq:VnT \l__xtable_tmpa_tl {#2}
1861   { % 只有合并单元格的第一列输出
1862     \__xtable_parse_merge_size:n { \l__xtable_merge_id_tl }
1863     \__xtable_parse_merge_fill:n { \l__xtable_merge_id_tl }
1864     \tl_set:Ne \l__xtable_tmpa_tl
1865     { \seq_item:Nn \l__xtable_merge_seq {1} }
1866     \str_if_eq:VnTF \l__xtable_tmpa_tl {#1}
1867     { % 如果是首行, 则完整输出
1868       \vbox_set_top:Nn \l__xtable_tmpa_box
1869       {
1870         \baselineskip=\g__xtable_cell_lineskip_dim
1871         \hsize=\l__xtable_wd_dim \parindent=0pt
1872         \centering \l__xtable_data_tl \vfill \par
1873       }
1874       \__xtable_parse_cell_size:n {#1, #2}
1875       \box_set_ht:Nn \l__xtable_tmpa_box { \l__xtable_ht_dim - \l__xtable_dfill_dim }
1876       \box_set_dp:Nn \l__xtable_tmpa_box { \l__xtable_dp_dim }
1877       \box_use:N \l__xtable_tmpa_box
1878     }
1879     { % 如果是非首行, 则只输出占位符
1880       \exp_args:Ne \rule { \dim_use:N \l__xtable_wd_dim } { Opt }
1881     }
1882     \hspace{\g__xtable_cell_sep_dim}
1883   }
1884 }

```

(End of definition for __xtable_print_merge_data:nn.)

`\printtable` 打印表格。

```

1885 \NewDocumentCommand \printtable { s 0{0em} }
1886 {
1887   \__xtable_set_row_sep:ne {Opt} { \dim_use:N \g__xtable_cell_sep_dim }
1888   \__xtable_set_col_sep:ne {Opt} { \dim_use:N \g__xtable_cell_sep_dim }
1889   \__xtable_calc_col_wd: \__xtable_calc_row_ht:
1890   \__xtable_calc_merge_size:
1891   \bool_if:NTF \g__xtable_col_header_bool
1892   { \int_set:Nn \l__xtable_cachea_int {0} }
1893   { \int_set:Nn \l__xtable_cachea_int {1} }

```

```

1894 \bool_if:NTF \g__xtable_row_header_bool
1895   { \int_set:Nn \l__xtable_cacheb_int {0} }
1896   { \int_set:Nn \l__xtable_cacheb_int {1} }
1897 \int_step_inline:nnn { \l__xtable_cachea_int } { \g__xtable_row_count_int }
1898 {
1899   \__xtable_support_sys_func:
1900   \hbox:n
1901     {
1902       \dim_zero:N \l__xtable_tmpa_dim
1903       \hspace{#2}
1904       \int_step_inline:nnn { \l__xtable_cacheb_int } { \g__xtable_col_count_int }
1905       {
1906         \__xtable_merge_if:nTF {##1, ####1}
1907         { \__xtable_print_merge_data:nn {##1} {####1} }
1908         {
1909           \__xtable_parse_cell:n {##1, ####1}
1910           \__xtable_parse_cell_fill:n {##1, ####1}
1911           \__xtable_parse_cell_ht:n {##1, ####1}
1912           \dim_set_eq:NN \l__xtable_cachea_dim \l__xtable_ht_dim
1913           \__xtable_parse_cell_size:n {##1, ####1}
1914           \__xtable_parse_row_align:n {##1}
1915           \__xtable_parse_col_align:n {####1}
1916           \__xtable_print_data:VVV
1917           \l__xtable_row_align_tl \l__xtable_col_align_tl \l__xtable_data_tl
1918           \hspace{\g__xtable_cell_sep_dim}
1919         }
1920       }
1921     }
1922   \IfBooleanF {#1}
1923     { \int_compare:nNnT {##1} < { \g__xtable_row_count_int } { \ } }
1924   \__xtable_restore_sys_func:
1925 }
1926 }

```

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

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。使用变量 \l__xtable_wd_dim，渲染好的盒子存储在 \l__xtable_cell_box 中。

```

1927 \cs_new:Nn \__xtable_render_data:nnn

```

```

1928 {
1929   \str_case:nnF{#2}
1930   {
1931     {l} { \tl_set:Nn \l__xtable_tmpa_tl { \raggedright } }
1932     {c} { \tl_set:Nn \l__xtable_tmpa_tl { \centering } }
1933     {r} { \tl_set:Nn \l__xtable_tmpa_tl { \raggedleft } }
1934   }
1935   { \tl_set:Nn \l__xtable_tmpa_tl {} }
1936   \tl_if_empty:nTF {#3}
1937   { \tl_set:Nn \l__xtable_tmpb_tl { \rule{1pt}{0pt} } }
1938   { \tl_set:Nn \l__xtable_tmpb_tl {#3} }
1939   \str_if_eq:nnTF {#1} {t}
1940   { \vbox_set_top:Nn } { \vbox_set:Nn }
1941   \l__xtable_cell_box
1942   {
1943     \baselineskip=3ex \hspace=\l__xtable_wd_dim \parindent=0pt
1944     \l__xtable_tmpa_tl \l__xtable_tmpb_tl \par
1945   }
1946 }
1947 \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；如果指定单元格已合并，则绘制空内容。使用 \l__xtable_x_dim 与 \l__xtable_y_dim 的坐标信息。

```

1948 \cs_new:Nn \__xtable_draw_cell:nn
1949 {
1950   \__xtable_parse_cell_fill:n {#1, #2}
1951   \__xtable_parse_cell_size:n {#1, #2}
1952   \__xtable_parse_row_align:n {#1}
1953   \__xtable_parse_col_align:n {#2}
1954   \__xtable_merge_if:nTF {#1, #2}
1955   {
1956     \bool_set_false:N \l__xtable_is_box_bool
1957     \tl_set:Nn \l__xtable_data_tl {}
1958   }
1959   { \__xtable_parse_cell:n {#1, #2} }
1960   \__xtable_render_data:VVV % 获取包含渲染内容的盒子
1961   \l__xtable_row_align_tl \l__xtable_col_align_tl \l__xtable_data_tl
1962   \tl_if_eq:NnTF \l__xtable_row_align_tl {m}
1963   {
1964     \dim_set:Nn \l__xtable_tmpb_dim
1965     { \box_ht:N \l__xtable_cell_box + \l__xtable_ufill_dim }

```

```

1966     \dim_set:Nn \l__xtable_tmpa_dim
1967     { \l__xtable_y_dim - \l__xtable_tmpb_dim }
1968     \dim_sub:Nn \l__xtable_tmpa_dim
1969     { (\l__xtable_ht_dim - \l__xtable_tmpb_dim) / 2 }
1970   }
1971   {
1972     \dim_set:Nn \l__xtable_tmpa_dim
1973     { \l__xtable_y_dim - \l__xtable_ht_dim }
1974   }
1975   \draw_box_use:Nn \l__xtable_cell_box
1976   { \l__xtable_x_dim, \l__xtable_tmpa_dim }
1977 }

```

(End of definition for __xtable_draw_cell:nn.)

__xtable_draw_merge:n 绘制指定单元格，#1#2 为数字坐标；如果指定单元格已合并，则绘制空内容。使用 \l__xtable_x_dim 与 \l__xtable_y_dim 的坐标信息。

```

1978 \cs_new:Nn \__xtable_draw_merge:n
1979 {
1980   \__xtable_parse_merge:n {#1}
1981   \int_set:Nn \l__xtable_tmpa_int
1982   { \seq_item:Nn \l__xtable_merge_seq {1} }
1983   \int_set:Nn \l__xtable_tmpb_int
1984   { \seq_item:Nn \l__xtable_merge_seq {2} }
1985   \dim_set:Nn \l__xtable_y_dim
1986   { \seq_item:Nn \g__xtable_row_loc_seq { \l__xtable_tmpa_int + 1 } }
1987   \dim_set:Nn \l__xtable_x_dim
1988   { \seq_item:Nn \g__xtable_col_loc_seq { \l__xtable_tmpb_int + 1 } }
1989   \dim_add:Nn \l__xtable_x_dim
1990   { \seq_item:Nn \g__xtable_col_sep_seq { \l__xtable_tmpb_int + 1 } / 2 }
1991   \dim_sub:Nn \l__xtable_y_dim { \c__xtable_std_dp_dim / 3 }
1992   \dim_sub:Nn \l__xtable_y_dim
1993   { \seq_item:Nn \g__xtable_row_sep_seq { \l__xtable_tmpa_int + 1 } / 2 }
1994   \__xtable_parse_merge_size:n {#1}
1995   \__xtable_parse_merge_fill:n {#1}
1996   \__xtable_parse_merge_ht:n {#1}
1997   \__xtable_parse_merge_align:n {#1}
1998   \__xtable_render_data:VVV % 获取包含渲染内容的盒子
1999   \l__xtable_row_align_tl \l__xtable_col_align_tl \l__xtable_data_tl
2000   \tl_if_eq:NnTF \l__xtable_row_align_tl {m}
2001   {
2002     \dim_set:Nn \l__xtable_tmpb_dim
2003     { \box_ht:N \l__xtable_cell_box + \l__xtable_ufill_dim }

```

```

2004     \dim_set:Nn \l__xtable_tmpa_dim
2005     { \l__xtable_y_dim - \l__xtable_tmpb_dim }
2006     \dim_sub:Nn \l__xtable_tmpa_dim
2007     { (\l__xtable_ht_dim - \l__xtable_tmpb_dim) / 2 }
2008   }
2009   {
2010     \dim_set:Nn \l__xtable_tmpa_dim
2011     { \l__xtable_y_dim - \l__xtable_ht_dim }
2012   }
2013   \draw_box_use:Nn \l__xtable_cell_box
2014     { \l__xtable_x_dim, \l__xtable_tmpa_dim }
2015   }

```

(End of definition for __xtable_draw_merge:n.)

__xtable_draw_cells: 绘制所有单元格，使用 \l__xtable_x_dim 与 \l__xtable_y_dim 的存储内容的左上角坐标信息。

```

2016 \cs_new:Nn \__xtable_draw_cells:
2017   {
2018     \dim_set:Nn \l__xtable_y_dim { -\c__xtable_std_dp_dim / 3 } % 补偿
2019     \bool_if:NTF \g__xtable_col_header_bool
2020       { \int_set:Nn \l__xtable_tmpa_int {0} }
2021       { \int_set:Nn \l__xtable_tmpa_int {1} }
2022     \int_step_inline:nnn {\l__xtable_tmpa_int} {\g__xtable_row_count_int}
2023     {
2024       \dim_sub:Nn \l__xtable_y_dim
2025       { \seq_item:Nn \g__xtable_row_sep_seq {##1+1} }
2026       \dim_zero:N \l__xtable_x_dim
2027       \bool_if:NTF \g__xtable_row_header_bool
2028         { \int_set:Nn \l__xtable_tmpb_int {0} }
2029         { \int_set:Nn \l__xtable_tmpb_int {1} }
2030       \int_step_inline:nnn { \l__xtable_tmpb_int } { \g__xtable_col_count_int }
2031       {
2032         \dim_add:Nn \l__xtable_x_dim
2033         { \seq_item:Nn \g__xtable_col_sep_seq {####1 + 1} }
2034         \__xtable_draw_cell:nn {##1} {####1}
2035         \dim_add:Nn \l__xtable_x_dim { \l__xtable_wd_dim }
2036       }
2037       \dim_sub:Nn \l__xtable_y_dim
2038       { \l__xtable_ht_dim + \l__xtable_dp_dim }
2039     }
2040     \int_step_inline:nn { \__xtable_parse_merge_count: }
2041     { \__xtable_draw_merge:n {##1} }

```

```
2042 }
```

(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。

```
2043 \cs_new:Nn \_xtable\_draw\_booktabs:n
2044 {
2045   \dim_zero:N \l\_xtable\_expand\_start\_dim
2046   \dim_zero:N \l\_xtable\_expand\_end\_dim
2047   \_xtable\_set\_rule\_style:n {toprule}
2048   \_xtable\_draw\_hrule:n {1}
2049   \draw\_path\_use\_clear:n { stroke }
2050   \bool_if:NT \g\_xtable\_col\_header\_bool
2051     {
2052       \_xtable\_set\_rule\_style:n {midrule}
2053       \_xtable\_draw\_hrule:n {2}
2054       \draw\_path\_use\_clear:n { stroke }
2055     }
2056   \_xtable\_set\_rule\_style:n {bottomrule}
2057   \_xtable\_draw\_hrule:n {-1}
2058   \draw\_path\_use\_clear:n { stroke }
2059   \_xtable\_set\_rule\_style:n {cmidrule}
2060   \clist_map_inline:nn {#1}
2061     { \_xtable\_draw\_hrule:n {##1+2} }
2062   \draw\_path\_use\_clear:n { stroke }
2063 }
2064 \cs_generate_variant:Nn \_xtable\_draw\_booktabs:n { V }
```

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

7.3.3 渲染边框-网格线

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

```
2065 \NewDocumentCommand {\_xtable\_row\_rule:nn} { 0{ } m }
2066 {
2067   \tl_set:Nn \l\_xtable\_style\_tl {#1}
2068   \tl_if_empty:nTF {#2}
2069     { \seq_clear:N \l\_xtable\_style\_seq }
2070     {
2071       \seq_set_split:Nnn \l\_xtable\_style\_seq {,} {#2}
2072       \seq_get_left:NN \l\_xtable\_style\_seq \l\_xtable\_tmpa\_tl
```

```

2073     \str_if_in:NnF \l__xtable_tmpa_tl {=}
2074     { \seq_pop_left:NN \l__xtable_style_seq \l__xtable_style_tl }
2075   }
2076   \__xtable_set_line_style:NNnn \g__xtable_row_rule_seq
2077     \l__xtable_style_seq {#1} {row}
2078   \bool_if:NF \g__xtable_col_header_bool
2079     { \seq_gpop_left:NN \g__xtable_row_rule_seq \l__xtable_tmpa_tl }
2080   \seq_gput_left:NV \g__xtable_row_rule_seq \l__xtable_style_tl
2081 }

```

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

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

```

2082 \NewDocumentCommand {\__xtable_col_rule:nn} { 0{ } m }
2083 {
2084   \tl_set:Nn \l__xtable_style_tl {#1}
2085   \tl_if_empty:nTF {#2}
2086     { \seq_clear:N \l__xtable_style_seq }
2087     {
2088       \seq_set_split:Nnn \l__xtable_style_seq {,} {#2}
2089       \seq_get_left:NN \l__xtable_style_seq \l__xtable_tmpa_tl
2090       \str_if_in:NnF \l__xtable_tmpa_tl {=}
2091       { \seq_pop_left:NN \l__xtable_style_seq \l__xtable_style_tl }
2092     }
2093   \__xtable_set_line_style:NNnn \g__xtable_col_rule_seq
2094     \l__xtable_style_seq {#1} {col}
2095   \seq_gput_left:NV \g__xtable_col_rule_seq \l__xtable_style_tl
2096 }

```

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

`__xtable_calc_grid_expand:N` 计算网格线的扩展长度²³。

```

2097 \cs_new:Nn \__xtable_calc_grid_expand:N
2098 {
2099   \seq_if_empty:NTF #1
2100     { \seq_set_split:Nnn \l__xtable_tmpa_seq {,} {0pt} }
2101     { \seq_set_eq:NN \l__xtable_tmpa_seq #1 }
2102   \seq_get_left:NN \l__xtable_tmpa_seq \l__xtable_style_tl
2103   \tl_if_empty:NTF \l__xtable_style_tl
2104     { \dim_zero:N \l__xtable_expand_start_dim }
2105     {
2106       \__xtable_parse_rule_style:V \l__xtable_style_tl

```

²³用于平衡线宽造成的转角过渡问题。

```

2107     \dim_set:Nn \l__xtable_expand_start_dim
2108     { \l__xtable_rule_dim/2 }
2109   }
2110   \seq_get_right:NN \l__xtable_tmpa_seq \l__xtable_style_tl
2111   \tl_if_empty:NTF \l__xtable_style_tl
2112     { \dim_zero:N \l__xtable_expand_end_dim }
2113     {
2114       \__xtable_parse_rule_style:V \l__xtable_style_tl
2115       \dim_set:Nn \l__xtable_expand_end_dim
2116         { \l__xtable_rule_dim/2 }
2117     }
2118 }

```

(End of definition for __xtable_calc_grid_expand:N.)

`__xtable_draw_grid:n` 绘制网格线型表格。

```

2119 \cs_new:Nn \__xtable_draw_grid:n
2120 {
2121   \group_begin:
2122     \cs_set_eq:NN \hrule \__xtable_row_rule:nn
2123     \cs_set_eq:NN \vrule \__xtable_col_rule:nn
2124     #1
2125   \group_end:
2126   \__xtable_calc_grid_expand:N \g__xtable_row_rule_seq
2127   \int_step_inline:nn { \seq_count:N \g__xtable_col_loc_seq }
2128     {
2129       \tl_set:Ne \l__xtable_style_tl
2130         { \seq_item:Nn \g__xtable_col_rule_seq {##1} }
2131       \tl_if_empty:NF \l__xtable_style_tl
2132         {
2133           \__xtable_set_rule_style:V \l__xtable_style_tl
2134           \__xtable_draw_vrule:n {##1}
2135           \draw_path_use_clear:n { stroke }
2136         }
2137     }
2138   \__xtable_calc_grid_expand:N \g__xtable_col_rule_seq
2139   \int_step_inline:nn { \seq_count:N \g__xtable_row_loc_seq }
2140     {
2141       \tl_set:Ne \l__xtable_style_tl
2142         { \seq_item:Nn \g__xtable_row_rule_seq {##1} }
2143       \tl_if_empty:NF \l__xtable_style_tl
2144         {
2145           \__xtable_set_rule_style:V \l__xtable_style_tl

```

```

2146         \__xtable_draw_hrule:n {##1}
2147         \draw_path_use_clear:n { stroke }
2148     }
2149 }
2150 }
2151 \cs_generate_variant:Nn \__xtable_draw_grid:n { V }

```

(End of definition for __xtable_draw_grid:n.)

7.3.4 渲染边框-全边框

__xtable_draw_cell_rule:nn 绘制指定单元格边框。

```

2152 \cs_new:Nn \__xtable_draw_cell_rule:nn
2153 {
2154     \dim_set:Nn \l__xtable_x_dim
2155         { \seq_item:Nn \g__xtable_col_loc_seq {#2+1} }
2156     \dim_set:Nn \l__xtable_tmpa_dim
2157         { \seq_item:Nn \g__xtable_col_loc_seq {#2+2} }
2158     \dim_set:Nn \l__xtable_y_dim
2159         { \seq_item:Nn \g__xtable_row_loc_seq {#1+2} }
2160     \dim_set:Nn \l__xtable_tmpb_dim
2161         { \seq_item:Nn \g__xtable_row_loc_seq {#1+1} }
2162     \dim_sub:Nn \l__xtable_tmpa_dim { \l__xtable_x_dim }
2163     \dim_sub:Nn \l__xtable_tmpb_dim { \l__xtable_y_dim }
2164     \draw_path_rectangle:nn
2165         { \l__xtable_x_dim, \l__xtable_y_dim }
2166         { \l__xtable_tmpa_dim, \l__xtable_tmpb_dim }
2167     \draw_path_use_clear:n { draw }
2168 }

```

(End of definition for __xtable_draw_cell_rule:nn.)

__xtable_draw_merge_rule:n 绘制指定单元格边框。

```

2169 \cs_new:Nn \__xtable_draw_merge_rule:n
2170 {
2171     \__xtable_parse_merge:n {#1}
2172     \dim_set:Nn \l__xtable_x_dim
2173         {
2174             \seq_item:Nn \g__xtable_col_loc_seq
2175                 { \seq_item:Nn \l__xtable_merge_seq {2} + 1 }
2176         }
2177     \dim_set:Nn \l__xtable_tmpa_dim
2178         {

```

```

2179     \seq_item:Nn \g__xtable_col_loc_seq
2180     { \seq_item:Nn \l__xtable_merge_seq {4} + 2 }
2181   }
2182 \dim_set:Nn \l__xtable_y_dim
2183 {
2184   \seq_item:Nn \g__xtable_row_loc_seq
2185   { \seq_item:Nn \l__xtable_merge_seq {3} + 2 }
2186 }
2187 \dim_set:Nn \l__xtable_tmpb_dim
2188 {
2189   \seq_item:Nn \g__xtable_row_loc_seq
2190   { \seq_item:Nn \l__xtable_merge_seq {1} + 1 }
2191 }
2192 \dim_sub:Nn \l__xtable_tmpa_dim { \l__xtable_x_dim }
2193 \dim_sub:Nn \l__xtable_tmpb_dim { \l__xtable_y_dim }
2194 \draw_path_rectangle:nn
2195   { \l__xtable_x_dim, \l__xtable_y_dim }
2196   { \l__xtable_tmpa_dim, \l__xtable_tmpb_dim }
2197 \draw_path_use_clear:n { draw }
2198 }

```

(End of definition for __xtable_draw_merge_rule:n.)

__xtable_draw_cell_rule: 绘制单元格边框。

```

2199 \cs_new:Nn \__xtable_draw_cell_rules:
2200 {
2201   \int_step_inline:nnn
2202     { \bool_if:NTF \g__xtable_col_header_bool {0} {1} }
2203     { \g__xtable_row_count_int }
2204     {
2205       \int_step_inline:nnn
2206         { \bool_if:NTF \g__xtable_row_header_bool {0} {1} }
2207         { \g__xtable_col_count_int }
2208         {
2209           \__xtable_merge_if:nF {##1, ####1}
2210           { \__xtable_draw_cell_rule:nn {##1} {####1} }
2211         }
2212       }
2213   \int_step_inline:nn { \__xtable_parse_merge_count: }
2214     { \__xtable_draw_merge_rule:n {##1} }
2215 }

```

(End of definition for __xtable_draw_cell_rule:.)

7.3.5 渲染选项

`\l__xtable_render_multicol_bool` 输入选项变量。

```

\l__xtable_render_multicol_bool 2216 \bool_new:N \l__xtable_render_multicol_bool
\l__xtable_render_longtable_bool 2217 \bool_new:N \l__xtable_render_longtable_bool
\l__xtable_render_format_tl 2218 \tl_new:N \l__xtable_render_format_tl
\l__xtable_render_rule_tl 2219 \tl_new:N \l__xtable_render_rule_tl

```

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

`xtable/render` 渲染选项。

```

2220 \keys_define:nn { xtable / render }
2221 {
2222   format .choices:nn = { border, grid, booktabs, custom }
2223   { \tl_set_eq:NN \l__xtable_render_format_tl \l_keys_choice_tl },
2224   format .initial:n = {border},
2225   booktabs .code:n = { \tl_set:Nn \l__xtable_render_format_tl { booktabs } },
2226   grid .code:n = { \tl_set:Nn \l__xtable_render_format_tl { grid } },
2227   border .code:n = { \tl_set:Nn \l__xtable_render_format_tl { border } },
2228   custom .code:n = { \tl_set:Nn \l__xtable_render_format_tl { custom } },
2229   multicol .bool_set:N = \l__xtable_render_multicol_bool,
2230   multicol .default:n = {true},
2231   multicol .initial:n = {false},
2232   longtable .bool_set:N = \l__xtable_render_longtable_bool,
2233   longtable .default:n = {true},
2234   longtable .initial:n = {false},
2235   rule .tl_set:N = \l__xtable_render_rule_tl,
2236   rule .initial:n = {}
2237 }

```

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

`xtable/render` 清空渲染选项。

```

2238 \cs_new:Nn \__xtable_clear_render_options:
2239 {
2240   \tl_clear:N \l__xtable_render_rule_tl
2241   \bool_set_false:N \l__xtable_render_multicol_bool
2242   \bool_set_false:N \l__xtable_render_longtable_bool
2243 }

```

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

7.3.6 渲染表格

`_xtable_set_table_sep:n` 设置表格边框序列。

```

2244 \cs_new:Nn \_xtable_set_table_sep:n
2245 {
2246   \_xtable_set_row_sep:ee
2247   { \dim_use:N \g__xtable_row_margin_dim }
2248   { \dim_eval:n { \g__xtable_row_margin_dim * 2 } }
2249   \_xtable_set_col_sep:ee
2250   {
2251     \bool_if:nTF {#1}
2252     { \dim_use:N \g__xtable_col_margin_dim }
2253     { Opt }
2254   }
2255   { \dim_eval:n { \g__xtable_col_margin_dim * 2 } }
2256 }

```

(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。

```

2257 \cs_new:Nn \_xtable_rendertable_pre:
2258 {
2259   \_xtable_set_table_sep:n {\c_true_bool}
2260   \_xtable_calc_col_wd:
2261   \_xtable_calc_row_ht:
2262   \_xtable_calc_merge_size:
2263   \_xtable_calc_coord:
2264   \draw_begin:
2265   \vbox_set_to_ht:Nnn \l__xtable_tmpa_box
2266   { \g__xtable_above_space_dim } {}
2267   \draw_box_use:N \l__xtable_tmpa_box
2268 }
2269 \cs_new:Nn \_xtable_rendertable_post:
2270 {
2271   \_xtable_support_sys_func:
2272   \_xtable_draw_cells:
2273   \draw_end:
2274   \_xtable_restore_sys_func:
2275   \_xtable_clear_render_options:
2276 }

```

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

`\rendertable` 绘制表格。

```
2277 \NewDocumentCommand \rendertable { 0{} }
2278 {
2279   \keys_set:nn { xtable/render } {#1}
2280   \__xtable_rendertable_pre:
2281   \str_case:NnF \l__xtable_render_format_tl
2282     {
2283       {booktabs} { \__xtable_draw_booktabs:V \l__xtable_render_rule_tl }
2284       {grid}      { \__xtable_draw_grid:V      \l__xtable_render_rule_tl }
2285     }
2286     { \__xtable_draw_cell_rules: }
2287   \__xtable_rendertable_post:
2288 }
```

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

```
2289 \endpackage
```

Change History

v0.1	添加独立的行列名设置功能	1
General: 内部测试		1
v0.2		
General: 初版发行		1
v0.3		
General: 修复表格保存、恢复、查看命令		1
整理优化内部接口		1
添加对无表头的表格的支持		1
添加按行或按列输入功能		1
v0.4		
General: 修改表头的数据结构并大幅重		
构内部函数		1
将渲染函数的参数改为键值接口		85
添加修改表格的功能		32
添加对合并单元格的支持		38
添加对脚注的支持		19
调整宏包组织结构及规范命名		1

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	environments:
·	<code>@@_data:</code>
..	<code>data</code>
<code>@@_data: (env.)</code>	<code>xtable</code>
<code>\l</code>	<code>\excelcolname</code>
C	F
<code>\cell</code>	<code>\file</code>
<code>\col</code>	<code>\footnote</code>
<code>\colalign</code>	<code>\footnotemark</code>
<code>\colname</code>	<code>\footnotetext</code>
<code>\colwidth</code>	
D	H
<code>data (env.)</code>	<code>\hrule</code>
<code>\data</code>	
<code>data</code>	
E	L
end internal commands:	<code>\loadtable</code>
<code>\end__xtable_data:</code>	<code>\logtable</code>
<code>\enddata</code>	
	M
	<code>\mergecell</code>

- P**
- `\printtable` 8, [1885](#)
- R**
- `\rendertable` 9, [86](#), 31, 43, [2277](#)
- `\row` 6, [1175](#)
- `\rowalign` 7, [1195](#)
- `\rowheight` 7, [1218](#)
- `\rowname` 4, [1108](#)
- `\rulepatternset` 10, [207](#)
- `\rulestyleset` 10, [207](#)
- `\rulewdset` 10, [207](#)
- S**
- `\savetable` 6, [1093](#)
- scan internal commands:
- `\s_htable_mark` [20](#), [24](#), [80](#), 218, 251,
 [267](#), [452](#), [499](#), [723](#), [724](#), [772](#), [794](#), 801
- `\showtable` 10, [805](#)
- V**
- `\vrule` 9, [2123](#)
- X**
- `xtable` (env.) [1175](#)
- `xtable` 4
- `xtable` internal commands:
- `\g_htable_above_space_dim`
 [49](#), 66, [2266](#)
- `\g_htable_below_space_dim` .. [49](#), 67
- `\l_htable_cachea_dim` ... [74](#), [115](#),
 [1381](#), [1405](#), [1419](#), [1445](#), [1452](#), [1505](#),
 [1531](#), [1551](#), [1559](#), [1827](#), [1839](#), [1912](#)
- `\l_htable_cachea_int`
 [35](#), [113](#), 1063, 1066,
 [1068](#), [1255](#), [1258](#), [1259](#), [1263](#), [1426](#),
 [1439](#), [1443](#), [1446](#), [1892](#), [1893](#), [1897](#)
- `\l_htable_cachea_prop`
 513, 530, 533, 536
- `\l_htable_cachea_seq`
 [119](#), [1383](#), [1404](#), [1416](#),
 [1427](#), [1440](#), [1448](#), [1507](#), [1530](#), [1543](#)
- `\l_htable_cachea_tl`
 [49](#), [117](#), 1057, 1064
- `\l_htable_cacheb_dim` . [115](#), [1382](#),
 [1410](#), [1424](#), [1506](#), [1532](#), [1551](#), [1559](#)
- `\l_htable_cacheb_int` . [113](#), [1256](#),
 [1260](#), [1261](#), [1263](#), [1895](#), [1896](#), [1904](#)
- `\l_htable_cacheb_prop` . 514, 546, 550
- `\l_htable_cacheb_seq`
 [119](#), [1384](#), [1409](#), [1421](#)
- `\l_htable_cacheb_tl`
 [49](#), [117](#), 1061, 1062, 1070
- `_htable_calc_col_wd:`
 [1379](#), [1379](#), [1889](#), [2260](#)
- `_htable_calc_coord:` [1708](#), [1708](#), [2263](#)
- `_htable_calc_grid_expand:N` ...
 [2097](#), [2097](#), [2126](#), [2138](#)
- `_htable_calc_merge_size:`
 [1622](#), [1622](#), [1890](#), [2262](#)
- `_htable_calc_row_ht:`
 [1503](#), [1503](#), [1889](#), [2261](#)
- `_htable_cell:mn` .. [1156](#), [1156](#), [1187](#)
- `\l_htable_cell_box`
 [76](#), [1772](#), [1941](#), [1965](#), [1975](#), [2003](#), [2013](#)
- `\c_htable_cell_box_tl` . [294](#), 466, 502
- `\c_htable_cell_content_tl`
 [294](#), 461, 795, 801
- `\g_htable_cell_data_prop`
 [297](#), [389](#), [427](#), [446](#),
 [451](#), [460](#), [497](#), [540](#), [550](#), [564](#), [578](#),
 [586](#), [589](#), [601](#), [604](#), [796](#), [802](#), [807](#), [833](#)
- `\g_htable_cell_dfill_prop`
 [1271](#), [1373](#), [1387](#), [1580](#)
- `\c_htable_cell_formula_tl` [294](#), 464
- `\g_htable_cell_ht_prop`
 [1271](#), [1494](#), [1510](#), [1587](#)
- `\l_htable_cell_input_seq`
 [876](#), [905](#),
 [911](#), [924](#), [929](#), [942](#), [959](#), [969](#), [973](#),
 [981](#), [1052](#), [1054](#), [1056](#), [1057](#), [1058](#),
 [1059](#), [1111](#), [1112](#), [1145](#), [1146](#), [1200](#),
 [1204](#), [1206](#), [1220](#), [1221](#), [1226](#), [1232](#)
- `\l_htable_cell_is_box_bool` 31
- `\g_htable_cell_lineskip_dim` ...
 . [49](#), [72](#), [1467](#), [1819](#), [1830](#), [1842](#), [1870](#)

- _xtable_cell_move_by_col:nnn [46](#), [61](#), [2252](#), [2255](#)
- [580](#), [595](#)
- _xtable_cell_move_by_row:nnn [580](#), [580](#)
- \g_xtable_cell_sep_dim [46](#), [56](#), [1882](#), [1887](#), [1888](#), [1918](#)
- \g_xtable_cell_ufill_prop [1271](#), [1371](#), [1386](#), [1578](#)
- \g_xtable_cell_wd_min_dim [49](#), [70](#), [1316](#), [1343](#)
- _xtable_check_input_env:n . [868](#), [868](#), [1095](#), [1100](#), [1105](#), [1120](#), [1125](#), [1210](#), [1215](#), [1236](#), [1241](#), [1246](#), [1253](#)
- _xtable_check_merge_range:nnnn [703](#), [703](#), [716](#)
- _xtable_clear_render_options: [2238](#), [2275](#)
- _xtable_col:nnn [1128](#), [1154](#), [1186](#)
- \g_xtable_col_align_seq [290](#), [382](#), [420](#), [439](#), [642](#), [676](#), [691](#), [826](#)
- \l_xtable_col_align_tl [37](#), [40](#), [103](#), [690](#), [782](#), [1917](#), [1961](#), [1999](#)
- \g_xtable_col_count_int [286](#), [317](#), [361](#), [380](#), [418](#), [437](#), [454](#), [598](#), [599](#), [813](#), [824](#), [1300](#), [1390](#), [1393](#), [1432](#), [1485](#), [1722](#), [1904](#), [2030](#), [2207](#)
- \l_xtable_col_count_int [562](#), [606](#)
- \g_xtable_col_header_bool [284](#), [394](#), [416](#), [435](#), [836](#), [1182](#), [1291](#), [1359](#), [1512](#), [1738](#), [1891](#), [2019](#), [2050](#), [2078](#), [2202](#)
- \l_xtable_col_loc_int [26](#), [27](#), [30](#), [35](#), [45](#), [46](#), [49](#), [50](#), [52](#), [53](#), [99](#), [340](#), [358](#), [360](#), [361](#), [362](#), [482](#), [487](#), [492](#), [910](#), [928](#), [957](#), [1063](#), [1065](#), [1066](#), [1067](#), [1068](#), [1131](#), [1143](#), [1159](#), [1256](#), [1260](#), [1261](#), [1262](#)
- \g_xtable_col_loc_seq [1282](#), [1711](#), [1712](#), [1717](#), [1730](#), [1734](#), [1777](#), [1782](#), [1790](#), [1795](#), [1988](#), [2127](#), [2155](#), [2157](#), [2174](#), [2179](#)
- \g_xtable_col_margin_dim
- [46](#), [61](#), [2252](#), [2255](#)
- \g_xtable_col_name_prop [288](#), [305](#), [316](#), [340](#), [358](#), [388](#), [392](#), [426](#), [445](#), [790](#), [832](#)
- _xtable_col_rule:nn [2082](#), [2082](#), [2123](#)
- \g_xtable_col_rule_seq [1769](#), [2093](#), [2095](#), [2130](#), [2138](#)
- \g_xtable_col_sep_seq [1266](#), [1299](#), [1301](#), [1302](#), [1303](#), [1306](#), [1307](#), [1430](#), [1641](#), [1720](#), [1725](#), [1729](#), [1733](#), [1990](#), [2033](#)
- \c_xtable_col_wd_regex [852](#)
- \g_xtable_col_wd_seq [1268](#), [1385](#), [1392](#), [1400](#), [1413](#), [1418](#), [1423](#), [1435](#), [1451](#), [1453](#), [1492](#), [1570](#), [1648](#), [1727](#)
- \g_xtable_col_wd_style_seq [292](#), [384](#), [422](#), [441](#), [659](#), [661](#), [664](#), [668](#), [670](#), [680](#), [701](#), [828](#), [1429](#)
- \c_xtable_csv_cell_regex . [838](#), [961](#)
- \c_xtable_csv_row_regex . . [838](#), [953](#)
- _xtable_data: [1183](#)
- _xtable_data_move:nnn [553](#), [553](#)
- \l_xtable_data_tl [31](#), [40](#), [41](#), [59](#), [101](#), [501](#), [505](#), [816](#), [1368](#), [1369](#), [1493](#), [1603](#), [1604](#), [1615](#), [1872](#), [1917](#), [1957](#), [1961](#), [1999](#)
- \l_xtable_dfill_dim [58](#), [59](#), [62](#), [63](#), [66](#), [69](#), [108](#), [1315](#), [1332](#), [1333](#), [1342](#), [1351](#), [1374](#), [1473](#), [1583](#), [1608](#), [1698](#), [1875](#)
- _xtable_dim_set_max:NN [151](#), [151](#), [1370](#), [1405](#), [1410](#), [1496](#), [1497](#), [1531](#), [1532](#)
- _xtable_dim_set_max:Nn [151](#), [153](#), [1324](#), [1328](#), [1333](#), [1347](#), [1349](#), [1351](#), [1470](#), [1472](#)
- _xtable_disable_sys_func: [179](#), [179](#), [1388](#), [1511](#), [1627](#)
- \l_xtable_dp_dim [62](#), [63](#), [65](#), [69](#), [73](#), [108](#), [1460](#), [1472](#), [1497](#), [1501](#), [1527](#), [1532](#), [1537](#), [1539](#), [1550](#), [1553](#), [1556](#), [1559](#), [1573](#), [1619](#), [1666](#), [1681](#),

- 1683, 1693, 1849, 1850, 1876, 2038
- _xtable_draw_booktabs:n
- 2043, 2043, 2064, 2283
- _xtable_draw_cell:nn
- 1948, 1948, 2034
- _xtable_draw_cell_rule: 2199
- _xtable_draw_cell_rule:nn
- 2152, 2152, 2210
- _xtable_draw_cell_rules: 2199, 2286
- _xtable_draw_cells: 2016, 2016, 2272
- _xtable_draw_grid:n
- 2119, 2119, 2151, 2284
- _xtable_draw_hruler:n 1773,
- 1773, 2048, 2053, 2057, 2061, 2146
- _xtable_draw_merge:n
- 1978, 1978, 2041
- _xtable_draw_merge_rule:n
- 2169, 2169, 2214
- _xtable_draw_vrule:n
- 1786, 1786, 2134
- _xtable_en_msg_support: . . . 33, 55
- \l_xtable_esc_status_bool
- 873, 1023, 1024, 1027, 1077
- \c_xtable_escape_str 76, 1026
- \l_xtable_expand_end_dim . . 72,
- 73, 1767, 1782, 1796, 2046, 2112, 2115
- \l_xtable_expand_start_dim . 72,
- 73, 1767, 1777, 1791, 2045, 2104, 2107
- _xtable_footnote: 158, 158, 182
- _xtable_footnote_print:
- 162, 174, 192
- \g_xtable_footnote_prop
- 155, 167, 171, 176, 181, 187
- _xtable_footnote_support:
- 162, 162, 188
- _xtable_footnotemark: 158, 160, 183
- _xtable_get_line_loc:NnN 321
- _xtable_get_line_loc:NnNTF
- 321, 333, 339, 347, 357
- _xtable_ginit_seq:Nnn 135, 140, 621
- _xtable_header_fill: 785, 785, 1193
- \l_xtable_ht_dim 62, 63, 65, 66, 69,
- 70, 73, 108, 1459, 1470, 1495, 1496,
- 1500, 1525, 1531, 1548, 1551, 1558,
- 1561, 1571, 1588, 1617, 1663, 1676,
- 1678, 1691, 1705, 1735, 1737, 1742,
- 1744, 1749, 1751, 1753, 1755, 1757,
- 1759, 1761, 1827, 1828, 1839, 1875,
- 1912, 1969, 1973, 2007, 2011, 2038
- _xtable_init_seq:Nnn . 135, 135, 145
- \l_xtable_input_bool
- 866, 870, 1188, 1191
- \l_xtable_input_col_header_bool
- . . . 878, 889, 890, 917, 971, 978, 1182
- \l_xtable_input_format_tl
- 878, 885, 887, 888, 1166, 1172
- \l_xtable_input_row_header_bool
- 878, 892, 922, 940, 1180
- \l_xtable_input_sep_tl 878, 894, 906
- _xtable_insure_style: 672, 672, 1192
- _xtable_int_gset_max:Nn
- 146, 148, 150,
- 312, 317, 453, 454, 573, 591, 606, 619
- _xtable_int_set_max:Nn . . 146, 146
- \l_xtable_is_box_bool
- . . 40, 101, 496, 503, 1367, 1602, 1956
- \c_xtable_json_cell_regex 842, 1052
- \c_xtable_lbrace_str . . 76, 997, 1013
- _xtable_line:nmm
- 1128, 1128, 1153, 1155
- _xtable_line_align:nmm
- 1195, 1195, 1211, 1216
- _xtable_line_name:Nmm
- 1108, 1108, 1121, 1126
- _xtable_line_size:nmm
- 1218, 1218, 1237, 1242
- _xtable_measure_cell_box_wd:n
- 1339, 1339, 1355, 1368, 1603
- _xtable_measure_cell_ht:nmm
- 1457, 1457, 1476, 1491, 1613
- _xtable_measure_cell_wd:n
- 1312, 1312, 1338, 1369, 1604

- __xtable_measure_col_wd:n [1356](#), [1356](#), [1396](#)
- __xtable_measure_merge_size: [1590](#), [1590](#), [1628](#)
- __xtable_measure_row_ht:n [1477](#), [1477](#), [1520](#)
- \c__xtable_merge_align_prop [83](#), [719](#), [752](#)
- \g__xtable_merge_align_seq . [298](#),
[386](#), [424](#), [443](#), [747](#), [751](#), [754](#), [779](#), [830](#)
- \g__xtable_merge_dfill_seq [1278](#), [1595](#), [1607](#), [1699](#)
- \g__xtable_merge_dp_seq [1274](#), [1594](#), [1618](#)
- \g__xtable_merge_ht_seq [1274](#), [1593](#), [1616](#), [1706](#)
- \l__xtable_merge_id_tl [40](#), [106](#), [765](#), [1857](#), [1862](#), [1863](#)
- __xtable_merge_if:n [756](#), [756](#)
- __xtable_merge_if:nTF [1364](#), [1487](#), [1906](#), [1954](#), [2209](#)
- \g__xtable_merge_info_seq [298](#), [385](#),
[423](#), [442](#), [718](#), [721](#), [749](#), [762](#), [771](#), [829](#)
- \c__xtable_merge_loc_prop [83](#)
- \g__xtable_merge_range_dp_seq [1279](#), [1626](#), [1677](#), [1680](#), [1694](#)
- \g__xtable_merge_range_ht_seq [1279](#), [1625](#), [1675](#), [1682](#), [1692](#)
- \g__xtable_merge_range_wd_seq [1279](#), [1624](#), [1650](#), [1690](#)
- \g__xtable_merge_ref_prop . . [298](#),
[390](#), [428](#), [447](#), [709](#), [738](#), [758](#), [765](#), [834](#)
- \l__xtable_merge_seq [40](#), [74](#), [106](#), [772](#), [773](#),
[1637](#), [1638](#), [1654](#), [1655](#), [1859](#), [1865](#),
[1982](#), [1984](#), [2175](#), [2180](#), [2185](#), [2190](#)
- \g__xtable_merge_ufill_seq [1277](#), [1596](#), [1609](#), [1701](#)
- \g__xtable_merge_wd_seq [1274](#), [1592](#), [1605](#)
- \l__xtable_offset_x_dim [1765](#)
- \l__xtable_offset_y_dim [1765](#)
- __xtable_origin_footnote: [156](#), [156](#), [193](#)
- __xtable_origin_footnotemark: [156](#), [157](#), [159](#), [161](#), [166](#), [170](#), [194](#)
- __xtable_parse_cell:n . [494](#), [494](#),
[507](#), [773](#), [815](#), [1366](#), [1489](#), [1909](#), [1959](#)
- __xtable_parse_cell_fill:n [1490](#), [1576](#), [1576](#), [1910](#), [1950](#)
- __xtable_parse_cell_ht:n [1585](#), [1585](#), [1911](#)
- __xtable_parse_cell_size:n [1565](#), [1565](#), [1874](#), [1913](#), [1951](#)
- __xtable_parse_col_align:n [683](#), [688](#), [1915](#), [1953](#)
- __xtable_parse_col_loc:n [331](#), [337](#), [344](#), [370](#)
- __xtable_parse_col_wd_style:n [693](#), [698](#), [1397](#), [1436](#)
- __xtable_parse_coord:n [367](#), [367](#), [896](#), [1254](#), [1257](#)
- __xtable_parse_csv:n . [949](#), [949](#), [1169](#)
- __xtable_parse_input:n [899](#), [899](#), [1168](#)
- __xtable_parse_json:n [1074](#), [1074](#), [1170](#)
- __xtable_parse_json_auxa:n [990](#), [990](#), [1086](#)
- __xtable_parse_json_auxb:n [1005](#), [1005](#), [1087](#)
- __xtable_parse_json_auxc:n [1021](#), [1021](#), [1088](#)
- __xtable_parse_json_row:n [1049](#), [1049](#), [1091](#)
- __xtable_parse_merge:n [768](#),
[768](#), [775](#), [1600](#), [1633](#), [1857](#), [1980](#), [2171](#)
- __xtable_parse_merge_align:n [776](#), [776](#), [1612](#), [1634](#), [1997](#)
- __xtable_parse_merge_count: [761](#), [761](#), [1598](#), [1631](#), [2040](#), [2213](#)
- __xtable_parse_merge_fill:n [1696](#), [1696](#), [1863](#), [1995](#)

- _xtable_parse_merge_ht:n
 [1703](#), [1703](#), [1996](#)
- _xtable_parse_merge_id:n
 [763](#), [763](#), [1856](#)
- _xtable_parse_merge_size:n
 [1687](#), [1687](#), [1862](#), [1994](#)
- _xtable_parse_new_col_loc:n
 [345](#), [355](#), [366](#), [375](#), [1064](#)
- _xtable_parse_new_coord:n
 [367](#), [372](#), [1139](#), [1160](#)
- _xtable_parse_new_row_loc:n
 [345](#), [345](#), [365](#), [374](#)
- _xtable_parse_row_align:n
 [683](#), [683](#), [1481](#), [1545](#), [1914](#), [1952](#)
- _xtable_parse_row_ht_style:n
 [693](#), [693](#), [1521](#)
- _xtable_parse_row_loc:n
 [331](#), [331](#), [343](#), [369](#)
- _xtable_parse_rule_style:n
 [262](#), [262](#), [283](#), [2106](#), [2114](#)
- \l_xtable_parse_status_int . [873](#),
 [996](#), [999](#), [1012](#), [1015](#), [1040](#), [1076](#), [1084](#)
- _xtable_print_data:nnn
 [1799](#), [1799](#), [1853](#), [1916](#)
- _xtable_print_merge_data:nn
 [1854](#), [1854](#), [1907](#)
- \l_xtable_quote_status_bool
 [873](#), [1029](#), [1032](#), [1036](#), [1078](#)
- \c_xtable_rbrace_str [76](#), [1038](#)
- _xtable_render_data:nnn
 [1927](#), [1927](#), [1947](#), [1960](#), [1998](#)
- \l_xtable_render_format_tl [2216](#),
 [2223](#), [2225](#), [2226](#), [2227](#), [2228](#), [2281](#)
- \l_xtable_render_longtable_bool
 [2216](#), [2232](#), [2242](#)
- \l_xtable_render_multicol_bool
 [2216](#), [2229](#), [2241](#)
- \l_xtable_render_rule_tl
 [2216](#), [2235](#), [2240](#), [2283](#), [2284](#)
- _xtable_rendertable_post:
 [2257](#), [2269](#), [2287](#)
- _xtable_rendertable_pre:
 [2257](#), [2257](#), [2280](#)
- _xtable_restore_sys_func:
 [179](#), [190](#), [1415](#), [1542](#), [1629](#), [1924](#), [2274](#)
- _xtable_row:nnn [1128](#), [1152](#), [1185](#)
- \g_xtable_row_align_seq
 [290](#), [381](#), [419](#), [438](#), [640](#), [674](#), [686](#), [825](#)
- \l_xtable_row_align_tl [37](#), [40](#),
 [63](#), [103](#), [685](#), [780](#), [1491](#), [1546](#), [1613](#),
 [1673](#), [1917](#), [1961](#), [1962](#), [1999](#), [2000](#)
- \g_xtable_row_count_int [286](#), [312](#),
 [351](#), [379](#), [417](#), [436](#), [453](#), [583](#), [584](#),
 [798](#), [810](#), [823](#), [1287](#), [1362](#), [1513](#),
 [1517](#), [1747](#), [1897](#), [1923](#), [2022](#), [2203](#)
- \l_xtable_row_count_int [559](#), [591](#)
- \g_xtable_row_dp_seq
 [1268](#), [1509](#), [1516](#), [1526](#), [1538](#),
 [1552](#), [1557](#), [1574](#), [1667](#), [1671](#), [1754](#)
- \g_xtable_row_header_bool
 [284](#), [393](#), [415](#), [434](#), [835](#), [1180](#), [1304](#),
 [1389](#), [1482](#), [1713](#), [1894](#), [2027](#), [2206](#)
- \c_xtable_row_ht_regex [852](#)
- \g_xtable_row_ht_seq
 [1268](#), [1508](#), [1515](#), [1524](#), [1536](#),
 [1549](#), [1560](#), [1572](#), [1664](#), [1669](#), [1752](#)
- \g_xtable_row_ht_style_seq
 [292](#), [383](#), [421](#), [440](#),
 [645](#), [647](#), [650](#), [653](#), [655](#), [678](#), [696](#), [827](#)
- \l_xtable_row_input_seq
 [876](#), [901](#), [920](#),
 [936](#), [951](#), [953](#), [955](#), [1041](#), [1080](#), [1091](#)
- \l_xtable_row_loc_int [26](#), [27](#),
 [30](#), [35](#), [45](#), [46](#), [49](#), [50](#), [52](#), [53](#), [99](#),
 [334](#), [348](#), [350](#), [351](#), [352](#), [482](#), [487](#),
 [492](#), [935](#), [954](#), [1070](#), [1072](#), [1130](#),
 [1144](#), [1158](#), [1255](#), [1258](#), [1259](#), [1262](#)
- \g_xtable_row_loc_seq
 [1282](#), [1736](#), [1737](#), [1742](#),
 [1757](#), [1761](#), [1778](#), [1783](#), [1791](#), [1796](#),
 [1986](#), [2139](#), [2159](#), [2161](#), [2184](#), [2189](#)
- \g_xtable_row_margin_dim
 [46](#), [60](#), [2247](#), [2248](#)

- \g_xtable_row_name_prop [288](#), [311](#),
[334](#), [348](#), [387](#), [391](#), [425](#), [444](#), [788](#), [831](#)
- _xtable_row_rule:nn [2065](#), [2065](#), [2122](#)
- \g_xtable_row_rule_seq
.. [1769](#), [2076](#), [2079](#), [2080](#), [2126](#), [2142](#)
- \g_xtable_row_sep_seq [1266](#), [1286](#),
[1288](#), [1289](#), [1290](#), [1293](#), [1294](#), [1658](#),
[1745](#), [1750](#), [1756](#), [1760](#), [1993](#), [2025](#)
- \l_xtable_rule_color_tl
..... [22](#), [201](#), [270](#), [279](#), [280](#)
- \l_xtable_rule_dim
..... [201](#), [271](#), [276](#), [2108](#), [2116](#)
- \g_xtable_rule_pattern_prop ...
..... [198](#), [208](#), [213](#), [236](#)
- \l_xtable_rule_pattern_tl
..... [22](#), [201](#),
[213](#), [214](#), [218](#), [236](#), [237](#), [268](#), [277](#), [278](#)
- \g_xtable_rule_style_prop
..... [198](#), [217](#), [248](#), [264](#)
- \l_xtable_rule_style_seq
..... [201](#), [250](#), [252](#), [253](#), [254](#), [266](#)
- \l_xtable_rule_style_tl
..... [201](#), [248](#), [251](#), [264](#), [267](#)
- \g_xtable_rule_wd_prop
..... [198](#), [210](#), [215](#), [242](#), [274](#)
- \l_xtable_rule_wd_tl
..... [22](#), [201](#), [215](#), [216](#),
[218](#), [242](#), [243](#), [269](#), [271](#), [274](#), [275](#), [276](#)
- _xtable_set_cell:n
..... [479](#), [479](#), [1148](#), [1161](#)
- _xtable_set_cell:nn
..... [457](#), [457](#), [467](#), [471](#), [925](#), [943](#)
- _xtable_set_cell:NNn
... [470](#), [470](#), [476](#), [481](#), [913](#), [983](#), [1069](#)
- _xtable_set_cell:nmnn
..... [449](#), [449](#), [456](#), [461](#), [464](#), [466](#)
- _xtable_set_cell_box:n .. [479](#), [489](#)
- _xtable_set_cell_box:nn
..... [457](#), [465](#), [469](#), [475](#)
- _xtable_set_cell_box:NNn
..... [470](#), [474](#), [478](#), [491](#)
- _xtable_set_cell_formula:n [479](#), [484](#)
- _xtable_set_cell_formula:nn ..
..... [457](#), [463](#), [468](#), [473](#)
- _xtable_set_cell_formula:NNn ..
..... [470](#), [472](#), [477](#), [486](#)
- _xtable_set_col_align:Nn
..... [639](#), [641](#), [677](#)
- _xtable_set_col_name:nN
... [309](#), [314](#), [320](#), [362](#), [931](#), [975](#), [1126](#)
- _xtable_set_col_sep:nn
..... [1284](#), [1297](#), [1311](#), [1888](#), [2249](#)
- _xtable_set_col_wd_style:Nnn ..
..... [643](#), [657](#), [681](#)
- _xtable_set_excel_col_names:n
..... [301](#), [301](#), [1106](#)
- _xtable_set_line_style:NNNn .. [610](#)
- _xtable_set_line_style:NNnn ..
.. [610](#), [640](#), [642](#), [645](#), [659](#), [2076](#), [2093](#)
- _xtable_set_merge_align:n
..... [744](#), [744](#), [1264](#)
- _xtable_set_merge_cell:nmnnn ..
..... [714](#), [714](#), [743](#), [1262](#)
- _xtable_set_row_align:Nn
..... [639](#), [639](#), [675](#)
- _xtable_set_row_ht_style:Nnn ..
..... [643](#), [643](#), [679](#)
- _xtable_set_row_name:nN
..... [309](#), [309](#), [319](#), [352](#), [1121](#)
- _xtable_set_row_sep:nn
..... [1284](#), [1284](#), [1310](#), [1887](#), [2246](#)
- _xtable_set_rule_pattern:n ...
..... [234](#), [234](#), [256](#)
- _xtable_set_rule_style:n
..... [234](#), [246](#),
[261](#), [2047](#), [2052](#), [2056](#), [2059](#), [2133](#), [2145](#)
- _xtable_set_rule_wd:n [234](#), [240](#), [257](#)
- _xtable_set_table_sep:n
..... [2244](#), [2244](#), [2259](#)
- \c_xtable_space_str ... [76](#), [994](#), [1009](#)
- \c_xtable_std_dp_dim
.. [58](#), [81](#), [1334](#), [1352](#), [1460](#), [1991](#), [2018](#)
- \c_xtable_std_ht_dim
..... [58](#), [81](#), [1329](#), [1350](#), [1459](#)

\l__xtable_style_seq	1485, 1567, 1572, 1574, 1635, 1642,
... 1771 , 2069 , 2071 , 2072 , 2074 ,	1645, 1652, 1659, 1662, 1714, 1716,
2077 , 2086 , 2088 , 2089 , 2091 , 2094	1720, 1722, 1739, 1741, 1745, 1747,
\l__xtable_style_tl	1981, 1986, 1993, 2020, 2021, 2022
... 37 , 103 , 695 , 700 , 1398 , 1402 ,	\l__xtable_tmpa_prop
1407 , 1413 , 1437 , 1522 , 1528 , 1537 ,	129 , 511 , 518 , 524 , 787 , 789 , 791 , 792
2067 , 2074 , 2080 , 2084 , 2091 , 2095 ,	\l__xtable_tmpa_seq
2102 , 2103 , 2106 , 2110 , 2111 , 2114 ,	... 129 , 268 , 269 , 270 , 499 , 500 ,
2129 , 2131 , 2133 , 2141 , 2143 , 2145	501 , 515 , 517 , 520 , 523 , 555 , 570 ,
__xtable_support_sys_func:	574 , 578 , 627 , 628 , 629 , 958 , 961 ,
... 179 , 185 , 1899 , 2271	962 , 965 , 967 , 1197 , 1224 , 1229 ,
__xtable_table_init: . 377 , 377 , 1177	1320 , 1321 , 2100 , 2101 , 2102 , 2110
__xtable_table_restore:n	\l__xtable_tmpa_str 125 , 538 ,
... 430 , 430 , 1096	542 , 547 , 1000 , 1016 , 1041 , 1047 , 1079
__xtable_table_restructure:nn .	\l__xtable_tmpa_tl
... 508 , 508	125 , 326 , 327 , 460 , 497 , 499 , 500 ,
__xtable_table_save:n 396 , 396 , 1101	502 , 522 , 524 , 525 , 529 , 530 , 586 ,
__xtable_tmapa_cs: . 1247 , 1259 , 1261	590 , 601 , 605 , 628 , 630 , 649 , 651 ,
\l__xtable_tmpa_bool 1319 , 1326 , 1330	652 , 653 , 663 , 665 , 666 , 667 , 668 ,
\l__xtable_tmpa_box	719 , 720 , 725 , 752 , 753 , 754 , 770 ,
133 , 1323 , 1325 , 1329 , 1334 , 1346 ,	772 , 778 , 781 , 783 , 794 , 795 , 796 ,
1348 , 1350 , 1352 , 1465 , 1471 , 1473 ,	800 , 802 , 904 , 906 , 920 , 938 , 964 ,
1817 , 1828 , 1840 , 1849 , 1850 , 1851 ,	968 , 969 , 1056 , 1058 , 1060 , 1062 ,
1868 , 1875 , 1876 , 1877 , 2265 , 2267	1202 , 1203 , 1204 , 1578 , 1579 , 1582 ,
__xtable_tmpa_cs:	1587 , 1588 , 1806 , 1807 , 1808 , 1810 ,
196 , 196 , 538 , 539 , 544 , 558 , 561 ,	1812 , 1821 , 1833 , 1845 , 1858 , 1860 ,
568 , 572 , 573 , 622 , 630 , 902 , 921 ,	1864 , 1866 , 1931 , 1932 , 1933 , 1935 ,
939 , 1143 , 1144 , 1149 , 1222 , 1224 , 1229	1944 , 2072 , 2073 , 2079 , 2089 , 2090
\l__xtable_tmpa_dim	\l__xtable_tmpb_box
... 59 , 63 , 123 , 1358 ,	133
1370 , 1377 , 1479 , 1496 , 1500 , 1657 ,	__xtable_tmpb_cs:
1661 , 1668 , 1670 , 1678 , 1683 , 1826 ,	196 , 197 , 559 , 562 , 573 , 908 , 927 , 945
1832 , 1838 , 1844 , 1902 , 1966 , 1968 ,	\l__xtable_tmpb_dim
1972 , 1976 , 2004 , 2006 , 2010 , 2014 ,	... 63 , 123 , 1480 , 1497 , 1501 ,
2156 , 2162 , 2166 , 2177 , 2192 , 2196	1964 , 1967 , 1969 , 2002 , 2005 , 2007 ,
\l__xtable_tmpa_int 121 , 558 ,	2160 , 2163 , 2166 , 2187 , 2193 , 2196
566 , 575 , 588 , 590 , 591 , 603 , 605 ,	\l__xtable_tmpb_int
606 , 617 , 618 , 620 , 631 , 635 , 636 ,	... 121 , 561 , 567 , 575 , 748 ,
717 , 739 , 746 , 750 , 913 , 919 , 935 ,	750 , 910 , 913 , 914 , 928 , 931 , 932 ,
943 , 946 , 954 , 983 , 986 , 1065 , 1070 ,	957 , 975 , 976 , 983 , 984 , 1568 , 1570 ,
1110 , 1114 , 1115 , 1134 , 1135 , 1249 ,	1983 , 1988 , 1990 , 2028 , 2029 , 2030
1251 , 1360 , 1361 , 1362 , 1483 , 1484 ,	\l__xtable_tmpb_prop
	... 129 , 512 , 519 , 525 , 529 , 544

- `\l__xtable_tmpb_seq` [129](#)
- `\l__xtable_tmpb_str` [125](#), [539](#), [543](#), [547](#)
- `\l__xtable_tmpb_tl` [125](#), [629](#),
[633](#), [635](#), [924](#), [925](#), [942](#), [943](#), [966](#),
[968](#), [1059](#), [1061](#), [1580](#), [1581](#), [1583](#),
[1802](#), [1803](#), [1813](#), [1937](#), [1938](#), [1944](#)
- `\l__xtable_ufill_dim` [58](#),
[59](#), [62](#), [63](#), [66](#), [69](#), [74](#), [108](#), [1314](#),
[1328](#), [1341](#), [1349](#), [1372](#), [1471](#), [1582](#),
[1610](#), [1700](#), [1827](#), [1839](#), [1965](#), [2003](#)
- `\l__xtable_wd_dim`
..... [58](#), [59](#), [65](#), [69](#), [70](#), [73](#), [76](#),
[108](#), [1316](#), [1324](#), [1343](#), [1347](#), [1370](#),
[1377](#), [1401](#), [1405](#), [1410](#), [1428](#), [1431](#),
[1434](#), [1446](#), [1450](#), [1452](#), [1454](#), [1569](#),
[1606](#), [1614](#), [1640](#), [1644](#), [1647](#), [1651](#),
[1689](#), [1710](#), [1712](#), [1717](#), [1719](#), [1724](#),
[1726](#), [1728](#), [1730](#), [1732](#), [1734](#), [1820](#),
[1831](#), [1843](#), [1871](#), [1880](#), [1943](#), [2035](#)
- `\l__xtable_x_dim` [77–79](#), [1763](#), [1976](#),
[1987](#), [1989](#), [2014](#), [2026](#), [2032](#), [2035](#),
[2154](#), [2162](#), [2165](#), [2172](#), [2192](#), [2195](#)
- `\l__xtable_y_dim` [77–](#)
[79](#), [1763](#), [1967](#), [1973](#), [1985](#), [1991](#),
[1992](#), [2005](#), [2011](#), [2018](#), [2024](#), [2037](#),
[2158](#), [2163](#), [2165](#), [2182](#), [2193](#), [2195](#)
- `xtable/input` [882](#)
- `xtable/package` [53](#)
- `xtable/package/cellsep` [9](#)
- `xtable/package/en` [9](#)
- `xtable/package/lineskip` [9](#)
- `xtable/package/margin` [9](#)
- `xtable/package/minwidth` [9](#)
- `xtable/package/vspace` [9](#)
- `xtable/render` [2220](#), [2238](#)

To do...

- 1 (p. 3): **Debug** 修复由于测量单元格自然尺寸导致的脚注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. 6): **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. 7): **功能** 添加合并单元格功能
- 9 (p. 9): **优化** 更新边框命令及说明
- 10 (p. 9): **优化**更新 `vspace` 的实现与说明
- 11 (p. 28): **Debug** 修复表格保存命令报错的 Bug
- 12 (p. 41): **优化** 显示表格的更多内容信息
- 13 (p. 42): **Debug** 修复 `\logtable` 实际调用成 `show` 函数的 Bug
- 14 (p. 46): **Debug** 修复 CSV 正则表达式与代码不匹配的 Bug
- 15 (p. 60): **Debug** 修复宽度样式为填充时长宽计算异常的 Bug
- 16 (p. 64): **Debug** 修复多行顶对齐时，相同行高居中对齐失效的 Bug
- 17 (p. 80): **功能** 处理无表头时的渲染情况
- 18 (p. 80): **功能** 三线表标题自动合并
- 19 (p. 83): **优化** 提供内外边框的设置接口
- 20 (p. 83): **优化** 适应合并单元格的框架

- 21 (p. [86](#)): **功能** 添加并排输出功能
- 22 (p. [87](#)): **Debug** 查找 Package multicol: Error saving partial page. 的原因