



IDF 导出器

July 18, 2019

Contents

1 IDFv3 导出器简介 2

2 指定供导出程序使用的元件模型 2

3 创建元件边框文件 4

4 创建边框的准则 6

4.1 包命名 6

4.2 注释 6

4.3 几何和部件编号条目 6

4.4 引脚方向和定位 6

4.5 尺寸提示 7

5 IDF 元件边框工具 8

5.1 idfcyl 8

5.2 idfrect 9

5.3 dxf2idf 10

6 idf2vrml 11

参考手册

Copyright

本文档是版权 (c) 2014-2015 年由其贡献者列出的如下。您可以根据 GNU 通用公共许可证 (<http://www.gnu.org/licenses/gpl.html>) 版本 3 或更高版本或知识共享归因许可证 (<http://creativecommons.org/licenses/by/3.0/>) 版本的条款分发和/修改它 3.0 或更高版本。

本指南中的所有商标均属于其合法所有者。

* 贡献者 *

Cirilo Bernardo

翻译

taotieren <admin@taotieren.com>, 2019

Telegram 简体中文交流群: https://t.me/KiCad_zh_CN

反馈

请将任何错误报告、建议或新版本引导到此处:

- 关于 KiCad 文档: <https://github.com/KiCad/kicad-doc/issues>
- 关于 KiCad 软件: <https://bugs.launchpad.net/kicad>
- 关于 KiCad 软件 i18n: <https://github.com/KiCad/kicad-i18n/issues>

出版日期和软件版本

2014 年 1 月 26 日发布。

1 IDFv3 导出器简介

IDF 导出器导出 IDFv3 脚注: [http://www.simplifiedsolutionsinc.com/images/idf_v30_spec.pdf] 兼容板 (.emn) 和库 (.emp) 文件, 用于将机械尺寸传达给机械 CAD 包。导出器目前出口板的边框和切口, 所有垫和安装通孔, 包括槽孔和元件边框; 这是与机械设计师互动所需的最基本的机械数据集。目前未导出 IDFv3 规范中描述的所有其他实体。

2 指定供导出程序使用的元件模型

IDF 导出器使用最初由 3D 查看器使用的 3D 模型文件属性。由于 3D 查看器, IDF 以及可能的未来机械 CAD 导出器通常对不同类型的文件格式感兴趣, 因此可以使用 3D 模型文件属性为多个导出器指定模型。

在封装编辑器或 Pcbnew 中, 编辑封装参数并单击 3D 设置选项卡 (参见链接: # 图-1[图-1]), 单击 **添加 3D 形状**, 然后选择过滤器 IDFv3 元件文件 (*.idf) (见链接: # 图-2[图-2])。选择所需的边框文件, 然后输入偏移和旋转所需的任何值。请注意, IDF 导出器仅使用偏移值和 Z 旋转值; 所有其他值都将被忽略。必须使用 IDF 板输出单位 (mm 或 thou) 和 IDF 坐标系指定偏移量, 这是一个右手坐标系, +Z 向观察者移动, +X 向观察者右侧, +Y 向观察者上侧。旋转必须以度为单位, 正旋转为逆时针旋转, 如 IDFv3 规范中所述。多个边框可以与适当的偏移组合以表示简单的元件, 例如插座中的 DIP 包。[BUG:] 在讨论中已经确定 Z 偏移的单位应该是英寸, 这与 VRML 模型偏移的单位一致。忽略 (X, Y) 偏移值也可能有用。这里提到的行为将在未来的某个时候发生变化。

一旦为所有需要的元件指定了模型, 从 pcbnew 中选择 **文件** 菜单, 然后选择 **导出**, 最后 **IDFv3 导出**。将弹出一个对话框 (参见链接: # 图-3[图-3]), 该对话框允许设置输出文件名和 IDF 输出单位 (mm 或 mils)。导出的 IDF 文件可以在免费的机械 CAD 软件 **FreeCAD** 中查看, 或使用 idf2vrml 工具转换为 VRML, 并可以使用任何合适的 VRML 查看器查看。

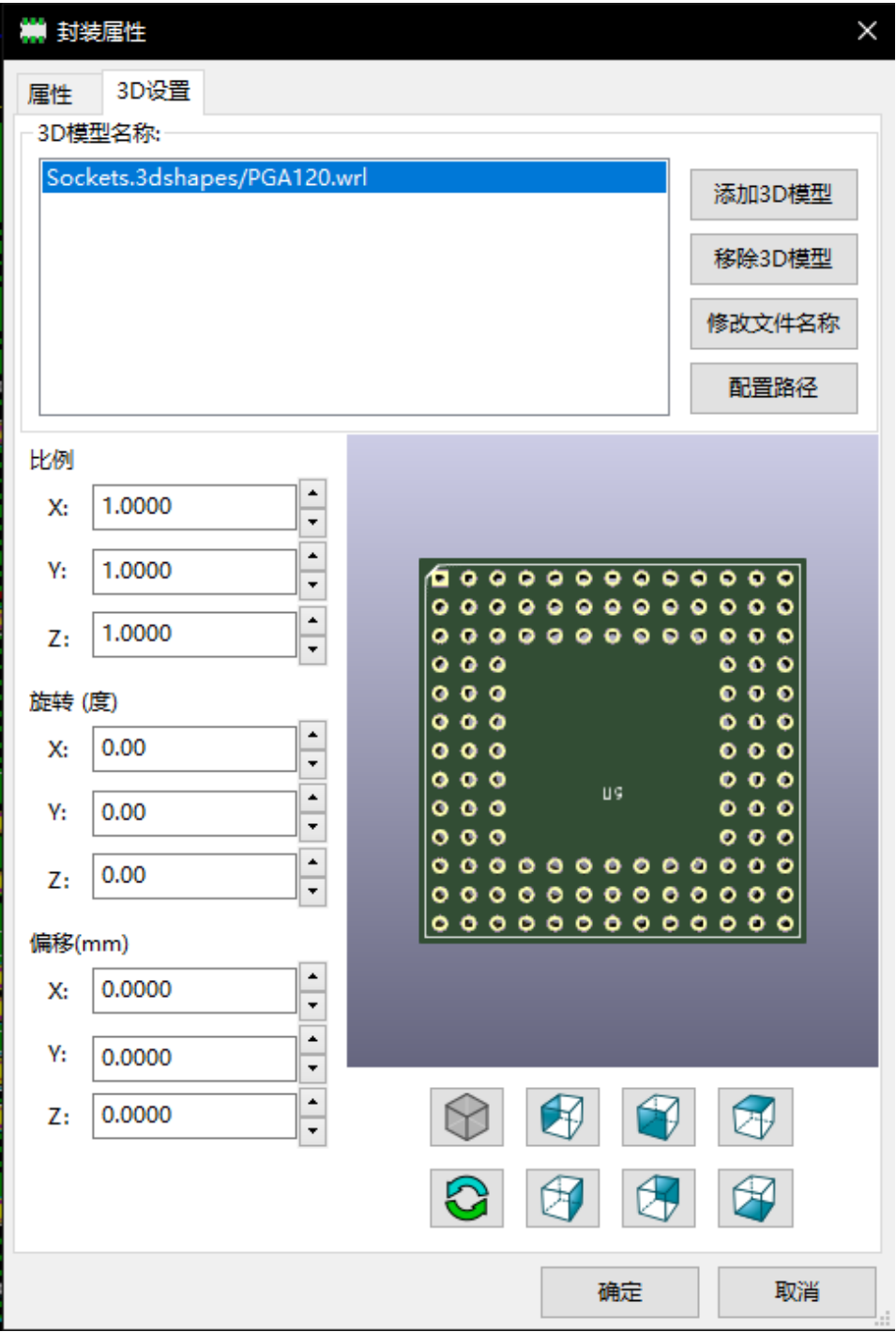


Figure 1: 封装属性, 3D 设置

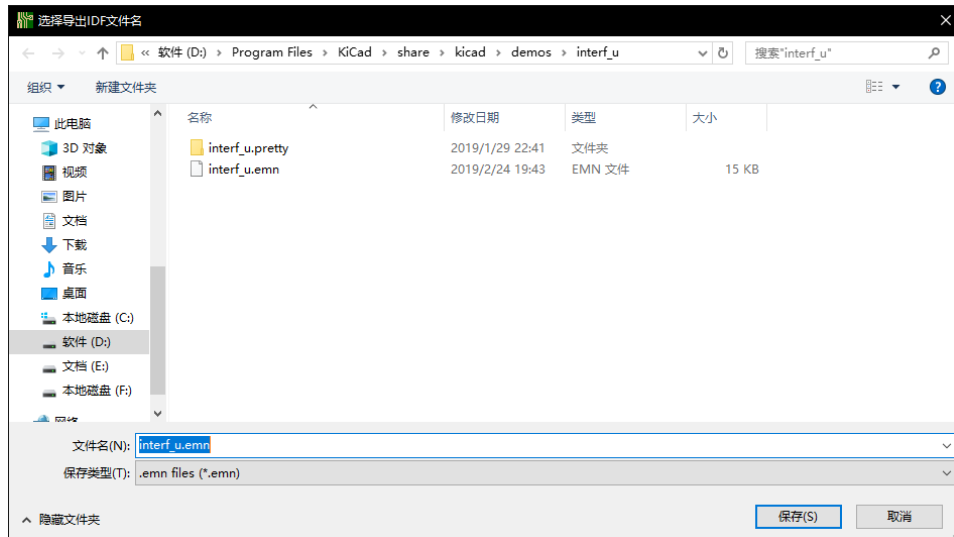


Figure 2: IDF 元件边框选择

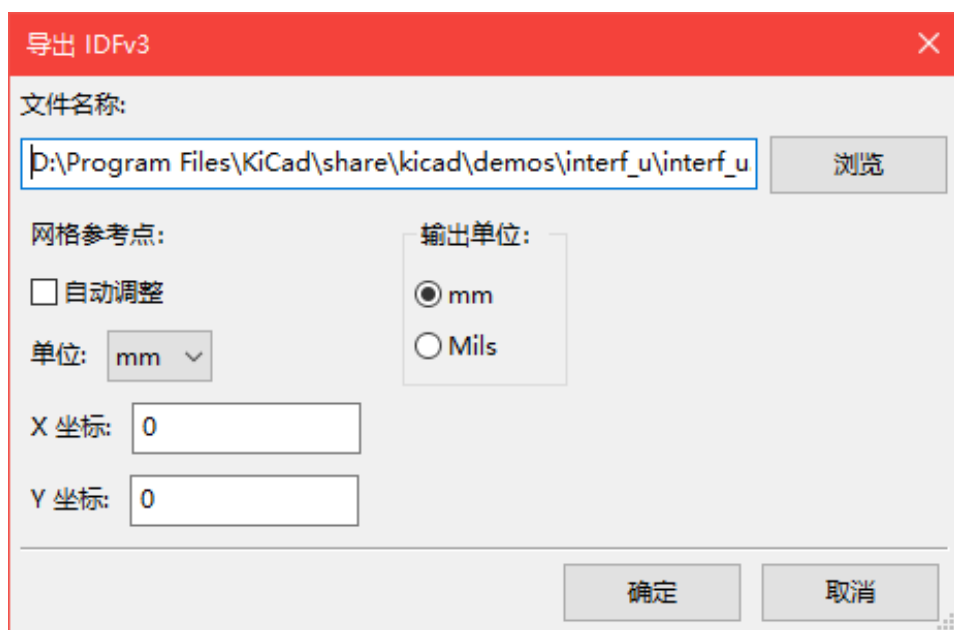


Figure 3: IDF 输出设置

3 创建元件边框文件

元件边框文件 (*.idf) 由规范文档中描述的单个.ELECTRICAL 或.MECHANICAL 部分组成。该部分之前可以有任意数量的注释行；导出器将注释行复制到库文件中，并可用于跟踪元数据，例如对用于确定元件边框和尺寸的文档的引用。

元件边框部分包含字符串，整数或浮点数字段。字符串是可包含空格的字符组合；如果字符串包含空格，则必须引用它。引号不得出现在字符串中。浮点数可以使用十进制或指数表示法表示，但十进制表示法是人类可读性的首选。小

数点必须是点而不是逗号。IDF 文件必须只包含 7 位 ASCII 字符; 使用 8 位字符将导致未定义的行为。

IDF 文件由 SECTIONS 组成, 其中包括由 FIELDS 组成的 RECORDS。对于 IDF 边框文件, 可能只存在一种类型的部分, 必须是.ELECTRICAL 或.MECHANICAL 之一。记录是单行文本, 可能包含一个或多个字段。字段是由一个或多个空格分隔的字符序列, 这些空格不出现在引号之间。记录的所有字段必须出现在一行中; 记录可能不会跨越行。

标题部分 (.ELECTRICAL 或.MECHANICAL) 被认为是该部分的第一个记录 (记录 1)。记录 1 必须跟随记录 2, 记录 2 有四个字段:

1. 几何名称: 与零件号组合的字符串必须形成元件边框的唯一标识符。对于标准化包, 包名称是几何名称的良好值, 例如 *SOT-23*。对于独特的包装, 制造商的部件号是几何名称的不错选择。
2. 部件号: 虽然显然是用于部件号, 例如 BS107, 但最好使用此字符串来帮助描述包。例如, 如果几何名称是 *TO-92*, 则部件号条目可用于描述焊盘的布局或该特定 TO-92 边框文件的方向。
3. IDF 单位: 这必须是 MM 或 THOU 之一, 它仅适用于描述此单一元件边框的单位。
4. 高度: 这是一个浮点数, 表示使用字段 3 中指定的单位的元件的标称高度。

记录 2 后面必须跟有许多记录 3 条目, 这些条目指定了元件的边框。记录 3 包含四个字段:

1. 循环索引: 0 (边框点以逆时针顺序指定) 或 1 (边框点以顺时针顺序指定)
2. X 坐标: 浮点数
3. Y 坐标: 浮点数
4. 包含的角度: 浮点数。如果该值为 0, 则从前一点到此点绘制直线段。如果值为 360, 则前一个点指定圆的中心, 此点指定圆上的点; 从不使用 -360 的值指定圆, 因为至少有一个主要的机械 CAD 软件包在这种情况下表现不佳。如果该值为负, 则从前一点到此点绘制顺时针弧, 如果该值为正, 则绘制逆时针弧。

只允许一个闭环, 并且无法指定切口。指定的最后一个点必须与第一个点相同, 除非边框是圆形。

示例 IDF 文件 1:

```
# a simple cylinder - this could represent an electrolytic capacitor
.ELECTRICAL
    "cylinder" "5mm OD, 5mm height" MM 5
    0 0 0 0
    0 2.5 0 360
.END_ELECTRICAL
```

示例 IDF 文件 2:

```
# an upside-down T
# a comment added for the sake of adding comments
.ELECTRICAL
    "Capital T" "5x8x10mm, upside down" MM 10
    0 -0.5 8 0
    0 -0.5 0.5 0
    0 -2.5 0.5 0
    0 -2.5 -0.5 180
```



```
0 2.5 -0.5 0
0 2.5 0.5 180
0 0.5 0.5 0
0 0.5 8 0
0 -0.5 8 180
.END_ELECTRICAL
```

4 创建边框的准则

在创建边框时，特别是在与他人共享工作时，文件的设计和命名的一致性可以帮助人们更快地找到文件并以最小的麻烦放置元件。

4.1 包命名

尝试提供有关文件名中可用边框的一些信息，以使用户大致了解边框是什么。例如，轴向引线圆柱形封装可能代表某些类型的电容器以及某些类型的电阻，因此将边框识别为水平或垂直轴向引线装置并在相关尺寸上添加一些额外信息是有意义的：直径，长度和音高是最重要的。如果设备具有唯一的边框，则制造商的部件号和用于指示设备类别的前缀就足够了。

4.2 注释

使用 IDF 文件中的注释为用户提供有关边框的更多信息，例如对用于尺寸信息的源的引用。

4.3 几何和部件编号条目

仔细考虑要赋予几何和部件编号条目的值。总之，这些字符串充当 MCAD 系统的唯一标识符。理想情况下，字符串的值对用户有一定的意义，但这不是必需的：这些值主要用于 MCAD 系统用作唯一 ID。理想情况下，所选择的值在任何大型边框集中都是唯一的；选择好的价值将导致更少的冲突，特别是在复杂的板上。

4.4 引脚方向和定位

对于通孔元件，没有广泛接受的方案来确定 3D 模型中的引脚方向和元件中心。为了保持一致性，如果只有 2 个引脚，它们必须沿 X 轴水平排列（见链接：[# 图-4\[图-4\]](#)），对于 3 个引脚，试图在 X 轴上保持水平排列 2。电解电容或钽电容等极化器件必须在引脚 1 上具有正极引线，二极管必须在引脚 1 上具有正极；这是为了保持原理图符号与 SMT 器件定义的方向的兼容性；但请注意，许多现有的 KiCad 原理图和封装都将阳极置于引脚 1。

Note

在 GitHub 上最新版本的 KiCad 封装中，阳极现在是针对 THT 的引脚 2 以及 SMT 元件。

对于 DIP 设备，边框的中心必须位于由销位置描述的矩形的中心，而引脚 1 最好位于左上角；这将与 SMT 元件的标准化方向保持一致；然而，相对于大多数现有的 KiCad 元件封装和 VRML 模型，这样的模型将旋转 -90 度。对于诸

如水平径向引线电容器或水平 TO-220 封装的器件，更喜欢将引线排成 X 行的一排，并且器件的主体向上延伸（参见链接：[# 图 4\[图 4\]](#)）。非极化垂直轴向引线元件必须在右侧有导线；极化垂直轴向引线元件可以在两侧都有导线，这取决于引脚 1 是在下端（导线在右侧）还是在上端（导线在左侧）。

Note

在当前版本的 KiCad 封装模块中，THT 元件的引脚沿 Y 轴而非 X 轴进行组织，器件的引脚 1 位于封装的原点而不是封装的中心。定位和定位元件边框以适合您的特定尺寸；这将避免为 IDF 元件边框指定非零旋转的需要。由于 IDF 导出器当前忽略 (X, Y) 偏移值，因此在 IDF 元件边框中使用正确的原点至关重要。

对于 SMT 元件，方向，封装中心和边框由各种标准定义。使用适合您工作的标准。还要记住，许多设备不符合任何标准；在这种情况下，可能最好通过在大纲文件名中使用制造商的部件号来识别违规设备。通常，SMT 边框是包含元件封装并包括引线的矩形；封装的方向使得引脚 1 尽可能靠近左上角，左上角通常被倒角以供视觉参考。

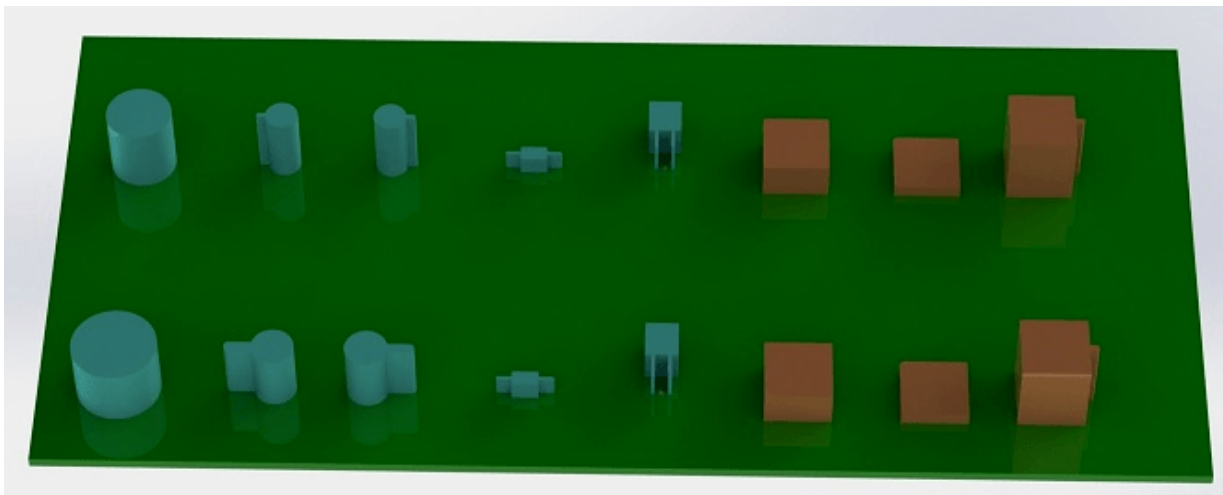


Figure 4: 程序生成的示例边框 idfcyl 和 idfrect 由 SolidWorks 生成并呈现。

从左到右分别是 (a) 垂直径向引线圆柱体，(b) 左边带有导线的垂直轴向引线圆柱体，(c) 带有导线的垂直轴向引线圆柱体，(d) 水平轴向引线圆柱体，(e) 水平径向引线圆柱，(f) 方形边框，平面，(g) 带倒角的方形边框，(h) 右侧轴向引线的方形边框。顶部边框以毫米为单位指定，而底部边框以英寸为单位指定。

4.5 尺寸提示

推挤边框的目的是让机械设计者了解每个部件占据的位置和物理空间。在典型情况下，机械设计师将使用更详细的机械模型替换一些原始边框，例如在检查时确保直角安装的 LED 适合面板上的孔。在大多数情况下，边框的准确性无关紧要，但优良作法是创建可传达最佳机械信息的边框。在少数情况下，用户可能希望将元件装配到具有非常小的多余空间的壳体中，例如在便携式音乐播放器中。在这种情况下，如果大多数推挤边框是元件的足够好的表示，那么机械设计师可能只需要在设计壳体时替换很少的模型。如果边框不是现实的可靠反映，那么机械设计师将浪费大量时间来更换模型以确保良好的贴合性。毕竟，如果你把垃圾放进去，你可以期待垃圾出来。如果您提供了良好的信息，您可以对良好的结果充满信心。

5 IDF 元件边框工具

许多命令行工具可用于帮助生成 IDF 元件边框。工具是：

1. **idfcyl**：创建垂直或水平圆柱的边框定向和轴向或径向引线
2. **idfrect**：创建一个矩形的边框，可能有任何一个左上角的轴向引线或倒角
3. **dxftoidf**：将 DXF 格式的图形转换为 IDF 元件边框

5.1 idfcyl

当 **idfcyl** 在没有参数的情况下被调用时，它会打印出一个使用说明及其输入的摘要：

idfcyl：该程序生成圆柱形元件的边框。

圆筒可以是水平的或垂直的。

水平圆柱体可以在一端或两端具有线。

垂直圆柱体可以具有至多一根可以是线
放在左侧或右侧。

输入：

单位：**mm**, **in**（毫米或英寸）

方向：**V**（垂直）

引线类型：**X**, **R**（轴向，径向）

体型直径

体型长度

板偏移

* 线径

* 节距

** 线侧：**L**, **R**（左，右）

*** 引线长度

文件名（必须以 * **.idf** 结尾）

注意：

* 仅用于水平方向或
带轴向引线的垂直方向

** 只有轴向引线的垂直方向才需要

*** 仅对于带有径向引线的水平方向需要

可以通过在命令行上输入任意参数来抑制注释。用户可以在命令行手动输入信息或创建脚本以生成边框。以下脚本创建一个单个圆柱轴向引线边框，右侧为引线：

```
#!/bin/bash
# 生成圆柱形 IDF 轮廓以用于测试目的
# 垂直 5mm 圆柱体，标称长度 8mm + 3mm 板偏移，
# 右侧轴向导线，直径 0.8mm，间距 3.5mm
```

```
idfcyl - 1 > /dev/null << _EOF
mm
v
x
5
8
3
0.8
3.5
r
cylvmm_1R_D5_L8_Z3_WD0.8_P3.5.idf
_EOF
```

5.2 idfrect

如果在没有参数的情况下调用 **idfrect**，则会打印出使用说明及其输入摘要：

```
idfrect: 该程序生成矩形元件的边框。
    该部件可以具有单个引线（轴向）或倒角
    左上角。
输入：
    单位：毫米，英寸（毫米或英寸）
    宽度：
    长度：
    高度：
    倒角：长度为 45 度。倒棱
    * 倾斜的角度：Y, N（引线始终向右边）
    ** 线径
    ** 球场
    文件名（必须以 *.idf 结尾）

注意：
    * 仅在 倒角 = 0 时才需要

    ** 仅对倾斜的角度元件有要求
```

可以通过在命令行上输入任意参数来抑制注释。用户可以在命令行手动输入信息或创建脚本以生成边框。以下脚本创建倒角矩形和轴向引线边框：

```
#!/bin/bash
# 生成各种矩形 IDF 轮廓以用于测试目的
# 10x10, 1mm 倒角, 2mm 高度
idfrect - 1 > /dev/null << _EOF
mm
10
10
2
1
```

```
rectMM_10x10x2_C0.5.idf
_EOF
# 6x10x12, 0.8mm 引脚, 6mm 间距
idfrect - 1 > /dev/null << _EOF
mm
10
10
12
0
Y
0.8
6
rectLMM_10x10x12_D0.8_P6.0.idf
_EOF
```

5.3 dxf2idf

用于指定元件概要的 DXF 文件可以使用免费软件 [LibreCAD](#) 进行准备，以获得最佳兼容性。如果在没有参数的情况下调用 **dxf2idf**，则会打印出使用说明及其输入摘要：

dxf2idf：此程序采用线段，圆弧和圆弧段
从 DXF 文件中创建 IDF 元件边框文件。

输入：

DXF 文件名：输入文件，必须以 `_.dxf_` 结尾
单位：毫米，英寸（毫米或英寸）
几何名称：字符串，根据 IDF 3.0版规范
部件名称：根据部件号的 IDF 3.0版规范
高度：边框的挤压高度
注释：所有非空行都是要添加的注释
IDF 文件。空行表示结束
注释块。
文件名：输出文件名，必须以 `_.idf_` 结尾

可以通过在命令行上输入任意参数来抑制注释。用户可以在命令行手动输入信息或创建脚本以生成边框。以下脚本从 DXF 文件 *test.dxf* 创建 5mm 高边框：

```
#!/bin/bash
# 从 DXF 文件生成 IDF 轮廓
dxf2idf - 1 > /dev/null << _EOF
test.dxf
mm
DXF TEST GEOMETRY
DXF TEST PART
5
这是从轮廓 'test.dxf' 生成的 IDF 测试文件
这是 IDF 的第二条注释，用于演示多条注释
```

```
test_dxf2idf.idf
_EOF
```

6 idf2vrm1

这个 idf2vrm1 工具读取一组 IDF 板 (.emn) 和一个 IDF 元件文件 (.emp) 并生成 VRML 文件；可以使用 VRML 查看器查看该文件。在用户无法访问 MCAD 软件的情况下此功能对于电路板装配的可视化非常有用；调用不带任何参数的 idf2vrm1 将导致显示用法消息：

```
>./ idf2vrm1
用法: idf2vrm1 -f input_file.emn -s scale_factor {-k} {-d} {-z} {-m}
标志:
    -k: 生成 KiCad 友好的 VRML 输出；默认是紧凑的 VRML
    -d: 禁止替换默认边框
    -z: 禁止渲染零高度边框
    -m: 打印对象映射到 stdout 以进行调试
生成 KiCad 使用的模型的示例: idf2vrm1 -f input.emn -s 0.3937008 -k
>
```

[**BUG:** 如果在 PCB 的背面指定了该实体, 则 idf2vrm1 工具当前无法在 emn 文件中正确呈现 **OTHER_OUTLINE** 实体; 但是你不会使用 KiCad 导出的文件注意到这个错误, 因为没有机制来指定这样的实体。基本上这个错误只是一个问题, 在极少数情况下, 您可能会渲染第三方的 emn 文件, 该文件确实使用了电路板背面的实体。]