

工程 3.0 格式说明

- [1 文档约定](#)
- [2 行数据](#)
 - [2.1 外层数据](#)
 - [2.2 内层数据](#)
- [3 文档块](#)
 - [3.1 文档头](#)
- [4 通用枚举值](#)
 - [4.1 EStrokeStyle \(线型\)](#)
 - [4.2 EPanelTransScope \(透明度控制范围\)](#)
 - [4.3 EAlign \(对齐模式\)](#)
 - [4.4 ESchFillStyle \(填充样式\)](#)
 - [4.5 EPadConnect \(焊盘-连接方式\)](#)
- [5 SCH_PAGE \(原理图页\)、SYMBOL \(符号\)、SIMULATION \(仿真格式\)](#)
 - [5.1 通用格式](#)
 - [5.1.1 META \(基本信息\)](#)
 - [5.1.2 CANVAS \(画布配置\)](#)
 - [5.2 基础图元格式](#)
 - [5.2.1 PART \(部件\)](#)
 - [5.2.2 GROUP \(分组控制\)](#)
 - [5.2.3 LINE \(线段\)](#)
 - [5.2.4 WIRE \(导线\)](#)
 - [5.2.5 BUS \(总线\)](#)
 - [5.2.6 BEZIER \(三阶贝塞尔线条\)](#)
 - [5.2.7 ATTR \(属性\)](#)
 - [5.2.8 TEXT \(文本\)](#)
 - [5.2.9 POLY \(多边形\)](#)
 - [5.2.10 CIRCLE \(圆\)](#)
 - [5.2.11 ARC \(圆弧\)](#)
 - [5.2.12 RECT \(矩形\)](#)
 - [5.2.13 PIN \(标号\)](#)
 - [5.2.14 OBJ \(二进制内嵌对象\)](#)
 - [5.2.15 COMPONENT \(元件\)](#)
 - [5.2.16 ELLIPSE \(椭圆\)](#)
 - [5.2.17 TABLE \(表格\)](#)
 - [5.2.18 MASK_REGION \(屏蔽区域\)](#)
 - [5.2.19 NG_SETTING \(仿真设置\)](#)
- [6 PANEL \(面板\)、PANEL_LIB \(面板库格式\)](#)
 - [6.1 通用格式](#)
 - [6.1.1 META \(基本信息\)](#)
 - [6.1.2 CANVAS \(画布配置\)](#)
 - [6.2 基础图元格式](#)
 - [6.2.1 GROUP \(分组控制\)](#)
 - [6.2.2 DIMENSION \(尺寸工具集\)](#)
 - [6.2.3 AUXLINE \(辅助线\)](#)
 - [6.3 面板多边形体系](#)
 - [6.3.1 面板单多边形的定义](#)
 - [6.3.2 面板复杂多边形的定义](#)
 - [6.3.3 POLY \(多边形\)](#)
 - [6.3.4 STRING \(文字\)](#)
- [7 PCB \(版图格式\)、FOOTPRINT \(封装格式\)](#)
 - [7.1 通用格式](#)
 - [7.1.1 META \(基本信息\)](#)
 - [7.1.2 CANVAS \(画布配置\)](#)
 - [7.1.3 LAYER \(层配置\)](#)
 - [7.1.4 LAYER_PHYS \(层物理特性配置\)](#)
 - [7.1.5 ACTIVE_LAYER \(配置当前激活层\)](#)
 - [7.2 基础图元格式](#)
 - [7.2.1 NET \(配置网络信息\)](#)
 - [7.2.2 PRIMITIVE \(图元配置\)](#)
 - [7.2.3 GROUP \(分组配置\)](#)
 - [7.2.4 SILK_OPTS \(丝印配置\)](#)
 - [7.2.5 PREFERENCE \(偏好\)](#)
 - [7.2.6 VIA \(过孔\)](#)
 - [7.2.7 PAD \(焊盘\)](#)
 - [7.2.8 LINE \(直线\)](#)
 - [7.2.9 ARC \(圆弧线\)](#)
 - [7.2.10 OBJ \(二进制内嵌对象\)](#)
 - [7.2.11 EQLEN_GRP \(等长组\)](#)
 - [7.3 多边形体系](#)
 - [7.3.1 PCB 单多边形的定义](#)
 - [7.3.2 PCB \(复杂多边形的定义\)](#)
 - [7.3.3 PARTITION \(分区\)](#)
 - [7.3.4 BOARD \(板子\)](#)
 - [7.3.5 POLY \(折线\)](#)
 - [7.3.6 FILL \(填充\)](#)
 - [7.3.7 LAYER_FILL \(层填充\)](#)
 - [7.3.8 REGION \(区域\)](#)
 - [7.3.9 POUR \(覆铜边框\)](#)

- [7.3.10 POURED \(覆铜结果\)](#)
 - [7.3.11 IMAGE \(图片\)](#)
 - [7.3.12 TEARDROP \(泪滴\)](#)
 - [7.3.13 FPC_FILL \(柔性工艺补强板\)](#)
 - [7.4 3D 外壳体系](#)
 - [7.4.1 SHELL \(外壳\)](#)
 - [7.4.2 CREASE \(侧面基准线\(折痕\)\)](#)
 - [7.4.3 SHELL_ENTITY \(外壳实体区域\)](#)
 - [7.4.4 BOSS \(螺丝柱\)](#)
 - [7.5 文字体系](#)
 - [7.5.1 STRING \(文字\)](#)
 - [7.5.2 ATTR \(属性\)](#)
 - [7.5.3 DIMENSION \(尺寸工具集\)](#)
 - [7.6 拼板](#)
 - [7.6.1 PANELIZE \(拼板\)](#)
 - [7.7 封装体系](#)
 - [7.7.1 COMPONENT \(器件实例\)](#)
 - [7.7.2 PAD_NET \(焊盘实例网络映射\)](#)
 - [7.8 设计规则体系](#)
 - [7.8.1 RULE_TEMPLATE \(设计规则模板\)](#)
 - [7.8.2 RULE \(设计规则\)](#)
 - [7.8.3 RULE_SELECTOR \(规则选择器\)](#)
 - [8 CONFIG \(工程设置格式\)](#)
 - [8.1 META \(基本信息\)](#)
 - [8.2 UNIVERSAL \(通用设置\)](#)
 - [8.2.1 ERelevanceDisplayRowType \(原理图 - 显示方式\)](#)
 - [8.2.2 ERelevanceBelongSchPage \(原理图 - 所在图页\)](#)
 - [8.2.3 ERelevanceLocation \(原理图 - 所在位置\)](#)
 - [9 SCH \(原理图格式\)](#)
 - [9.1 META \(基本信息\)](#)
 - [9.2 原理图设计规则](#)
 - [9.3 DIFFERENTIAL_PAIR \(差分对\)](#)
 - [9.4 NET_CLASS \(网络类\)](#)
 - [9.5 EOL_NET_GRP \(等长网络组\)](#)
 - [10 DEVICE \(器件格式\)](#)
 - [10.1 META \(基本信息\)](#)
 - [11 BOARD \(板子格式\)](#)
 - [11.1 META \(基本信息\)](#)
 - [12 BLOB \(二进制数据格式\)](#)
 - [12.1 BLOB \(二进制数据\)](#)
 - [13 FONT \(字体缓存格式\)](#)
 - [13.1 FONT \(字体路径\)](#)
 - [14 INSTANCE \(实例属性格式\)](#)
 - [14.1 文档头 id](#)
 - [14.2 INSTANCE_ATTR \(属性覆盖\)](#)
 - [15 VARIANT \(变体格式\)](#)
 - [15.1 META \(基本信息\)](#)
 - [15.2 VARIANT_GROUPED \(未归组\)](#)
 - [15.3 META_Z_INDEX \(排序信息\)](#)
 - [16 COMPONENT_GROUP \(元件分组格式\)](#)
 - [16.1 META \(基本信息\)](#)
 - [16.2 META_Z_INDEX \(排序信息\)](#)
 - [16.3 GROUP_INDEX \(元件分组在变体的排序\)](#)
 - [17 GROUP_DATA \(元件分组的属性数据格式\)](#)
 - [17.1 文档头 id](#)
 - [17.2 GROUP_DATA \(属性数据\)](#)

1 文档约定

- 所具有范围特性的图元，以及文件本身，都称作 `块级图元`
- 工程的所有数据在一个文件里，首行必须是 `文档头`(`DOCHEAD` 块级图元)，用于分块
- 以行为单位，每一行都是由两个合法的 `JSON` 对象拼接而成，第一个对象是用于最终一致性框架解析使用，后一个是 `原子结构对象`
- `JSON` 对象的键名采用驼峰命名，每个单词的首字母大写（除了第一个单词的首字母外），并且单词之间没有下划线或其他分隔符
- 所有图元都要带上文件内的 `唯一编号`
- `旋转角度` 以逆时针方向为正，统一使用角度制
- 若无特殊说明，所有坐标、长度、大小统一使用 `0.01 inch` 为单位
- 所有颜色都使用 `"#RRGGBB"` 的方式表达，如果需要表示无颜色（完全透明），则用 `""`
- 所有样式内容（除了 `样式标识` 和 `编号`）都可以置 `null` 表示采用默认样式
- 所有使用 `是否xxxx` 描述的属性，都使用布尔类型表示
- 几乎所有图元都要带上文件内的 `唯一编号`
- 几乎所有图元都带有 `锁定` 参数，对于已经锁定的图元，编辑器里应表现出如下行为特征
 1. 不能拖拽
 2. 不能删除
 3. 不能通过鼠标键盘调整大小

4. 不能通过鼠标键盘调整形状

2 行数据

整个数据可当做一个大字符串，可根据分隔符 `\n` 分割为一行一行数据，数据是按行按顺序一行一行解析。如：

```
第一行数据
第二行数据
第三行数据
第四行数据
```

每一行的数据，根据分隔符 `||` 可将一行数据分割成内外两层 JSON 对象数据。如：

```
外层数据 || 内层数据
```

2.1 外层数据

外层数据的类型定义是统一的，每一行数据的外层结构都一样，后续的具体图元类型时不再一一解析外层数据：

是一个 JSON 对象，用于最终一致性框架使用，保证数据的一致性，该对象按键名定义的类型如下表：

键名	值类型	是否必需	描述
type	string	是	类型名：用于识别该行属于哪种图元，每一种图元都有对应字符串
id	string	是	唯一编号：用于区分同类型下的不同数据，需保证在文档内唯一，只要 id 一样就表示同一数据
ticket	number	是	逻辑时钟：用于保证数据的唯一性；如果相同 id 的两条数据，通过对比该值保留值大的一条

格式中的 id 唯一编号主要是：

- 随机编号：绝大部分图元 id 是 16 位随机字符串编码；
- 类型名：有一些 id 值就是其数据数据类型名，与类型名 type 字段一致
- JSON 数组字符串：有一些特殊的图元，其 id 是类型名和其他字符串组成的数组，该数组是有含义的，会在对应图元处说明

如类型名为 ATTR、唯一编号为 68ea867bde655fd2、逻辑时钟为 2 的示例：

```
{ "type": "ATTR", "id": "68ea867bde655fd2", "ticket": 2 }
```

2.2 内层数据

也是 JSON 对象，但每一种图元都对应一个内层数据类型，具体看各图元的类型定义

3 文档块

有一行特殊的数据，其外层数据的类型名为 DOCHHEAD，称其为文档头，是一个块级图元，用于区分文档块：

文档数据的第一行必须是文档头，逐行解析时，当解析遇到文档头时，表示接下来解析的数据都属于该文档，直到遇到下一个文档头前，都属于同一块数据，如：

```
第一行 文档头
第二行 数据
第三行 数据
第四行 文档头
第五行 数据
第六行 数据
```

第一到第三行为一个文档块数据；第四到第六行为另一个文档块数据

3.1 文档头

外层数据的类型名为 DOCHHEAD，其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
docType	EHeaderType	是	文档类型：用于标识什么类型文档，固定枚举值
uuid	string	是	文档编号：工程内唯一，用于区分不同文档
client	string	是	终端标识：用于数据保存时，表示不同的用户
updateTime	number	是	更新时间：保存数据时的时间，数据的编辑时间
version	string	是	版本：一般与更新时间保持一致，当文档从另外一工程导入时有区别，表示该文档是来源于哪个版本

EHeaderType 的值及代表文档类型如下表：

值	文档类型
CONFIG	工程配置
BOARD	板子
SCH	原理图
SCH_PAGE	原理图页
PCB	PCB 页
PANEL	面板
SYMBOL	符号
FOOTPRINT	封装
DEVICE	器件
BLOB	二进制数据，图片
INSTANCE	实例属性
PANEL_LIB	面板库
FONT	字体
VARIANT	变体

值	文档类型
COMPONENT_GROUP	元件分组
COMPONENT_GROUP_DATA	元件分组属性
SIMULATION	仿真页

根据键名和具体值组成的数据格式如下：

```
{ "type": "DOCHEAD", "ticket": 1 } | { "docType": "SCH_PAGE", "client": "8e13a9703e0ad2c4", "uuid": "2e14f5d942c47f54", "updateTime": 1760597173242 }
```

外层数据：文档头的外层数据不记录唯一编号，文档头数据不参与一致性数据比较，逻辑时钟仅递增当前工程最大逻辑时钟

内层数据：文档类型是原理图页，终端标识是”8e13a9703e0ad2c4”，文档编号是”8e13a9703e0ad2c4”，更新时间是 1760597173242，版本是”1760597173242”

4 通用枚举值

格式中有一部分数据的值类型是枚举值，枚举值都为 大写字母加下划线； 在本文的值类型如果是以 E 字母开头的驼峰名，都表示为枚举值，一些共用的枚举值定义如下：

4.1 EStrokeStyle（线型）

值	含义
SOLID	实线
SHORT_DASH	短划线
DOT	点线
DOT_DASH	点划线

4.2 EPanelTransScope（透明度控制范围）

值	含义
PRINT	打印
COVER	遮盖
PRINT_AND_COVER	打印 + 遮盖

4.3 EAlign（对齐模式）

值	含义
LEFT_BOTTOM	左底
CENTER_BOTTOM	中底
RIGHT_BOTTOM	右底
LEFT_MIDDLE	左中
CENTER_MIDDLE	中中
RIGHT_MIDDLE	右中
LEFT_TOP	左顶
CENTER_TOP	中顶
RIGHT_TOP	右顶

4.4 ESchFillStyle（填充样式）

值	含义
NONE	无填充
SOLID	实心填充
GRID	网格
HORIZONTAL_LINE	横线
VERTICAL_LINE	竖线
RHOMBIC	菱形网格
LEFT_SLASH_LINE	左斜线
RIGHT_SLASH_LINE	右斜线

4.5 EPadConnect（焊盘-连接方式）

值	含义
DIVERGENCE	热焊（发散）
DIRECT	直连
NON_CONNECT	无连接

5 SCH_PAGE（原理图页）、SYMBOL（符号）、SIMULATION（仿真格式）

原理图页、符号页和仿真页的图元属性目前是共用的，只有个别地方有差异，有差异地方特殊会标明

5.1 通用格式

5.1.1 META（基本信息）

不同的文档类型对应的基本信息不一样，其外层数据类型名都为 `META`

5.1.1.1 原理图页

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
title	string	是	文档名称
source	string	是	来源的 <code>uuid</code> (工程库独有)
zIndex	number	是	排序大小
schematic	string	是	所属原理图

5.1.1.2 符号

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
title	string	是	名称
description	string	是	库描述
tags	string[]	是	自定义分类
source	string	是	来源的 <code>uuid</code> (工程库独有)
docType	number	是	库类型

5.1.1.3 仿真

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
title	string	是	文档名称
source	string	是	来源的 <code>uuid</code> (工程库独有)
zIndex	number	是	排序大小
docType	number	是	文档类型

5.1.2 CANVAS（画布配置）

画布附加信息，目前仅记录了原点位置，其外层数据类型名为 `CANVAS`。其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
originX	number	是	画布原点 <code>X</code> 坐标
originY	number	是	画布原点 <code>Y</code> 坐标

根据键名和具体值组成的数据格式如下：

```
{ "type": "CANVAS", "id": "CANVAS", "ticket": 1 } || { "originX": 0, "originY": 0 }
```

5.2 基础图元格式

5.2.1 PART（部件）

只有符号页才有，如果是单 `Part` 符号也必须有一个 `PART`；符号页的所有图元带上对应的部件编号，表示归属于该部件；

其外层数据类型名为 `PART`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
title	string	是	部件名称

根据键名和具体值组成的数据格式如下：

```
{ "type": "PART", "id": "partId", "ticket": 1 } || { "title": "part.1" }
```

5.2.2 GROUP（分组控制）

用于将多个图元归属为一组，其外层数据类型名为 `GROUP`。其内层数据按键名定义的类型如下表

键名	值类型	是否必需	描述
parentId	string	否	父级分组编号，为空则表示无父级
title	string	是	分组名称，无名称为空字符串 ""

根据键名和具体值组成的数据格式如下：

```
{ "type": "GROUP", "id": "UUID", "ticket": 1 } || { "groupId": "1", "parentId": "0", "title": "Logo" }
```

5.2.3 LINE（线段）

其外层数据类型名为 `LINE`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
lineGroup	string	是	所属的导线或总线组 <code>id</code>
startX	number	是	起点坐标 <code>X</code>
startY	number	是	起点坐标 <code>Y</code>
endX	number	是	结束坐标 <code>X</code>
endY	number	是	结束坐标 <code>Y</code>
strokeColor	string null	是	颜色， <code>null</code> 为默认
strokeStyle	EStrokeStyle null	是	样式， <code>null</code> 为默认，枚举值查看 EStrokeStyle 线型
fillColor	string null	是	填充颜色，"" 不填充，填充自动闭合起始点和结束点， <code>null</code> 为默认
strokeWidth	number null	是	宽度， <code>null</code> 为默认

键名	值类型	是否必需	描述
fillStyle	ESchFillStyle null	是	填充样式, null 为默认, 枚举值查看 ESchFillStyle 填充样式

根据键名和具体值组成的数据格式如下:

```
{ "type": "LINE", "ticket": 34, "id": "8c3ea2cd24930792" } | { "fillColor": null, "fillStyle": null, "strokeColor": null, "strokeStyle": nu
```

5.2.4 WIRE（导线）

原理图页和仿真页才有该图元：导线由线段组成，线段上记录导线编号；导线必须带上 **NET** 属性标识网络名称：

其外层数据类型名为 **WIRE**；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
groupId	string	是	分组控制的编号，没有默认为空；标识属于哪个组
locked	boolean	是	是否锁定
zIndex	number	是	Z 轴高度，用于层级显示，值越大的图元显示在最上层

根据键名和具体值组成的数据格式如下:

```
{ "type": "WIRE", "ticket": 33, "id": "886b587e4b874a39" } | { "groupId": "", "locked": false, "zIndex": 1 }
```

5.2.5 BUS（总线）

原理图页和仿真页才有该图元：线段上记录总线编号；总线必须带上 **NET** 属性标识网络名称：

其外层数据类型名为 **WIRE**；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
groupId	string	是	分组控制的编号，没有默认为空；标识属于哪个组
locked	boolean	是	是否锁定
zIndex	number	是	Z 轴高度，用于层级显示，值越大的图元显示在最上层
busEntry	{ [id: string]: TSchBusEntry }	是	以字符串作为键（键代表对象的唯一标识 id），值类型为 TSchBusEntry 的字典对象。

5.2.5.1 TSchBusEntry（总线接入标识）

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
groupId	string	是	分组控制的编号，没有默认为空；标识属于哪个组
locked	boolean	是	是否锁定
zIndex	number	是	Z 轴高度，用于层级显示，值越大的图元显示在最上层
order	number	是	顺序编号；在隶属的 BUS 里的顺序编号，可重复
pointX	number	是	端点 X 坐标
pointY	number	是	端点 Y 坐标
rotation	number	是	旋转角度（角度制）：绕 端点 旋转

根据键名和具体值组成的数据格式如下:

```
{ "type": "BUS", "ticket": 40, "id": "460e21b56f114e2e" } | { "groupId": "", "locked": false, "busEntry": { "46d0fd3a05f8796d": { "order": 0,
```

5.2.6 BEZIER（三阶贝塞尔线条）

其外层数据类型名为 **BEZIER**；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partId	string	否	部件编号；符号页专属，表示归属于某部件下，非符号忽略该字段
groupId	string	是	分组编号，没有则为空
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度, null 为默认
controls	number[]	是	控制点
strokeColor	string null	是	颜色, null 为默认
strokeStyle	EStrokeStyle null	是	样式, null 为默认, 枚举值查看 EStrokeStyle 线型
fillColor	string null	是	填充颜色, "" 不填充, 填充自动闭合起始点和结束点, null 为默认
strokeWidth	number null	是	宽度, null 为默认
fillStyle	ESchFillStyle null	是	填充样式, null 为默认, 枚举值查看 ESchFillStyle 填充样式

根据键名和具体值组成的数据格式如下:

```
{ "type": "BEZIER", "ticket": 49, "id": "8970ea24bc33130f" } | { "fillColor": null, "fillStyle": null, "strokeColor": null, "strokeWidth":
```

5.2.7 ATTR（属性）

ATTR 是一个较为通用的图元，其含义为

1. 表达一个由 键(KEY)—值(VALUE) 组成的的多个属性中的一个
2. 可以在画布上显示，并控制显示哪些内容，以及样式位置等

当隶属编号没有指定时，则默认隶属于当前块级图元上

其外层数据类型名为 **ATTR**；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partId	string	否	部件编号，符号页专属，原理图忽略该字段；用于标识该图元属于符号的哪个部件
groupId	string	是	分组控制的编号，没有默认为空；标识属于哪个组
locked	boolean	是	是否锁定
zIndex	number	是	Z 轴高度，用于层级显示，值越大的图元显示在最上层

键名	值类型	是否必需	描述
parentId	string	是	隶属编号，表示隶属于哪个图元，"" 表示属于当前块
key	string	是	属性 Key，属性键名
value	string	是	属性 Value，属性键值
keyVisible	boolean null	是	是否显示 Key 值，null 表示采用默认
valueVisible	boolean null	是	是否显示 Value 值，null 表示采用默认
positionX	number null	是	位置 X 坐标，未显示过的属性位置固定为 null
positionY	number null	是	位置 Y 坐标，未显示过的属性位置固定为 null
rotation	number	是	旋转角度，绕 位置 旋转
color	string null	是	颜色，null 表示采用默认
fillColor	string null	是	背景色，null 表示采用默认
fontFamily	string null	是	字体名称，null 表示采用默认
fontSize	number null	是	字体大小，与坐标等单位相同，null 表示采用默认
strikeout	boolean null	是	是否加删除线，null 表示采用默认
underline	boolean null	是	是否加下划线，null 表示采用默认
italic	boolean null	是	是否斜体，null 表示采用默认
fontWeight	boolean null	是	是否加粗，null 表示采用默认
align	EAlign	是	对齐模式，枚举值查看 EAlign 对齐模式
version	2.0	否	值为 2.0 表示从 2.2 版本转换过来的数据，因为 2.2 和 3.0 的文本原点解析不一致

根据键名和具体值组成的数据格式如下：

```
{"type":"ATTR","ticket":2,"id":"68ea867bde655fd2"}|{|"x":2547,"y":159,"rotation":0,"color":null,"fontFamily":null,"fontSize
```

5.2.8 TEXT（文本）

其外层数据类型名为 TEXT；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partId	string	否	部件编号；符号页专属，表示归属于某部件下，非符号忽略该字段
groupId	string	是	分组编号，没有则为空
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
x	number	是	文本坐标 X
y	number	是	文本坐标 Y
rotation	number	是	旋转角度（角度制），绕文本本坐标旋转
value	string	是	文本内容；任意字符
version	'2.0'	否	表示来自 2.0 版本
color	string null	是	颜色，null 为默认
fillColor	string null	是	背景色，null 为默认
fontFamily	string null	是	字体名称，null 为默认
fontSize	number null	是	字体大小，与坐标等单位相同，null 为默认
strikeout	boolean null	是	是否加删除线，null 为默认
underline	boolean null	是	是否加下划线，null 为默认
italic	boolean null	是	是否斜体，null 为默认
fontWeight	boolean null	是	是否加粗，null 为默认
align	EAlign	是	对齐模式，枚举值查看 EAlign 对齐模式

根据键名和具体值组成的数据格式如下：

```
{"type":"TEXT","ticket":50,"id":"046489ed9e632b77"}|{|"x":780,"y":-320,"rotation":0,"color":null,"fontFamily":null,"fontSiz
```

5.2.9 POLY（多边形）

其外层数据类型名为 POLY；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partId	string	否	部件编号；符号页专属，表示归属于某部件下，非符号忽略该字段
groupId	string	是	分组编号，没有则为空
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
points	TDot[]	是	点集坐标
closed	boolean	是	是否自动闭合：如果自动闭合，则结束点会自动连上起始点
strokeColor	string null	是	颜色，null 为默认
strokeStyle	EStrokeStyle null	是	样式，null 为默认，枚举值查看 EStrokeStyle 线型
fillColor	string null	是	填充颜色："" 不填充，填充自动闭合起始点和结束点，null 为默认
strokeWidth	number null	是	宽度，null 为默认
fillStyle	ESchFillStyle null	是	填充样式，null 为默认，枚举值查看 ESchFillStyle 填充样式

根据键名和具体值组成的数据格式如下：

```
{ "type": "POLY", "id": "UUID", "ticket": 1 }|{|"partId": "", "groupId": 0, "locked": false, "zIndex": 7.35, "points": [390, 26
```

5.2.10 CIRCLE（圆）

其外层数据类型名为 CIRCLE；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partId	string	否	部件编号；符号页专属，表示归属于某部件下，非符号忽略该字段
groupId	string	是	分组编号，没有则为空

键名	值类型	是否必需	描述
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度, null 为默认
centerX	number	是	圆心 X
centerY	number	是	圆心 Y
radius	number	是	半径 r
strokeColor	string null	是	颜色, null 为默认
strokeStyle	EStrokeStyle null	是	样式, null 为默认, 枚举值查看 EStrokeStyle 线型
fillColor	string null	是	填充颜色: "" 不填充, 填充自动闭合起始点和结束点, null 为默认
strokeWidth	number null	是	宽度, null 为默认
fillStyle	ESchFillStyle null	是	填充样式, null 为默认, 枚举值查看 ESchFillStyle 填充样式

根据键名和具体值组成的数据格式如下：

```
{"type":"CIRCLE","ticket":28,"id":"cda155f9448228df"}|{"centerX":-40,"centerY":10,"radius":10,"strokeColor":null,"strokeSt
```

5.2.11 ARC（圆弧）

其外层数据类型名为 `ARC`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partId	string	否	部件编号：符号页专属，表示归属于某部件下，非符号忽略该字段
groupId	string	是	分组编号，没有则为空
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度, null 为默认
startX	number	是	起始 X
startY	number	是	起始 Y
referX	number	是	参考 X
referY	number	是	参考 Y
endX	number	是	结束 X
endY	number	是	结束 Y
strokeColor	string null	是	颜色, null 为默认
strokeStyle	EStrokeStyle null	是	样式, null 为默认, 枚举值查看 EStrokeStyle 线型
fillColor	string null	是	填充颜色: "" 不填充, 填充自动闭合起始点和结束点, null 为默认
strokeWidth	number null	是	宽度, null 为默认
fillStyle	ESchFillStyle null	是	填充样式, null 为默认, 枚举值查看 ESchFillStyle 填充样式

根据键名和具体值组成的数据格式如下：

```
{"type":"ARC","ticket":27,"id":"ie24"}|{"startX":40,"startY":10,"endX":35,"endY":5,"referX":38.53553,"referY":6.46447,"str
```

5.2.12 RECT（矩形）

矩形由其对角的两个点定义，其旋转是绕 `点1` 进行的：

其外层数据类型名为 `RECT`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partId	string	否	部件编号：符号页专属，表示归属于某部件下，非符号忽略该字段
groupId	string	是	分组编号，没有则为空
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度, null 为默认
dotX1	number	是	点 1 X
dotY1	number	是	点 1 Y
dotX2	number	是	点 2 X
dotY2	number	是	点 2 Y
radiusX	number	是	圆角半径 X: 0 表示非圆角
radiusY	number	是	圆角半径 Y: 0 表示非圆角
rotation	number	是	旋转角度（角度制），绕点 1 旋转
strokeColor	string null	是	颜色, null 为默认
strokeStyle	EStrokeStyle null	是	样式, null 为默认, 枚举值查看 EStrokeStyle 线型
fillColor	string null	是	填充颜色: "" 不填充, 填充自动闭合起始点和结束点, null 为默认
strokeWidth	number null	是	宽度, null 为默认
fillStyle	ESchFillStyle null	是	填充样式, null 为默认, null 为默认, 枚举值查看 ESchFillStyle 填充样式

根据键名和具体值组成的数据格式如下：

```
{"type":"RECT","ticket":29,"id":"faca7a9870d002dc"}|{"radiusX":0,"radiusY":0,"dotX1":10,"dotX2":30,"dotY1":20,"dotY2":30,"
```

5.2.13 PIN（标号）

PIN 必须具有 **NAME** 属性和 **NUMBER** 属性； 其外层数据类型名为 `PIN`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partId	string	否	部件编号：符号页专属，表示归属于某部件下，非符号忽略该字段
groupId	string	是	分组编号，没有则为空
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度, null 为默认
display	boolean	是	是否在原理图显示
x	number	是	位置 X
y	number	是	位置 Y

键名	值类型	是否必需	描述
length	number	是	引脚长度
rotation	number	是	旋转角度（角度制）： 0 90 180 270
color	string	是	引脚颜色
pinShape	EPinShape	是	引脚形状枚举值

EPinShape 引脚形状枚举值如下：

值	含义
NONE	无
CLOCK	时钟
INVERTED	反相
INVERTED_CLOCK	反相时钟

根据键名和具体值组成的数据格式如下：

```
{ "type": "PIN", "ticket": 30, "id": "66ed739cbe85b256" } || { "display": true, "x": -10, "y": 30, "length": 10, "rotation": 180, "color": null,
```

5.2.14 OBJ（二进制内嵌对象）

内嵌于图页上的图片和文件等数据，可作为附件下载，以及直接显示（EDA 自行决定，不在格式内要求）

其外层数据类型名为 OBJ；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partId	string	否	部件编号：符号页专属，表示归属于某部件下，非符号忽略该字段
groupId	string	是	分组编号，没有则为空
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高，null 为默认度
fileName	string	是	文件名
startX	number	是	左上角 X
startY	number	是	左上角 Y
width	number	是	宽
height	number	是	高
rotation	number	是	旋转角度（角度制）：绕左上角旋转
isMirror	boolean	是	是否镜像
content	string	是	二进制数据，有两种模式一般格式，1.遵循 Data Urls 规范 data:[<mediatype>][;<base64>,<data> 2.BLOB 引用模式，blob:hashid

根据键名和具体值组成的数据格式如下：

```
{ "type": "OBJ", "ticket": 34, "id": "e9fef51ae02ee614" } || { "startX": 70, "startY": 20, "width": 174, "height": 177, "rotation": 0, "isMirror
```

5.2.15 COMPONENT（元件）

其外层数据类型名为 COMPONENT；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
groupId	string	是	分组编号，没有则为空
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
partId	string	是	子库编号：默认 ""
x	number	是	位置 X
y	number	是	位置 Y
rotation	number	是	旋转角度（角度制）：绕位置旋转
isMirror	boolean	是	是否镜像
attrs	object	是	自定义属性，是一个动态键值对集合，其键是字符串，值可以是任意类型(目前基本是字符串或者数)

- COMPONENT 引用了 Symbol，Symbol 支持多 PART，所以带了 部件编号 属性指示具体哪一个
- COMPONENT 下可绑定许多 ATTR，具体的属性行为将由工具定义

Component 所引用的 Symbol 图元一定是按照如下顺序执行的变换

- 按照 旋转角度 绕原点 (0,0) 逆时针旋转
- 如果 是否镜像 为 1，则绕原点 (0,0) 所在的 Y 轴进行水平镜像
- 根据 位置 进行平移

或者可以理解成如下等价的变换（但是实现更繁琐一些）

- 根据 位置 进行平移
- 按照 旋转角度 绕 位置 逆时针旋转
- 如果 是否镜像 为 1，则绕 位置 所在的 Y 轴进行水平镜像

根据键名和具体值组成的数据格式如下：

```
{ "type": "COMPONENT", "ticket": 51, "id": "548cc7e4efcc835c" } || { "partId": "pid8a0e77bacb214e", "x": 295, "y": -495, "rotation": 0, "isMi
```

5.2.16 ELLIPSE（椭圆）

其外层数据类型名为 ELLIPSE；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partId	string	否	部件编号：符号页专属，表示归属于某部件下，非符号忽略该字段
groupId	string	是	分组编号，没有则为空
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认

键名	值类型	是否必需	描述
centerX	number	是	中点 centerX
centerY	number	是	中点 centerY
radiusX	number	是	水平半径 radiusX
radiusY	number	是	垂直半径 radiusY
rotation	number	是	旋转角度（角度制）
strokeColor	string null	是	颜色，null 为默认
strokeStyle	EStrokeStyle null	是	样式，null 为默认，枚举值查看 EStrokeStyle 线型
fillColor	string null	是	填充颜色：“” 不填充，填充自动闭合起始点和结束点，null 为默认
strokeWidth	number null	是	宽度，null 为默认
fillStyle	ESchFillStyle null	是	填充样式，null 为默认，枚举值查看 ESchFillStyle 填充样式

根据键名和具体值组成的数据格式如下：

```
{ "type": "ELLIPSE", "ticket": 35, "id": "cc33d553b34ee38e" } || { "radiusX": 30, "radiusY": 20, "centerX": -170, "centerY": 30, "strokeColor":
```

5.2.17 TABLE（表格）

其外层数据类型名为 TABLE；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
groupId	string	是	分组编号，没有则为空
zIndex	number null	是	Z 轴高度，null 为默认
startX	number	是	左上角 X
startY	number	是	左上角 Y
rowSizes	number[]	是	行高
colSizes	number[]	是	列宽
rowLocked	number[]	是	行锁定
colLocked	number[]	是	列锁定
rotation	number	是	旋转角度（角度制）
tableCell	TTableCell[]	是	表格单元格

5.2.17.1 TTableCell（表格单元格）

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
id	string	是	唯一编号
value	string	是	内容
rowIndex	number	是	行
columnIndex	number	是	列
rowSpan	number	是	宽度（占多少列）
colSpan	number	是	高度（占多少行）
topStyle	TLineStyle	是	边框线形样式（上）
rightStyle	TLineStyle	是	边框线形样式（右）
bottomStyle	TLineStyle	是	边框线形样式（下）
leftStyle	TLineStyle	是	边框线形样式（左）
fontStyle	TTableFontStyle	是	字体样式
lineHeight	number	是	行间距
zIndex	number null	是	Z 轴高度，null 为默认

TLineStyle 按键名定义的类型如下表：

键名	值类型	是否必需	描述
strokeColor	string null	是	颜色，null 为默认
strokeStyle	EStrokeStyle null	是	样式，null 为默认，枚举值查看 EStrokeStyle 线型
fillColor	string null	是	填充颜色：“” 不填充，填充自动闭合起始点和结束点，null 为默认
strokeWidth	number null	是	宽度，null 为默认
fillStyle	ESchFillStyle null	是	填充样式，null 为默认，枚举值查看 ESchFillStyle 填充样式

TTableFontStyle 按键名定义的类型如下表：

键名	值类型	是否必需	描述
color	string null	是	颜色，null 为默认
fillColor	string null	是	背景色，null 为默认
fontFamily	string null	是	字体名称，null 为默认
fontSize	number null	是	字体大小，与坐标等单位相同，null 为默认
strikeout	boolean null	是	是否加删除线，null 为默认
underline	boolean null	是	是否加下划线，null 为默认
italic	boolean null	是	是否斜体，null 为默认
fontWeight	boolean null	是	是否加粗，null 为默认
hAlign	EHAlign	是	水平对齐
vAlign	EVAlign	是	垂直对齐

EHAlign 水平对齐模式枚举值如下：

值	含义
LEFT	左
CENTER	中
RIGHT	右

EVALign 垂直对齐模式枚举值如下：

值	含义
TOP	上
MIDDLE	中
BOTTOM	下

根据键名和具体值组成的数据格式如下：

```
{ "type": "TABLE", "ticket": 36, "id": "9a1299f11752742b" } || { "startX": -180, "startY": 130, "rotation": 0, "rowSizes": [20], "colSizes": [
```

5.2.18 MASK_REGION（屏蔽区域）

其外层数据类型名为 MASK_REGION；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
groupId	string	是	分组编号，没有则为空
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
maskExpand	boolean	是	是否屏蔽，默认是
dotX1	number	是	点 1 X
dotY1	number	是	点 1 Y
dotX2	number	是	点 2 X
dotY2	number	是	点 2 Y
expandWidth	number null	是	展开宽度，null 为默认
expandHeight	number null	是	展开高度，null 为默认
strokeColor	string null	是	颜色，null 为默认
strokeStyle	EStrokeStyle null	是	样式，null 为默认，枚举值查看 EStrokeStyle 线型
fillColor	string null	是	填充颜色；"" 不填充，填充自动闭合起始点和结束点，null 为默认
strokeWidth	number null	是	宽度，null 为默认
fillStyle	ESchFillStyle null	是	填充样式，null 为默认，枚举值查看 ESchFillStyle 填充样式

根据键名和具体值组成的数据格式如下：

```
{ "type": "MASK_REGION", "ticket": 58, "id": "3bf4aa1ecf9f1642" } || { "dotX1": 145, "dotX2": 230, "dotY1": -470, "dotY2": -400, "expandWidth":
```

5.2.19 NG_SETTING（仿真设置）

仿真页专有数据

其外层数据类型名为 NG_SETTING；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
acSetting	object	是	AC 设置
acSetting.acAnalysisType	string	是	AC 分析类型
acSetting.freMult	string	是	频率倍数
acSetting.startFrequency	string	是	起始频率
acSetting.stopFrequency	string	是	停止频率
dcSetting	object	是	DC 设置
dcSetting.dcSourceData1	TDCSourceData	是	DC 源数据 1
dcSetting.dcSourceData2	TDCSourceData	是	DC 源数据 2
dcSetting.sourceStatus	boolean	是	源状态
trSetting	object	是	瞬态分析设置
trSetting.timeStep	string	是	时间步长
trSetting.stopTime	string	是	停止时间
trSetting.startTime	string	是	起始时间
trSetting.maxTimeStep	string	是	最大时间步长
currentSetting	‘AC’ ‘DC’ ‘TR’	是	当前设置

5.2.19.1 TDCSourceData（仿真设置源数据）

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
sourceName	string	是	源名称
startValue	string	是	起始值
stopValue	string	是	停止值
increment	string	是	增量值

根据键名和具体值组成的数据格式如下：

6 PANEL（面板）、PANEL_LIB（面板库格式）

面板页和面板库共用一套图元

- 面板颜色支持三种模式
 - HEX：如 #FFFEAB，格式含义为 `#(R){G}{B}`，忽略大小写
 - RGB：如 `rgb(255,0,0)`，格式含义为 `rgb({R},{G},{B})`，忽略大小写，每个参数范围 0 ~ 255
 - CMYK：如 `cmyk(15,100,20,0)`，格式含义为 `cmyk({C},{M},{Y},{K})`，忽略大小写，每个参数范围 0 ~ 100

6.1 通用格式

6.1.1.1 META（基本信息）

不同的文档类型对应的基本信息不一样，其外层数据类型名都为 META

6.1.1.1.1 面板

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
title	string	是	文档名称
zIndex	number	是	排序大小

6.1.1.1.2 面板库

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
title	string	是	名称
description	string	是	库描述
tags	string[]	是	自定义分类

6.1.2 CANVAS（画布配置）

画布配置信息，目前已占用的一些字段

其外层数据类型名为 CANVAS；其内层数据按键名定义的类型如下表：

键名	类型	是否必需	描述
originX	number	是	画布原点 X
originY	number	是	画布原点 Y
width	string	是	宽
height	string	是	高
material	string	是	材质
thickness	string	是	厚度
print	string	是	打印方式
craft	string	是	表面工艺
desc	string	是	描述
coverColor	string	是	遮盖颜色
backgroundColor	string	是	背景颜色
orderWidth	string	是	下单板框最大宽度
orderHeight	string	是	下单板框最大高度

根据键名和具体值组成的数据格式如下：

```
{ "type": "CANVAS", "ticket": 1, "id": "CANVAS" } || { "material": "acrylic", "thickness": "0.8mm", "print": "Bottom Side", "craft": "Transp
```

6.2 基础图元格式

6.2.1 GROUP（分组控制）

其外层数据类型名为 GROUP；其内层数据按键名定义的类型如下表：

键名	类型	是否必需	描述
parentId	string	是	父级分组编号：为 0 则表示无父级
title	string	是	分组名称：可选，根据实际需求处理
visible	boolean	是	是否可见
locked	boolean	是	是否锁定

根据键名和具体值组成的数据格式如下：

```
{ "type": "GROUP", "ticket": 9, "id": "e532ef4202fe22131" } || { "parentId": "0", "title": "", "visible": true, "locked": false }
```

6.2.2 DIMENSION（尺寸工具集）

其外层数据类型名为 DIMENSION；其内层数据按键名定义的类型如下表：

键名	类型	是否必需	描述
layer	number	是	层
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
valid	boolean	是	是否生效
visible	boolean	是	是否可见
cover	number null	是	遮盖透明度，null 为默认
name	string	是	名称
dimensionType	string	是	尺寸类型
unit	string	是	单位（预留）：mm cm inch mil 或 null 为跟随画布单位
strokeWidth	number	是	线宽（预留，与画布尺寸无关）
accuracy	number	是	精度（预留）
controlDot	number[]	是	控制点 X1 Y1 X2 Y2_
relationIds	string[]	是	关联图元 Id
rotation	number	是	旋转角度

根据键名和具体值组成的数据格式如下：

```
{ "type": "DIMENSION", "ticket": 4, "id": "e119b4dd7d409ea8b" } || { "layer": 8, "dimensionType": "LENGTH-CONSTRAINT", "unit": "", "strokeW
```

6.2.3 AUXLINE（辅助线）

其外层数据类型名为 AUXLINE；其内层数据按键名定义的类型如下表：

辅助线通过 长度 属性切换点与线模式

键名	类型	是否必需	描述
groupId	string	是	分组编号，没有则为空（预留）；0 不分组，非 0 为组标志
layer	number	是	层
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
visible	boolean	是	是否可见
cover	number null	是	遮盖透明度，null 为默认
name	string	是	名称
x	number	是	X 坐标
y	number	是	Y 坐标
rotation	number	是	旋转角度（角度制）
strokeWidth	number	是	线宽（预留，非画布尺寸）
strokeStyle	EStrokeStyle null	是	线型，null 为默认，枚举值查看 EStrokeStyle 线型
color	string	是	颜色
length	number	是	长度（非画布尺寸）

根据键名和具体值组成的数据格式如下：

```
{ "type": "AUXLINE", "ticket": 4, "id": "e5406686c5e3161fe" } || { "groupId": "0", "layer": 8, "x": 368.504, "y": 1195.276, "rotation": 0, "str
```

6.3 面板多边形体系

SVG 中 path 是一个对多边形优秀的抽象。但由于 PCB 内用不到其中相对位置等功能，并且有条件设计更方便解析的方式。所以仿造 SVG 的 path 创造了一种类似的表达多边形的方式。多边形体系内 POLY REGION POUR 支持互相转换

6.3.1 面板单多边形的定义

单多边形为首尾重合的一条不间断的线所描述的区域。如果首尾不重合需要将其自动重合。

```
[300, 200, "L", 400, 200, "A", 10, 20, 0, 1, 0, 400, 220, "C", 200, 500, 400, 300, 100, 100]
```

```
["R", 100, 200, 300, 300, 0]
```

```
["ELLIPSE", 100, 200, 5, 10, 1]
```

6.3.1.1 L（直线模式）

X Y L X Y X Y ... 模式为直线模式，所有坐标将用直线将其连一一连起来

6.3.1.2 A/CA（椭圆弧模式）

startX startY A rx ry rot isLarge isCCW endX endY 模式为圆弧模式

- startX/startY: 开始坐标
- rx/ry: 水平半径、垂直半径
- rot: 旋转角度（绕圆心）
- isLarge: 是否大弧
- isCCW: 是否逆时针
- endX/endY: 结束坐标

startX startY CA rx ry rot isLarge isCCW endX endY 模式为中心圆弧模式

6.3.1.3 C（三阶贝塞尔模式）

X1 Y1 C X2 Y2 X3 Y3 X4 Y4 ... 模式为三阶贝塞尔模式，所有坐标为其控制点

6.3.1.4 R（矩形模式）

R X Y width height rx ry rot isCCW 矩形模式与其它都不兼容，是一个独立的模式

- X/Y: 左上角坐标
- width/height: 宽高
- rx/ry: 圆角
- rot: 旋转角度（绕 X/Y）
- isCCW: 是否逆时针

6.3.1.5 ELLIPSE（椭圆模式）

ELLIPSE cx cy rx ry rot isCCW 椭圆模式与其它都不兼容，是一个独立的模式

- cx/cy: 中心点坐标
- rx/ry: 水平半径、垂直半径
- rot: 旋转角度（绕 cx/cy）
- isCCW: 是否逆时针

6.3.2 面板复杂多边形的定义

```
[[单多边形1], [单多边形2]]
```

复杂多边形可以包含多个单多边形，通过 `fill-rule` 将其组合（详情参考 `SVG path`）。以实现多边形的例如 `与` `非` 的布尔运算，常用于带洞的多边形目前固定使用 `nonezero` 这个 `fill-rule`

6.3.3 POLY（多边形）

其外层数据类型名为 `POLY`；其内层数据按键名定义的类型如下表：

键名	类型	是否必需	描述
groupId	string	是	分组编号，没有则为空（预留）： <code>0</code> 不分组，非 <code>0</code> 为组标志
layer	number	是	层
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度， <code>null</code> 为默认
valid	boolean	是	是否生效
visible	boolean	是	是否可见
cover	number null	是	遮盖透明度， <code>null</code> 为默认
name	string	是	名称
rotation	number	是	控制点旋转角度（角度制）
strokeWidth	number	是	线宽
strokeStyle	EStrokeStyle null	是	线型， <code>null</code> 为默认，枚举值查看 EStrokeStyle 线型
strokeColor	string	是	描边颜色
fillColor	string	是	填充模式
opacity	number	是	不透明度： <code>0-1</code>
autoClose	boolean	是	是否自动闭合首尾端点
ploys	any[][]	是	路径，参考 面板复杂多边形的定义
displayFill	boolean	否	是否显示填充
displayStroke	boolean	否	是否显示描边
matrix	number[][] null	是	变换矩阵， <code>null</code> 为默认
transScope	EPanelTransScope	否	控制范围，枚举值查看 EPanelTransScope 透明度控制范围
transPrint	number null	否	自定义打印层不透明度：被控制范围选中，默认 <code>0.3</code> ，未选中默认 <code>0</code> ，范围 <code>0-1</code> ， <code>null</code> 为默认
transWhite	number null	否	自定义遮盖层不透明度：被控制范围选中，默认 <code>0.3</code> ，未选中默认 <code>0</code> ，范围 <code>0-1</code> ， <code>null</code> 为默认

根据键名和具体值组成的数据格式如下：

```
{ "type": "POLY", "ticket": 2, "id": "e79b342e010515f9e" } | { "groupId": "0", "layer": 1, "rotation": 0, "strokeWidth": 3.937007874015748,
```

6.3.4 STRING（文字）

其外层数据类型名为 `STRING`；其内层数据按键名定义的类型如下表：

键名	类型	是否必需	描述
groupId	string	是	分组编号，没有则为空（预留）： <code>0</code> 不分组，非 <code>0</code> 为组标志
layer	number	是	层
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度， <code>null</code> 为默认
valid	boolean	是	是否生效
visible	boolean	是	是否可见
cover	number null	是	遮盖透明度， <code>null</code> 为默认
name	string	是	名称
rotation	number	是	控制点旋转角度（角度制）
value	string	是	内容
fontFamily	string	是	字体名称
fontSize	number	是	字号
fontWeight	number	是	字体粗细
italic	boolean	是	是否斜体
underline	boolean	是	是否下划线
strikeout	boolean	是	是否删除线
align	EAlign	是	对齐模式，枚举值查看 EAlign 对齐模式
strokeColor	string	是	颜色
opacity	number	是	不透明度： <code>0-1</code>
matrix	number[][] null	是	变换矩阵， <code>null</code> 为默认
strokes	boolean	是	是否变换描边
path	any[][]	是	路径，参考 面板复杂多边形的定义
transScope	EPanelTransScope	否	控制范围，枚举值查看 EPanelTransScope 透明度控制范围
transPrint	number null	否	自定义打印层不透明度：被控制范围选中，默认 <code>0.3</code> ，未选中默认 <code>0</code> ，范围 <code>0-1</code> ， <code>null</code> 为默认
transWhite	number null	否	自定义遮盖层不透明度：被控制范围选中，默认 <code>0.3</code> ，未选中默认 <code>0</code> ，范围 <code>0-1</code> ， <code>null</code> 为默认

根据键名和具体值组成的数据格式如下：

```
{ "type": "STRING", "ticket": 3, "id": "efc5949f2664c1144" } | { "groupId": "0", "layer": 1, "rotation": 0, "value": "文字", "fontFamily": "黑
```

7 PCB（版图格式）、FOOTPRINT（封装格式）

PCB 页和封装页共用

7.1 通用格式

7.1.1 META（基本信息）

不同的文档类型对应的基本信息不一样，其外层数据类型名都为 `META`

7.1.1.1 PCB（页）

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
title	string	是	文档名称
source	string	是	来源的 uuid
zIndex	number	是	排序大小
board	string	是	所属板子
parent	string	否	父 PCB

7.1.1.2 封装

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
title	string	是	名称
description	string	是	库描述
tags	string[]	是	自定义分类
source	string	是	来源的 uuid(工程库独有)

7.1.2 CANVAS（画布配置）

其外层数据类型名为 CANVAS；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
originX	number	是	画布原点 X
originY	number	是	画布原点 Y
unit	string	是	显示单位（不会影响格式里数据的单位）
gridXSize	number	是	网格尺寸 X
gridYSize	number	是	网格尺寸 Y
snapXSize	number	是	栅格尺寸 X
snapYSize	number	是	栅格尺寸 Y
altSnapXSize	number	是	Alt 栅格尺寸 X
altSnapYSize	number	是	Alt 栅格尺寸 Y
gridType	EGridType	是	网格类型
multiGridType	EGridType	是	加粗网格类型
multiGridRatio	number	是	加粗网格倍数: number
highlightValue	number	是	高亮亮度值

EGridType 网格类型枚举值如下：

值	含义
NONE	无
GRID	网格
OUTLETS	网点

根据键名和具体值组成的数据格式如下：

```
{ "type": "CANVAS", "ticket": 1, "id": "CANVAS" } | { "originX": 0, "originY": 0, "unit": "mil", "gridXSize": 5, "gridYSize": 5, "snapXSize": 5
```

7.1.3 LAYER（层配置）

其外层数据类型名为 LAYER；外层数据的唯一编号是一个 JSON 数组字符串，数组按顺序定义的结构如：

顺序	值类型	描述
1	LAYER	固定值，作用是避免与其他唯一编号重复
2	number	层编号，固定枚举值看层定义

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
layerType	string	是	层类型，是固定枚举值
layerName	string	是	层别名，需要唯一
use	boolean	是	是否使用
show	boolean	是	是否显示
locked	boolean	是	是否锁定
activeColor	string	是	激活颜色
activateTransparency	number	是	激活透明度
inactiveColor	string	是	非激活颜色
inactiveTransparency	number	是	非激活透明度

7.1.3.1 层定义

层类型和层编号是固定枚举值，不能自定义，具体请参考如下枚举：

层编号	层类型	描述
1	TOP	顶层
2	BOTTOM	底层
3	TOP_SILK	顶层丝印
4	BOT_SILK	底层丝印
5	TOP_SOLDER_MASK	顶层助焊
6	BOT_SOLDER_MASK	底层助焊

层编号	层类型	描述
7	TOP_PASTE_MASK	顶层阻焊
8	BOT_PASTE_MASK	底层阻焊
9	TOP_ASSEMBLY	顶层装配层
10	BOT_ASSEMBLY	底层装配层
11	OUTLINE	边框
12	MULTI	多层
13	DOCUMENT	文档
14	MECHANICAL	机械层
47	HOLE	孔层
48	COMPONENT_SHAPE	元件外形层
49	COMPONENT_MARKING	元件标识层
50	PIN_SOLDERING	引脚焊接层
51	PIN_FLOATING	引脚悬空层
52	COMPONENT_MODEL	元件模型层
53	3D_SHELL_OUTLINE	3d 外壳层-边框
54	3D_SHELL_TOP	3d 外壳层-顶层
55	3D_SHELL_BOTTOM	3d 外壳层-底层
56	DRILL_DRAWING	钻孔图层
57	RATLINE	飞线层
58	TOP_STIFFENER	顶层补强
59	BOTTOM_STIFFENER	底层补强
71-100	CUSTOM	自定义层
101-131	SUBSTRATE	基板

特殊层：

有些层允许设置多个层，层 ID 一般依次递增：

- 内层（15-46）：层类型有信号层（SIGNAL）和内电层（PLANE）两种，目前最大可设置32个内层，层 ID 依次递增，即内层 1 的层 ID 为 15，内层 32 的层 ID 为 46。
- 自定义层（CUSTOM）（71-100）：目前最大允许设置30个自定义层，层 ID 依次递增，即自定义层 1 的层 ID 为 71，自定义层 30 的层 ID 为 100。
- 基板层（SUBSTRATE）（101-131）：目前最大允许设置31个基板层，层 ID 依次递增，即基板层 1 的层 ID 为 101，基板层 30 的层 ID 为 131。

还有一些不出现在文档中，PCB 内置的层，保证层 ID 不与以下层 ID 重复即可：

- DRCERROR: 1001
- Mask: 1002
- TextColor: 1003
- Partition: 1004

根据键名和具体值组成的数据格式如下：

```
{"type":"LAYER","ticket":2,"id":["\LAYER\","1"]}||{"layerId":1,"layerType":"TOP","layerName":"Top Layer","use":true,"show":
```

7.1.4 LAYER_PHYS（层物理特性配置）

其外层数据类型名为 LAYER_PHYS；外层数据的唯一编号是一个 JSON 数组字符串，数组按顺序定义的结构如：

顺序	值类型	描述
1	LAYER_PHYS	固定值，作用是避免与其他唯一编号重复
2	number	层编号，固定枚举值看层定义

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
material	string	是	层材质
thickness	number	是	厚度
permittivity	number	是	介电常数
lossTangent	number	是	损耗切线
isKeepIsland	boolean	是	内电层是否保留孤岛
zIndex	number	是	顺序大小

根据键名和具体值组成的数据格式如下：

```
{"type":"LAYER_PHYS","ticket":62,"id":["\LAYER_PHYS\","3"]}||{"material":null,"thickness":0,"permittivity":null,"lossTanger
```

7.1.5 ACTIVE_LAYER（配置当前激活层）

其外层数据类型名为 ACTIVE_LAYER；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
layerId	number	是	层编号，固定枚举值看层定义

根据键名和具体值组成的数据格式如下：

```
{"type":"LAYER_PHYS","ticket":62,"id":["\LAYER_PHYS\","3"]}||{"material":null,"thickness":0,"permittivity":null,"lossTanger
```

7.2 基础图元格式

7.2.1 NET（配置网络信息）

其外层数据类型名为 NET：外层数据的唯一编号是一个 JSON 数组字符串，数组按顺序定义的结构如：

顺序	值类型	描述
1	NET	固定值，作用是避免与其他唯一编号重复
2	string	网络名称

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
netType	string null	是	网络类型：null 为无类型
specialColor	string null	是	特殊颜色：null 为无特殊颜色
retLine	boolean	是	是否显示飞线
differentialName	string null	是	差分对名称：null 为非差分对
isPositiveNet	boolean	是	是否差分对正极
equalLengthGroupName	string null	是	等长组名称：null 为非等长组

根据键名和具体值组成的数据格式如下：

```
{"type":"NET","ticket":72,"id":["\NET\","\"]}{{"netType":null,"specialColor":null,"retLine":true,"differentialName":nul
```

7.2.2 PRIMITIVE（图元配置）

其外层数据类型名为 PRIMITIVE：外层数据的唯一编号是一个 JSON 数组字符串，数组按顺序定义的结构如：

顺序	值类型	是否必需	描述
1	PRIMITIVE	是	固定值，作用是避免与其他唯一编号重复
2	string	是	图元类型名，固定枚举值
3	number	否	层编号，参考层定义；带有该参数表示该图元配置只在对应层生效

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
display	boolean	是	是否显示
pick	boolean	否	是否可拾取
color	string	否	颜色
transparency	number	否	透明度

根据键名和具体值组成的数据格式如下：

```
{"type":"PRIMITIVE","ticket":92,"id":["\PRIMITIVE\","\ALL"]}(){{"display":true,"pick":false}
```

7.2.3 GROUP（分组配置）

其外层数据类型名为 GROUP：其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
groupName	string	是	分组名称

根据键名和具体值组成的数据格式如下：

```
{"type":"GROUP","ticket":133,"id":["1"]}(){{"groupName":"group1"}
```

7.2.4 SILK_OPTS（丝印配置）

其外层数据类型名为 SILK_OPTS：外层数据的唯一编号是一个 JSON 数组字符串，数组按顺序定义的结构如：

顺序	值类型	描述
1	SILK_OPTS	固定值，作用是避免与其他唯一编号重复
2	number	层编号，固定枚举值看层定义

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
defaultColor	string	是	默认颜色
baseColor	string	是	底色

根据键名和具体值组成的数据格式如下：

```
{"type":"SILK_OPTS","ticket":129,"id":["\SILK_OPTS\","3"]}(){{"defaultColor":"#000000","baseColor":"#FFFFFF"}
```

7.2.5 PREFERENCE（偏好）

其外层数据类型名为 PREFERENCE：其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
startTrackWidthFollowLast	boolean	是	起始布线是否跟随上次设置
lastTrackWidth	number	是	上次布线宽度
startViaSizeFollowLast	boolean	是	起始打孔尺寸是否跟随上次设置
lastViaInnerDiameter	number	是	上次打孔内径
lastViaDiameter	number	是	上次打孔外径
snap	boolean	是	是否自动吸附
routingMode	ERoutingMode	是	布线模式
routingCorner	ERoutingConner	是	布线拐角模式
removeLoop	boolean	是	布线是否自动移除回路

键名	值类型	是否必需	描述
rotatingObject	boolean	是	是否单对象旋转
trackFollow	boolean	是	导线是否跟随封装移动
stretchTrackMinCorner	number	是	拉伸导线最小拐角比率（比线宽）
preferenceConfig	string	是	层堆叠偏好来源
realTimeUpdateUnusedLayers	boolean	是	是否自动移除未使用焊盘
unusedPadRange	EUnusedPadRange	是	移除未使用焊盘的范围
pushVia	EPushViaOptimization	是	推挤过孔优化
pathOptimization48ePushed	EPathOptimization	是	被推挤导线路径优化
currentPathOptimization48ePushed	ECurrentPathOptimization	是	当前导线路径优化
removeCircuitsContainingVias	boolean	是	移除有过孔的回路
removeAntenna	boolean	是	移除天线

ERoutingMode 布线模式枚举值如下：

值	含义
NONE	无
PUSH	推挤
SURROUND	环绕
OBSTRUCT	阻挡
SURROUND_AND_PUSH	环绕并推挤

ERoutingConner 布线拐角模式枚举值如下：

值	含义
L45	线条 45 度
L90	线条 90 度
L	线条自由角度
R45	圆弧 45 度
R90	圆弧 90 度
R	圆弧自由角度

EUnusedPadRange 移除未使用焊盘的范围枚举值如下：

值	说明
ALL	所有
PAD	仅焊盘
VIA	仅过孔

EPushViaOptimization 推挤过孔优化枚举值如下：

值	含义
OPTIMIZA_OPEN	开
OPTIMIZA_NCNE	关

EPathOptimization 被推挤导线路径优化枚举值如下：

值	含义
NONE	无
SINGLE	单段
ALL	整段

ECurrentPathOptimization 当前导线路径优化枚举值如下：

值	含义
OPTIMIZA_STRONG	强
OPTIMIZA_WEAK	弱
OPTIMIZA_NCNE	关

根据键名和具体值组成的数据格式如下：

```
{ "type": "PREFERENCE", "ticket": 131, "id": "PREFERENCE" } || { "startTrackWidthFollowLast": false, "lastTrackWidth": 10, "startViaSize": 10 }
```

7.2.6 VIA（过孔）

其外层数据类型名为 VIA；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 0 不分组，非 0 为组标志，相同组标志的为一组
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
netName	string	是	NET，网络名称
ruleName	string	是	过孔层类型：设计规则名称，定义过孔的开始层结束层
centerX	number	是	坐标 X
centerY	number	是	坐标 Y
holeDiameter	number	是	孔直径
viaDiameter	number	是	焊盘直径
viaType	EViaType	是	过孔类型
topSolderExpansion	number null	是	顶层阻焊扩展：null 为遵循规则
bottomSolderExpansion	number null	是	底层阻焊扩展：null 为遵循规则

键名	值类型	是否必需	描述
unusedInnerLayers	number[]	是	隐藏焊盘层（可选）：被隐藏焊盘的层数组

EViaType 过孔类型枚举值如下：

值	含义
NORMAL	通孔(普通过孔)
BLIND	盲埋孔
SUTURE	缝合孔

根据键名和具体值组成的数据格式如下：

```
{ "type": "VIA", "ticket": 134, "id": "c34473f1ef63f532" } | { "partitionId": "", "groupId": "1", "netName": "123", "ruleName": "", "centerX": 100, "centerY": 100, "num": 1, "layerId": 1, "width": 10, "height": 10, "holeType": "NORMAL", "holeWidth": 5, "holeHeight": 5, "holeDepth": 10, "holeAngle": 0, "holeOffsetX": 0, "holeOffsetY": 0, "holeRelativeAngle": 0, "holePlated": true, "holePadType": "EPadFuncType", "holeTopSolderExpansion": 10, "holeBottomSolderExpansion": 10, "holeTopPasteExpansion": 10, "holeBottomPasteExpansion": 10, "holeConnectMode": "EPadConnect", "holeSpokeSpace": 10, "holeSpokeWidth": 10, "holeSpokeAngle": 0, "unusedInnerLayers": [], "padLen": 10, "refs": [] }
```

7.2.7 PAD（焊盘）

其外层数据类型名为 PAD；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 0 不分组，非 0 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
netName	string	是	NET，网络名称
num	string	是	焊盘编号
centerX	number	是	焊盘原点 X
centerY	number	是	焊盘原点 Y
padAngle	number	是	焊盘旋转角度（角度制）
hole	THoleDef	是	孔，null 表示无孔
defaultPad	TPadDef	是	默认焊盘：参考焊盘
specialPad	object[]	是	*特殊焊盘
padOffsetX	number	是	孔偏移 X
padOffsetY	number	是	孔偏移 Y
relativeAngle	number	是	孔相对焊盘旋转角度（角度制）
plated	boolean	是	plated 是否金属化孔壁
padType	EPadFuncType	是	焊盘功能
topSolderExpansion	number null	是	顶层阻焊扩展：null 为遵循规则
bottomSolderExpansion	number null	是	底层阻焊扩展：null 为遵循规则
topPasteExpansion	number null	是	顶层助焊扩展：null 为遵循规则
bottomPasteExpansion	number null	是	底层助焊扩展：null 为遵循规则
connectMode	EPadConnect null	是	热焊-连接方式：null 为遵循规则，枚举值查看 EPadConnect_焊盘-连接方式
spokeSpace	number null	是	热焊-发散间距：null 为遵循规则
spokeWidth	number null	是	热焊-发散线宽：null 为遵循规则
spokeAngle	number null	是	热焊-发散角度：null 为遵循规则
unusedInnerLayers	number[]	是	隐藏焊盘层（可选）：被隐藏焊盘的层数组
padLen	number	是	引脚长度
refs	string[]	否	关联的图元编号

EPadFuncType 焊盘功能枚举值如下：

值	含义
NORMAL	普通焊盘
TEST	测试点
MARKER	标识点

根据键名和具体值组成的数据格式如下：

```
{ "type": "PAD", "ticket": 131, "id": "b70a17b5a4f0460a" } | { "partitionId": "", "groupId": 0, "netName": "", "layerId": 1, "num": "1", "centerX": 100, "centerY": 100, "width": 10, "height": 10, "holeType": "NORMAL", "holeWidth": 5, "holeHeight": 5, "holeDepth": 10, "holeAngle": 0, "holeOffsetX": 0, "holeOffsetY": 0, "holeRelativeAngle": 0, "holePlated": true, "holePadType": "EPadFuncType", "holeTopSolderExpansion": 10, "holeBottomSolderExpansion": 10, "holeTopPasteExpansion": 10, "holeBottomPasteExpansion": 10, "holeConnectMode": "EPadConnect", "holeSpokeSpace": 10, "holeSpokeWidth": 10, "holeSpokeAngle": 0, "unusedInnerLayers": [], "padLen": 10, "refs": [] }
```

7.2.7.1 THoleDef（孔定义）

键名	值类型	是否必需	描述
holeType	EHoleType	是	孔类型（ROUND 长圆孔、RECT 方孔）
width	number	是	宽
height	number	是	高

根据键名和具体值组成的数据格式如下：

```
{ "type": "PAD", "ticket": 135, "id": "858d9aac8017c201" } | { "partitionId": "3323181973b61938", "groupId": 0, "netName": "", "layerId": 1, "num": "1", "centerX": 100, "centerY": 100, "width": 10, "height": 10, "holeType": "NORMAL", "holeWidth": 5, "holeHeight": 5, "holeDepth": 10, "holeAngle": 0, "holeOffsetX": 0, "holeOffsetY": 0, "holeRelativeAngle": 0, "holePlated": true, "holePadType": "EPadFuncType", "holeTopSolderExpansion": 10, "holeBottomSolderExpansion": 10, "holeTopPasteExpansion": 10, "holeBottomPasteExpansion": 10, "holeConnectMode": "EPadConnect", "holeSpokeSpace": 10, "holeSpokeWidth": 10, "holeSpokeAngle": 0, "unusedInnerLayers": [], "padLen": 10, "refs": [] }
```

7.2.7.2 TPadDef（焊盘定义）

7.2.7.2.1 长圆焊盘

键名	值类型	是否必需	描述
padType	ELLIPSE	是	固定值表示长圆焊盘
width	number	是	宽
height	number	是	高

7.2.7.2.2 矩形焊盘

键名	值类型	是否必需	描述
----	-----	------	----

键名	值类型	是否必需	描述
padType	RECT	是	固定值表示矩形焊盘
width	number	是	宽
height	number	是	高
radius	number	是	圆角半径

7.2.7.2.3 正多边形焊盘

键名	值类型	是否必需	描述
padType	OVAL	是	固定值表示正多边形焊盘
width	number	是	宽
height	number	是	高

7.2.7.2.4 多边形焊盘

键名	值类型	是否必需	描述
padType	POLYGON	是	固定值表示多边形焊盘
path	any[]	是	路径, 参考 PCB_单多边形的定义

7.2.8 LINE（直线）

其外层数据类型名为 `LINE`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 <code>0</code> 不分组，非 <code>0</code> 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
netName	string	是	NET，网络名称
startX	number	是	开始 X
startY	number	是	开始 Y
endX	number	是	结束 X
endY	number	是	结束 Y
width	number	是	线宽

根据键名和具体值组成的数据格式如下：

```
{ "type": "LINE", "ticket": 451, "id": "e155" } || { "partitionId": "", "groupId": 0, "netName": "+5V", "layerId": 1, "startX": 250.8, "startY": 250.8, "endX": 250.8, "endY": 250.8, "width": 0.5 }
```

7.2.9 ARC（圆弧线）

其外层数据类型名为 `ARC`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 <code>0</code> 不分组，非 <code>0</code> 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
netName	string	是	NET，网络名称
startX	number	是	开始 X
startY	number	是	开始 Y
endX	number	是	结束 X
endY	number	是	结束 Y
angle	number	是	圆弧角，逆时针正，顺时针负
width	number	是	线宽
arcType	EArcType	是	圆弧类型： <code>0</code> 两点圆弧， <code>1</code> 中心圆弧
specialColor	string	否	特殊颜色

EArcType 圆弧类型枚举值如下：

值	含义
DOT	两点圆弧
CENTER	中心圆弧

根据键名和具体值组成的数据格式如下：

```
{ "type": "ARC", "ticket": 20, "id": "e72" } || { "partId": "TSB008A2530A.1", "groupId": "", "locked": false, "zIndex": 13, "startX": 15, "startY": 15, "endX": 15, "endY": 15, "angle": 90, "width": 0.5, "arcType": 0 }
```

7.2.10 OBJ（二进制内嵌对象）

其外层数据类型名为 `OBJ`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 <code>0</code> 不分组，非 <code>0</code> 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
fileName	string	是	文件名
startX	number	是	左上 X

键名	值类型	是否必需	描述
startY	number	是	左上 Y
width	number	是	宽
height	number	是	高
angle	number	是	旋转角度（角度制），绕左上点
mirror	boolean	是	原始图片是否水平镜像
path	string	是	二进制数据， <code>blob:</code> 开头后带着的是 <code>BLOB</code> 文档里的唯一编号，代表引用其具体内容
specialColor	string	否	特殊颜色

根据键名和具体值组成的数据格式如下：

```
{ "type": "OBJ", "ticket": 132, "id": "50777617e3a102f4" } | { "partitionId": "", "groupId": 0, "layerId": 3, "fileName": "0d4dbd377ee32c90"
```

7.2.11 EQLEN_GRP（等长组）

其外层数据类型名为 `DIMENSION`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
name	string	是	等长组名称
sort	number	是	排序
pads	[string, string][]	是	用 <code>位号:焊盘编号</code> 标识焊盘的数组

根据键名和具体值组成的数据格式如下：

7.3 多边形体系

SVG 中 `path` 是一个对多边形优秀的抽象。但由于 `PCB` 内用不到其中相对位置等功能，并且有条件设计更方便解析的方式。所以仿造 SVG 的 `path` 创造了一种类似的表达多边形的方式。多边形体系内 `POLY` `REGION` `POUR` 支持互相转换

7.3.1 PCB 单多边形的定义

单多边形为首尾重合的一条不间断的线所描述的区域。如果首尾不重合需要将其自动重合。

```
[300, 200, "L", 400, 200, "ARC", 400, 220, 15, "C", 200, 500, 400, 300, 100, 100]
```

```
["R", 100, 200, 300, 300, 0]
```

```
["CIRCLE", 100, 200, 5, 1]
```

7.3.1.1 L（直线模式）

`X Y L X Y X Y ...` 模式为直线模式，所有坐标将用直线将其连一一连起来

7.3.1.2 ARC/CARC 圆弧模式

`startX startY ARC angle endX endY` 模式为圆弧模式

- `startX/startY` 开始坐标
- `angle`: 圆弧角，逆时针正，顺时针负
- `endX/endY` 结束坐标

`startX startY CARC angle endX endY` 中心圆弧交互模式

7.3.1.3 C（三阶贝塞尔模式）

`X1 Y1 C X2 Y2 X3 Y3 X4 Y4 ...` 模式为三阶贝塞尔模式，所有坐标为其控制点

7.3.1.4 R（矩形模式）

`R X Y width height rot isCCW round` 矩形模式与其它都不兼容，是一个独立的模式

- `X/Y`: 左上坐标
- `width`: 宽
- `height`: 高
- `rot`: 旋转角度
- `isCCW`: 是否逆时针
- `round`: 圆角半径

7.3.1.5 CIRCLE（圆形模式）

`CIRCLE cx cy r isCCW` 圆形模式与其它都不兼容，是一个独立的模式

- `cx/cy`: 中心点坐标
- `r`: 半径
- `isCCW`: 是否逆时针

7.3.2 PCB（复杂多边形的定义）

```
[[[单多边形1: 外框], [单多边形2: 内洞]]]
```

复杂多边形可以包含多个单多边形，固定第一个多边形顺时针，表示外框，后续所有多边形逆时针，表示内洞

7.3.3 PARTITION（分区）

其外层数据类型名为 `PARTITION`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
----	-----	------	----

键名	值类型	是否必需	描述
name	string	是	分区名称
fileUuid	string	是	子图 UUID
path	any[][]	是	分区形状路径，参考 PCB 复杂多边形的定义

根据键名和具体值组成的数据格式如下：

```
{ "type": "PARTITION", "ticket": 253, "id": "85dcb50b7ce51008" } | { { "name": "PARTITION1", "fileUuid": "803d0f93d825424a", "path": [ [ "R",
```

7.3.4 BOARD（板子）

其外层数据类型名为 `BOARD`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
path	any[][]	是	路径，参考 PCB 复杂多边形的定义

根据键名和具体值组成的数据格式如下：

```
{ "type": "BOARD", "ticket": 130, "id": "BOARD" } | { { "path": [ [ "R", -1815, 1640, 1800, 1570, 0, 0 ] ] }
```

7.3.5 POLY（折线）

其外层数据类型名为 `POLY`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 <code>0</code> 不分组，非 <code>0</code> 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度， <code>null</code> 为默认
netName	string	是	NET，网络名称
width	number	是	线宽
path	any[]	是	路径，参考 PCB 单多边形的定义
polyType	EPolyType	是	折线类型： <code>0</code> 普通 <code>1</code> 板框
specialColor	string	否	特殊颜色

EPolyType 折线类型枚举值如下：

值	含义
NORMAL	普通
BOARD_OUTLINE	板框

根据键名和具体值组成的数据格式如下：

```
{ "type": "POLY", "ticket": 16, "id": "e11" } | { { "partId": "CC0805KRX7R9BB104.1", "groupId": "", "locked": false, "zIndex": 11, "points": [ {
```

7.3.6 FILL（填充）

其外层数据类型名为 `FILL`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 <code>0</code> 不分组，非 <code>0</code> 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度， <code>null</code> 为默认
netName	string	是	NET，网络名称
width	number	是	线宽
fillStyle	EPcbFillStyle	是	填充模式： <code>0</code> 实心填充 <code>1</code> 网格填充 <code>2</code> 内电层填充
path	any[]	是	路径，参考 PCB 单多边形的定义
isBridgingCopper	boolean	否	是否是桥接铜
networkList	string[]	否	桥接网络
refs	string[]	否	关联的图元编号
specialColor	string	否	特殊颜色

EPcbFillStyle PCB 填充类型枚举值如下：

值	含义
SOLID	实心填充
GRID	网格填充
PLANE	内电层填充

根据键名和具体值组成的数据格式如下：

```
{ "type": "FILL", "ticket": 149, "id": "e22" } | { { "groupId": 0, "netName": "", "layerId": 49, "width": 0.2, "fillStyle": "SOLID", "path": [ [ "C
```

7.3.7 LAYER_FILL（层填充）

其外层数据类型名为 `LAYER_FILL`；外层数据的唯一编号是一个 JSON 数组字符串，数组按顺序定义的结构如：

顺序	值类型	描述
1	LAYER_FILL	固定值，作用是避免与其他唯一编号重复
2	number	层编号，固定枚举值看层定义

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
fill	TLayerFill[]	是	层填充对象数组

TLayerFill 层填充对象，按键名定义的类型如下表：

键名	值类型	是否必需	描述
id	string	是	唯一编号，随机编码
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 0 不分组，非 0 为组标志，相同组标志的为一组
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
netName	string	是	NET，网络名称
width	number	是	线宽
path	any[]	是	路径，参考 PCB 单多边形的定义
isBridgingCopper	boolean	否	是否是桥接铜
networkList	string[]	否	桥接网络
refs	string[]	否	关联的图元编号
specialColor	string	否	特殊颜色

根据键名和具体值组成的数据格式如下：

7.3.8 REGION（区域）

其外层数据类型名为 REGION；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 0 不分组，非 0 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
width	number	是	线宽
prohibitType	EProhibitType[]	是	禁止类型数组
path	any[][]	是	路径，参考 PCB 复杂多边形的定义
name	string	否	名称（可选）
regionType	ERegionType	是	区域类型：0 禁止区域，1 约束区域

EProhibitType 区域禁止类型枚举值如下：

值	含义
COMPONENT	禁止元件
VIA	禁止过孔
TRACK	禁止布线
FILL	禁止放置填充区域
COPPER	禁止覆铜
PLANE	禁止内电层

ERegionType 区域类型枚举值如下：

值	含义
PROHIBIT	禁止区域
CONSTRAINT	约束区域

根据键名和具体值组成的数据格式如下：

```
{"type":"REGION","ticket":1405,"id":"e4491"}|{"partitionId":"","groupId":0,"layerId":1,"width":0.2,"prohibitType":["COPPER"]}
```

7.3.9 POUR（覆铜边框）

其外层数据类型名为 POUR；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 0 不分组，非 0 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
netName	string	是	NET，网络名称
width	number	是	线宽
name	string	是	覆铜名称
order	number	是	覆铜优先级
path	any[][]	是	路径，参考 PCB 复杂多边形的定义
pourType	TPourDef	是	覆铜类型定义
keepIsland	boolean	是	是否保留孤岛

根据键名和具体值组成的数据格式如下：

```
{"type":"POUR","ticket":1409,"id":"e3211"}|{"partitionId":"","groupId":0,"netName":"GND","layerId":1,"width":0.2,"name":"E"}
```

7.3.9.1 TPourDef（覆铜类型定义）

7.3.9.1.1 实心填充

键名	值类型	是否必需	描述
pourType	TPourType.SOLID	是	实心填充
fineness	number	是	最小覆铜细度

```
["SOLID", 2]
```

- 实心填充 `SOLID`
- 最小覆铜细度（生产优化用，AD 里的 Neck），`0` 为不开启生产优化

```
["POUR", "e100", "GND", 1, "BOTGND", 2, 复杂多边形, ["SOLID", 2], 1, 0]
```

```
["POUR", "e100", "GND", 1, "BOTGND", 2, 复杂多边形, ["SOLID", 0], 1, 0]
```

7.3.9.1.2 线填充

键名	值类型	是否必需	描述
pourType	TPourType.GRID_LINE TPourType.HORIZONTAL_LINE TPourType.VERTICAL_LINE	是	线填充类型
angle	number	是	旋转角度
strokeWidth	number	是	线宽
strokeSpacing	number	是	线距

```
["LINE", 0, 0, 10, 20]
```

- 线填充 `LINE`
- 填充模式：`0` 网格填充 `1` 水平线填充 `2` 垂直线填充
- 旋转角度
- 线宽
- 线距

```
["POUR", "e100", "GND", 1, "", 9, 复杂多边形, ["LINE", 0, 0, 10, 20], 0.6, 1, 0, 0]
```

7.3.10 POURED（覆铜结果）

其外层数据类型名为 `POURED`；外层数据的唯一编号是一个 `JSON` 数组字符串，数组按顺序定义的结构如：

顺序	值类型	描述
1	<code>POURED</code>	固定值，作用是避免与其他唯一编号重复
2	<code>string</code>	覆铜边框图元对应的唯一编号

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
pourFill	TPourFill[]	是	覆铜填充路径数组

7.3.10.1 TPourFill（覆铜填充路径）

键名	值类型	是否必需	描述
id	<code>string</code>	是	唯一标识
strokeWidth	<code>number</code>	是	描边线宽： <code>0</code> 为不描边
fill	<code>boolean</code>	是	是否填充
path	<code>any[][]</code>	是	路径，参考 PCB 复杂多边形的定义

根据键名和具体值组成的数据格式如下：

```
{"type":"POURED","ticket":400,"id":["\POURED\","\e70\"]}||{"pourFill":[{"strokeWidth":0,"fill":true,"path":[[866.41,1935,
```

7.3.11 IMAGE（图片）

其外层数据类型名为 `IMAGE`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	<code>string</code>	否	所属分区编号，不存在或为空字符串表示无分区
groupId	<code>string</code>	是	分组编号，没有则为 <code>0</code> 不分组，非 <code>0</code> 为组标志，相同组标志的为一组
layerId	<code>number</code>	是	层编号
locked	<code>boolean</code>	是	是否锁定
zIndex	<code>number null</code>	是	<code>Z</code> 轴高度， <code>null</code> 为默认
startX	<code>number</code>	是	左上 <code>X</code>
startY	<code>number</code>	是	左上 <code>Y</code>
width	<code>number</code>	是	宽
height	<code>number</code>	是	高
angle	<code>number</code>	是	旋转角度（角度制），绕起始点
mirror	<code>boolean</code>	是	原始图片是否水平镜像
path	<code>any[][]</code>	是	路径，参考 PCB 复杂多边形的定义
specialColor	<code>string</code>	否	特殊颜色

根据键名和具体值组成的数据格式如下：

```
{"type":"IMAGE","ticket":129,"id":["60e26b227303cb51"]}||{"partitionId":"","groupId":0,"layerId":1,"startX":-1130,"startY":15
```

7.3.12 TEARDROP（泪滴）

其外层数据类型名为 `TEARDROP`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	<code>string</code>	否	所属分区编号，不存在或为空字符串表示无分区

键名	值类型	是否必需	描述
groupId	string	是	分组编号，没有则为 0 不分组，非 0 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
netName	string	是	NET，网络名称
path	any[]	是	路径，参考 PCB 单多边形的定义
refs	string[]	否	关联的图元编号

根据键名和具体值组成的数据格式如下：

```
{"type":"TEARDROP","ticket":1423,"id":"e1356"}|{"partitionId":"","netName":"+3.3V","layerId":1,"path":[1906.1323,-1523.632
```

7.3.13 FPC_FILL（柔性工艺补强板）

其外层数据类型名为 FPC_FILL；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 0 不分组，非 0 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
material	EFpcMaterial	是	材质
thickness	number	是	厚度
path	any[][]	是	路径，参考 PCB 复杂多边形的定义

EFpcMaterial 柔性工艺补强板材质枚举值如下：

值	含义
PI	Polyimide
FR4	FR4
STEEL_SHEET	Stainless Steel
M3_DOUBLE_SIDED_TAPE	3M Tape
EMI_SHIELDING_FILM	EMI Shielding Film

根据键名和具体值组成的数据格式如下：

```
{"type":"FPC_FILL","ticket":132,"id":"4b81c6c47e46d66d"}|{"partitionId":"","groupId":0,"layerId":58,"material":"Polyimide"
```

7.4 3D 外壳体系

7.4.1 SHELL（外壳）

其外层数据类型名为 SHELL；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 0 不分组，非 0 为组标志，相同组标志的为一组
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
shellType	E3DShellOutlineType	是	外壳类型
shellHeight	number	是	外壳高度
pcbHeight	number	是	PCB 高度
strokeWidth	number	是	*线宽（弃用）
path	any[][]	是	路径，参考 PCB 复杂多边形的定义
thickness	number	是	外壳厚度
direction	E3DShellPushCoverDir	否	推盖方向
bottomHeight	number	否	下壳高度
topInnerHeight	number	否	上壳内壁高度
bottomInnerHeight	number	否	下壳内壁高度

E3DShellOutlineType 3D 外壳类型枚举值如下：

值	含义
PUSH_COVER	推盖
UPPER_LOWER_SHELLS	上下壳

E3DShellPushCoverDir 3D 外壳推盖方向枚举值如下：

值	含义
X_AXIS_POSITIVE	X 轴正向
X_AXIS_NEGATIVE	X 轴负向
Y_AXIS_POSITIVE	Y 轴正向
Y_AXIS_NEGATIVE	Y 轴负向

根据键名和具体值组成的数据格式如下：

```
{"type":"SHELL","ticket":133,"id":"635cf5dbed32ac34"}|{"partitionId":"","groupId":0,"shellType":"T&B","shellHeight":1181.1
```

7.4.2 CREASE（侧面基准线（折痕））

其外层数据类型名为 CREASE；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 0 不分组，非 0 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
startX	number	是	起始 X
startY	number	是	起始 Y
endX	number	是	结束 X
endY	number	是	结束 Y
angle	number	是	折叠角度

根据键名和具体值组成的数据格式如下：

```
{"type":"CREASE","ticket":134,"id":"7950955c69851729"}|{"partitionId":"","groupId":0,"layerId":53,"startX":-10490,"startY":
```

7.4.3 SHELL_ENTITY（外壳实体区域）

其外层数据类型名为 SHELL_ENTITY；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 0 不分组，非 0 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
type	EEntityType	是	实体区域类型
entityBelong	EEntityBelong	否	实体 - 归属
slotOptions	ESlotOption[]	否	挖槽 - 选项
depth	number	是	深度
strokeWidth	number	是	线宽
path	any[][]	是	路径，参考 PCB 复杂多边形的定义
refs	string[]	否	关联的图元编号

EEntityType 实体区域类型枚举值如下：

值	含义
SLOT	挖槽
ENTITY	实体

EEntityBelong 实体归属枚举值如下：

值	含义
AUTO	自动
UPPER_SHELLS	上壳
LOWER_SHELLS	下壳

ESlotOption 挖槽选项枚举值如下：

值	含义
OUTLINE	外壳边框
SCREW_PILLAR	螺丝柱
ENTITY	实体

根据键名和具体值组成的数据格式如下：

```
{"type":"SHELL_ENTITY","ticket":134,"id":"307b9ad6b32b684e"}|{"partitionId":"","groupId":0,"layerId":53,"depth":590.55,"st
```

7.4.4 BOSS（螺丝柱）

其外层数据类型名为 BOSS；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 0 不分组，非 0 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
centerX	number	是	中心 X
centerY	number	是	中心 Y
specification	ESpecification null	是	螺丝型号：“M2” “M3” 等，或者 null 表示自定义
depth	number	是	螺丝柱高度
thickness	number	是	螺丝柱壁厚
holeDiameter	number	是	螺丝柱通孔直径
counterBore	object null	是	沉头参数
counterBore.height	number	是	沉头孔高度
counterBore.diameter	number	是	沉头孔直径
stiffener	object null	是	加强筋参数
stiffener.topWidth	number	是	加强筋上端宽度
stiffener.bottomWidth	number	是	加强筋下端宽度
stiffener.height	number	是	加强筋高度

键名	值类型	是否必需	描述
stiffener.stiffThickness	number	是	加强筋厚度

ESpecification 螺丝型号枚举值如下：

值	含义
M2	M2
M3	M3
M4	M4
M5	M5
M6	M6

根据键名和具体值组成的数据格式如下：

```
{ "type": "BOSS", "ticket": 132, "id": "d087ce2533d63dfa" } || { "partitionId": "", "groupId": 0, "layerId": 54, "centerX": -9880, "centerY":
```

7.5 文字体系

7.5.1 STRING（文字）

其外层数据类型名为 `STRING`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 <code>0</code> 不分组，非 <code>0</code> 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
x	number	是	位置 X
y	number	是	位置 Y
text	string	是	内容
fontFamily	string	是	字体名称
fontSize	number	是	字号
strokeWidth	number	是	粗细
bold	boolean	是	是否加粗
italic	boolean	是	是否斜体
origin	EAlign	是	对齐模式，枚举值查看 EAlign 对齐模式
angle	number	是	旋转角度（角度制）
reverse	boolean	是	是否反相扩展
expansion	number	是	反相扩展尺寸
mirror	boolean	是	是否镜像
specialColor	string	否	特殊颜色

根据键名和具体值组成的数据格式如下：

```
{ "type": "STRING", "ticket": 138, "id": "e97211d9113b9f5c" } || { "partitionId": "", "groupId": 0, "layerId": 1, "x": -3535, "y": -4915, "text
```

7.5.2 ATTR（属性）

其外层数据类型名为 `ATTR`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 <code>0</code> 不分组，非 <code>0</code> 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
parentId	string	是	父级编号
x	number null	是	位置 X，null 为没显示过
y	number null	是	位置 Y，null 为没显示过
key	string	是	Key
value	string	是	Value
keyVisible	boolean	是	是否显示 Key
valueVisible	boolean	是	是否显示 Value
fontFamily	string null	是	字体名称，null 为默认
fontSize	number null	是	字号，null 为默认
strokeWidth	number	是	粗细
bold	boolean	是	是否加粗
italic	boolean null	是	是否斜体，null 为默认
origin	EAlign	是	对齐模式，枚举值查看 EAlign 对齐模式
angle	number	是	旋转角度（角度制）
reverse	boolean	是	是否反相扩展
expansion	number	是	反相扩展尺寸
mirror	boolean	是	是否镜像
specialColor	string	否	特殊颜色

根据键名和具体值组成的数据格式如下：

```
{ "type": "ATTR", "ticket": 146, "id": "513026b3e6259d89e0" } || { "partitionId": "", "groupId": 0, "parentId": "513026b3e6259d89", "layerI
```

7.5.3 DIMENSION（尺寸工具集）

其外层数据类型名为 DIMENSION；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 0 不分组，非 0 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
type	EDimensionType	是	尺寸类型
unit	EUnitName	是	单位
strokeWidth	number	是	线宽
precision	number	是	精度
textFollow	boolean	是	文字是否跟随
coords	number[]	是	坐标集
text	TPcbString	是	文本
specialColor	string	否	特殊颜色

EDimensionType 尺寸类型枚举值如下：

值	含义
LENGTH	长度
RADIUS	半径
ANGLE	角度

EUnitName 单位枚举值如下：

值	含义
MIL	密耳
INCH	英寸
CM	厘米
MM	毫米

根据键名和具体值组成的数据格式如下：

```
{ "type": "DIMENSION", "ticket": 139, "id": "7fc2c0598d155d64" } | { "partitionId": "", "type": "LENGTH", "layerId": 13, "unit": "mil", "str
```

7.6 拼板

7.6.1 PANELIZE（拼版）

其外层数据类型名为 PANELIZE；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
on	boolean	是	是否启用
row	number	是	行数
column	number	是	列数
rowSpacing	number	是	行距
columnSpacing	number	是	列距
onlyOutline	boolean	是	是否只拼边框
horizontalStamp	TPanelizeStamp	是	水平邮票孔参数
verticalStamp	TPanelizeStamp	是	垂直邮票孔参数
horizontalSize	TPanelizeSide	是	水平工艺边参数
verticalSize	TPanelizeSide	是	垂直工艺边参数

根据键名和具体值组成的数据格式如下：

```
{ "type": "PANELIZE", "ticket": 130, "id": "PANELIZE" } | { "on": false, "row": 1, "column": 1, "rowSpacing": 0, "columnSpacing": 0, "onlyOutl
```

7.6.1.1 TPanelizeStamp（邮票孔参数）

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
on	boolean	是	是否启用
stampHoleGroupQuantity	number	是	邮票孔组数
stampHoleDiameter	number	是	邮票孔直径
stampHoleQuantityPerGroup	number	是	邮票孔每组数量
stampHoleSpacing	number	是	邮票孔间距

7.6.1.2 TPanelizeSide（工艺边参数）

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
on	boolean	是	是否启用
sideHeight	number	是	工艺边高度
positionHoleDiameter	number	是	定位孔直径
markDiameter	number	是	Mark 点直径
markExpansion	number	是	Mark 点阻焊扩展

7.7 封装体系

7.7.1 COMPONENT（器件实例）

其外层数据类型名为 COMPONENT；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
groupId	string	是	分组编号，没有则为 0 不分组，非 0 为组标志，相同组标志的为一组
layerId	number	是	层编号
locked	boolean	是	是否锁定
zIndex	number null	是	Z 轴高度，null 为默认
x	number	是	位置 X
y	number	是	位置 Y
angle	number	是	旋转角度（角度制）
attrs	object	是	自定义属性
specialColor	string	否	特殊颜色

根据键名和具体值组成的数据格式如下：

```
{"type":"COMPONENT","ticket":141,"id":"f9b37caf21e1d7e2"}|{"partitionId":"","groupId":0,"layerId":1,"x":-11195,"y":-7820,"
```

7.7.2 PAD_NET（焊盘实例网络映射）

其外层数据类型名为 PAD_NET；外层数据的唯一编号是一个 JSON 数组字符串，数组按顺序定义的结构如：

顺序	值类型	描述
1	PAD_NET	固定值，作用是避免与其他唯一编号重复
2	string	所属器件实例编号
3	string	焊盘编号
4	string	封装内焊盘 ID

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
partitionId	string	否	所属分区编号，不存在或为空字符串表示无分区
padNet	string	是	网络名
padLen	number	是	引脚长度

根据键名和具体值组成的数据格式如下：

```
{"type":"PAD_NET","ticket":145,"id":["PAD_NET","f9b37caf21e1d7e2","2","e14"]}|"partitionId":"","padNet":"","pac
```

7.8 设计规则体系

7.8.1 RULE_TEMPLATE（设计规则模板）

设计规则模板逻辑上有两种理解方式，PCB 自行根据需要选择

1. 作为其他设计规则的基版，其他设计规则是对模板的覆盖
2. 与其他设计规则互斥，有模板其他设计规则只是暂存，不产生任何效果
3. 作为来源于哪个模板的标识，不影响后续实际规则的效力（当前采用的方案）

其外层数据类型名为 RULE_TEMPLATE；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
name	string	是	模板名称

根据键名和具体值组成的数据格式如下：

```
{ "type": "RULE_TEMPLATE", "ticket": 1 }|{ "name": "JLPCB Capability(High Frequency Board)" }
```

7.8.2 RULE（设计规则）

同一 规则类型 设计规则出现的顺序，需要和【规则管理】左侧树的顺序一致

其外层数据类型名为 RULE；外层数据的唯一编号是一个 JSON 数组字符串，数组按顺序定义的结构如：

顺序	值类型	描述
1	RULE	固定值，作用是避免与其他唯一编号重复
2	ERuleType	规则类型：具体看规则类型枚举定义
3	string	规则名称

ERuleType 设计规则类型枚举值如下：

值	含义
SAFE	间距 - 安全间距
OTHER	间距 - 其他间距
TRACK	物理 - 导线
BLIND	物理 - 盲埋孔
RADIUS	物理 - 孔径
PLANE	平面 - 内电层
COPPER	平面 - 铺铜
PASTE	扩展 - 阻焊
SOLDER	扩展 - 助焊
DIFFER_ENTAIL	物理 - 差分对

值	含义
NET_LENGTH	物理 - 网络长度范围
AUTO_ROUTER	布线 - 自动布线
NET_LENGTH_TOLERANCE	物理 - 网络长度公差

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
ruleState	ERuleStatus	是	规则状态：具体看规则状态枚举定义
ruleContext	TRuleContext[ERuleType]	是	规则内容：与规则类型值绑定，一个规则类型对应一个规则内容

ERuleStatus 规则状态枚举值如下：

值	含义
NORMAL	普通规则
DEFAULT	默认规则

7.8.2.1 TRuleContext[ERuleType] 规则内容

ERuleType 类型对应的规则内容如下

7.8.2.1.1 ERuleType.SAFE 间距 - 安全间距

按键名定义的类型如下表：

键名	值类型	是否必需	描述
unit	string	是	单位
isForAll	EShowStatus	是	显示状态
safeSpacing	TSafeSpacing[]	是	安全间距配置

safeSpacing 字段的数组元素 TSafeSpacing 结构：

键名	值类型	是否必需	描述
layerId	number	是	层编号
isOpen	boolean	是	是否展开
content	number[][]	是	具体内容
columnNames	string[]	是	列名

EShowStatus 安全间距-显示状态枚举值如下：

值	含义
ALL	全部
LAYERED	分层

根据键名和具体值组成的数据格式如下：

```
{ "type": "RULE", "id": "UUID", "ticket": 1 } [{"ruleType": "Safe Clearance", "ruleName": "通用", "ruleState": 1, "ruleContext": {}]
```

7.8.2.1.2 ERuleType.OTHER 间距 - 其他间距

按键名定义的类型如下表：

键名	值类型	是否必需	描述
unit	string	是	单位
deviceClearance	number	是	元件到元件
thru2SmdClearance	number	是	插件焊盘到 SMD 元件
holeClearance	number	是	钻孔到钻孔

7.8.2.1.3 ERuleType.TRACK 物理 - 导线

按键名定义的类型如下表：

键名	值类型	是否必需	描述
unit	string	是	单位
track	object	是	导线配置

track 字段的对象属性：

键名	值类型	是否必需	描述
isOpen	boolean	是	是否展开
content	object[]	是	具体内容

content 字段的数组元素结构：

键名	值类型	是否必需	描述
layerId	number	是	层编号
stroMin	number	是	线宽 - 最小值
stroDef	number	是	线宽 - 默认值
stroMax	number	是	线宽 - 最大值

7.8.2.1.4 ERuleType.BLIND 物理 - 盲埋孔

按键名定义的类型如下表：

键名	值类型	是否必需	描述
blinds	object	是	盲埋孔配置

blinds 字段的对象属性：

键名	值类型	是否必需	描述
content	object[]	是	具体内容

content 字段的数组元素结构：

键名	值类型	是否必需	描述
name	string	是	名称
staLayerId	number	是	起始层
endLayerId	number	是	结束层
viaSizeRule	string	是	过孔尺寸

7.8.2.1.5 ERuleType.RADIUS 物理 - 孔径

按键名定义的类型如下表：

键名	值类型	是否必需	描述
unit	string	是	单位
minRadius	number	是	过孔外直径 - 最小
defRadius	number	是	过孔外直径 - 默认
maxRadius	number	是	过孔外直径 - 最大
minInner	number	是	过孔内直径 - 最小
defInner	number	是	过孔内直径 - 默认
maxInner	number	是	过孔内直径 - 最大

7.8.2.1.6 ERuleType.PLANE 平面 - 内电层

按键名定义的类型如下表：

键名	值类型	是否必需	描述
unit	string	是	单位
mulPad	object	是	多层焊盘配置

mulPad 对象属性：

键名	值类型	是否必需	描述
isOpen	boolean	是	是否展开
content	数组对象	是	具体内容

mulPad 字段的对象里的 content 数组元素结构：

键名	值类型	是否必需	描述
layerId	number	是	层编号
connType	EpadConnect	是	连接方式，枚举值查看 EpadConnect_焊盘-连接方式
spoSpac	number	是	发散间距
spoWidth	number	是	发散线宽
spoAng	number	是	发散角度

7.8.2.1.7 ERuleType.COPPER 平面 - 铺铜

按键名定义的类型如下表：

键名	值类型	是否必需	描述
unit	string	是	单位
sglPad	object	是	单层焊盘配置
mulPad	object	是	多层焊盘配置
track	object	是	导线配置

对象属性（sglPad, mulPad, track 共有）：

键名	值类型	是否必需	描述
isOpen	boolean	是	是否展开
content	数组对象	是	具体内容

sglPad/mulPad 的 content 数组元素结构：

键名	值类型	是否必需	描述
layerId	number	是	层编号
connType	EpadConnect	是	连接方式，枚举值查看 EpadConnect_焊盘-连接方式
spoSpac	number	是	发散间距
spoWidth	number	是	发散线宽
spoAng	number	是	发散角度

track 的 content 数组元素结构：

键名	值类型	是否必需	描述
layerId	number	是	层编号
connType	ETrackConnect	是	连接方式

ETrackConnect 规则-导线-连接方式枚举值如下：

值	含义
DIRECT	直连
NOM_CONNECT	无连接

7.8.2.1.8 ERuleType.PASTE 扩展 - 助焊

按键名定义的类型如下表：

键名	值类型	是否必需	描述
unit	string	是	单位
padTopExpan	number	是	焊盘 - 顶层扩展
padBotExpan	number	是	焊盘 - 底层扩展
testPointTopExpan	number	是	测试点 - 顶层扩展
testPointBotExpan	number	是	测试点 - 底层扩展

7.8.2.1.9 ERuleType.SOLDER 扩展 - 阻焊

按键名定义的类型如下表：

键名	值类型	是否必需	描述
unit	string	是	单位
padTopExpan	number	是	焊盘 - 顶层扩展
padBotExpan	number	是	焊盘 - 底层扩展
viaTopExpan	number	是	过孔 - 顶层扩展
viaBotExpan	number	是	过孔 - 底层扩展
testPointTopExpan	number	是	测试点 - 顶层扩展
testPointBotExpan	number	是	测试点 - 底层扩展

7.8.2.1.10 ERuleType.DIFFER_ENTAIL 物理 - 差分对

按键名定义的类型如下表：

键名	值类型	是否必需	描述
unit	string	是	单位
stroWidth	object	是	线宽配置
spacing	object	是	间距配置
differPairLenTolerMax	number	是	差分对长度误差

对象属性（stroWidth 和 spacing 共有）：

键名	值类型	是否必需	描述
isOpen	boolean	是	是否展开
content	数组对象	是	具体内容

stroWidth 的 content 数组元素结构：

键名	值类型	是否必需	描述
layerId	number	是	层编号
stroMin	number	是	线宽 - 最小值
stroDef	number	是	线宽 - 默认值
stroMax	number	是	线宽 - 最大值

spacing 的 content 数组元素结构：

键名	值类型	是否必需	描述
layerId	number	是	层编号
spacMin	number	是	间距 - 最小值
spacDef	number	是	间距 - 默认值

7.8.2.1.11 ERuleType.NET_LENGTH 物理 - 网络长度范围

按键名定义的类型如下表：

键名	值类型	是否必需	描述
unit	string	是	单位
netLenMin	number	是	最大值
netLenMax	number	是	最小值

7.8.2.1.12 ERuleType.AUTO_ROUTER 布线 - 自动布线

按键名定义的类型如下表：

键名	值类型	是否必需	描述
routingCorner	EAutoRoutingCorner	是	布线角度
isKeep	boolean	是	是否保留已有布线数据
routingEffectPriority	EAutoRoutingPriority	是	效果优先级(已废弃)
layers	number[]	是	布线层
ignoreNets	string[]	是	忽略的网络

EAutoRoutingCorner 自动布线-拐角枚举值如下：

值	含义
FORTY_FIVE	45 度
NINETY	90 度拐角
NINETY_ARC	90 度圆弧拐角

EAutoRoutingPriority 自动布线-效果优先级枚举值如下：

值	含义
SPEED_FIRST	速度优先
COMPLETION_FIRST	完成度优先

7.8.2.1.13 ERuleType.NET_LENGTH_TOLERANCE 物理 - 网络长度公差

按键名定义的类型如下表：

键名	值类型	是否必需	描述
unit	string	是	单位
netLengthTolerance	number	是	网络长度公差

7.8.3 RULE_SELECTOR（规则选择器）

其外层数据类型名为 `RULE_SELECTOR`；外层数据的唯一编号是一个 `JSON` 数组字符串，数组按顺序定义的结构如：

顺序	值类型	
1	<code>RULE_SELECTOR</code>	固定值，作用是避免与其他唯一编号重复
2	<code>{(string number (string number))[]}</code>	选择器：1、网络类型： <code>["NET_CLASS", "High Speed"]</code> ；2、网络： <code>["NET", "GND"]</code> ；3、层： <code>["LAYER", 3]</code> ；4、区域： <code>["REGION", "e10"]</code> ；5、 <code>["FOOT</code>

其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
ruleOrder	number	是	优先级：数值越小，优先级越高
ruleKeyValue	<code>{[key in ERuleType]?: string}</code>	否	规则， <code>Key</code> 为规则类， <code>Value</code> 为规则名称，每个规则类下只能选择一个规则
copperValue	<code>{[ERuleType.COPPER]?: string}</code>	否	铺铜规则(选择值)
innerPlaneValue	<code>{[ERuleType.PLANE]?: string}</code>	否	内电层规则(选择值)

根据键名和具体值组成的数据格式如下：

```
{ "type": "RULE_SELECTOR", "id": "UUID", "ticket": 1 } | { "ruleSelect": ["NET", "GND"], "ruleOrder": 0, "ruleKeyValue": { "Safe Cl
```

8 CONFIG（工程设置格式）

8.1 META（基本信息）

其外层数据类型名为 `META`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
defaultSheet	string	是	默认图纸 uuid

根据键名和具体值组成的数据格式如下：

```
{ "type": "META", "ticket": 29, "id": "META" } | { "defaultSheet": "ae038fff0b42ee51" } |
```

8.2 UNIVERSAL（通用设置）

其外层数据类型名为 `UNIVERSAL`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
allowLibRename	boolean	是	是否工程库重名
defaultNetName	boolean	是	原理图 - 是否默认网络名
wireMultipleNet	boolean	是	原理图 - 是否导线多网络
netFlagCrossLayerConnection	boolean	是	原理图 - 是否网络标识跨层连接
netLabelCrossPageConnection	boolean	是	原理图 - 是否网络标识跨页连接
busGenerateNetClass	boolean	是	原理图 - 是否网络标识总线生成网络类
relevanceDisplayRow	<code>ERelevanceDisplayRowType</code>	是	原理图 - 显示方式
relevanceBelongSchPage	<code>ERelevanceBelongSchPage</code>	是	原理图 - 所在图页
relevanceLocation	<code>ERelevanceLocation</code>	是	原理图 - 所在位置

8.2.1 ERelevanceDisplayRowType（原理图 - 显示方式）

值	含义
SINGLE	单行显示
MULTIPLE	多行显示

8.2.2 ERelevanceBelongSchPage（原理图 - 所在图页）

值	含义
NONE	不显示
NAME	名称
NUMBER	序号

8.2.3 ERelevanceLocation（原理图 - 所在位置）

值	含义
NONE	不显示
ZONE	区域

根据键名和具体值组成的数据格式如下：

```
{ "type": "UNIVERSAL", "ticket": 1, "id": "UNIVERSAL" } | { "allowLibRename": true, "defaultNetName": true, "wireMultipleNet": true, "netF
```

9 SCH（原理图格式）

9.1 META（基本信息）

其外层数据类型名为 `META`；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
----	-----	------	----

键名	值类型	是否必需	描述
title	string	是	文档名称
source	string	是	来源的 uuid(导入的图页才有)
zIndex	number	是	排序大小
board	string	是	所属板子

根据键名和具体值组成的数据格式如下：

```
{ "type": "META", "ticket": 5, "id": "META" } || { "title": "Schematic1", "source": "", "board": "b097b3c4a9a9ac20", "zIndex": null }
```

9.2 原理图设计规则

原理图的设计规则与 PCB 页的设计规则一样的结构，此处不再重复叙述

9.3 DIFFERENTIAL_PAIR（差分对）

其外层数据类型名为 DIFFERENTIAL_PAIR；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
name	string	是	名称
negativeNet	string	是	负级网络
positiveNet	string	是	正级网络

根据键名和具体值组成的数据格式如下：

```
{ "type": "DIFFERENTIAL_PAIR", "ticket": 405, "id": "[\ "DIFFERENTIAL_PAIR\ ", \ "DP1\ "]" } || { "positiveNet": "$1N1", "negativeNet": "$1N2" }
```

9.4 NET_CLASS（网络类）

其外层数据类型名为 NET_CLASS；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
name	string	是	名称
nets	string[]	是	网络组

根据键名和具体值组成的数据格式如下：

```
{ "type": "NET_CLASS", "ticket": 387, "id": "[\ "NET_CLASS\ ", \ "AAS\ "]" } || { "nets": [] }
```

9.5 EQL_NET_GRP（等长网络组）

其外层数据类型名为 EQL_NET_GRP；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
name	string	是	名称
nets	string[]	是	网络组

根据键名和具体值组成的数据格式如下：

```
{ "type": "EQL_NET_GRP", "ticket": 407, "id": "[\ "EQL_NET_GRP\ ", \ "AASD\ "]" } || { "nets": [] }
```

10 DEVICE（器件格式）

10.1 META（基本信息）

其外层数据类型名为 META；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
title	string	是	名称
source	string	是	来源的 uuid(工程库独有)
tags	string[]	是	自定义分类
images	string[]	是	图片链接
attributes	ATTRIBUTES	是	属性

根据键名和具体值组成的数据格式如下：

```
{ "type": "META", "ticket": 28, "id": "META" } || { "title": "Drawing-Symbol_A4", "tags": [], "source": "bc676184ec9748d7b372ad543982403a" }
```

11 BOARD（板子格式）

11.1 META（基本信息）

其外层数据类型名为 META；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
title	string	是	板子名称
zIndex	number	是	表示排序层级

根据键名和具体值组成的数据格式如下：

```
{ "type": "META", "ticket": 3, "id": "META" } || { "title": "Board1", "zIndex": 1 }
```

12 BLOB（二进制数据格式）

目前仅存储图片，全工程共用，对应图页里只记录数据的 `uuid`

键名	值类型	是否必需	描述
notGrouped	boolean	是	是否有未归组

根据键名和具体值组成的数据格式如下：

```
{ "type": "VARIANT_GROUPED", "ticket": 322, "id": "VARIANT_GROUPED" } || { "notGrouped": false }
```

15.3 META_Z_INDEX（排序信息）

其外层数据类型名为 META_Z_INDEX；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
zIndex	number	是	排序

根据键名和具体值组成的数据格式如下：

```
{ "type": "META_Z_INDEX", "ticket": 321, "id": "META_Z_INDEX" } || { "zIndex": 1 }
```

16 COMPONENT_GROUP（元件分组格式）

16.1 META（基本信息）

其外层数据类型名为 META；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
title	string	是	名称
parent	string	是	子元件分组的父元件分组 uuid
belong	string	是	属于直接变体的元件分组不参与元件分组树构建
schematicId	string	是	原理图

根据键名和具体值组成的数据格式如下：

```
{ "type": "META", "ticket": 323, "id": "META" } || { "title": "", "schematicId": "342654b3f16c4494", "belong": "872d21fald4f57df", "parent": "" }
```

16.2 META_Z_INDEX（排序信息）

其外层数据类型名为 META_Z_INDEX；其内层数据按键名定义的类型如下表：

键名	值类型	是否必需	描述
zIndex	number	是	元件分组树中排序

根据键名和具体值组成的数据格式如下：

```
{ "type": "META_Z_INDEX", "ticket": 324, "id": "META_Z_INDEX" } || { "zIndex": 0 }
```

16.3 GROUP_INDEX（元件分组在变体的排序）

键名	值类型	是否必需	描述
zIndex	number	是	在变体树中的排序

其外层格式的 id 就是变体的唯一编号,格式示例：

```
{ "type": "GROUP_INDEX", "ticket": 374, "id": "872d21fald4f57df" } || { "zIndex": 1 }
```

17 GROUP_DATA（元件分组的属性数据格式）

17.1 文档头 id

组合 id,以 @ 分割：

- 第一个是图元 id
- 第二个是实例页 id

```
{ "type": "GROUP_DATA", "id": "e176@uuid", "ticket": 1 } || data
```

17.2 GROUP_DATA（属性数据）

其外层数据类型名为 GROUP_DATA；其内层数据是一个以字符串为键名任意字符为值的字符对象 其文档格式示例如下：

```
{ "type": "GROUP_DATA", "id": "e176@uuid1", "ticket": 1 } || { "Designator": "U15", "ASDF": "1234" }  
{ "type": "GROUP_DATA", "id": "e177@uuid1", "ticket": 2 } || { "NUMBER": 2 }  
{ "type": "GROUP_DATA", "id": "e176@uuid2", "ticket": 3 } || { "Author": "abc" }
```