



# kicad



# kicad

**IDF Exporter**

**March 2, 2016**

## Contents

<b>1</b>	<b>IDF Exporter について</b>	<b>2</b>
<b>2</b>	<b>IDF Exporter で使用するコンポーネントモデルの指定</b>	<b>2</b>
<b>3</b>	<b>コンポーネント・アウトライン・ファイルの生成</b>	<b>4</b>
<b>4</b>	<b>アウトライン作成のためのガイドライン</b>	<b>6</b>
4.1	パッケージの命名 . . . . .	6
4.2	コメント . . . . .	6
4.3	ジオメトリと部品番号の入力 . . . . .	7
4.4	ピンの向きと配置 . . . . .	7
4.5	寸法についての Tips . . . . .	8
<b>5</b>	<b>IDF コンポーネント・アウトライン・ツール</b>	<b>8</b>
5.1	idfcyl . . . . .	9
5.2	idfrect . . . . .	10
5.3	dxidf . . . . .	11
<b>6</b>	<b>idf2vrml</b>	<b>12</b>

## リファレンスマニュアル

### 著作権

このドキュメントは以下の貢献者により著作権所有 © 2014-2015 されています。あなたは、GNU General Public License ( <http://www.gnu.org/licenses/gpl.html> ) のバージョン 3 以降、あるいはクリエイティブ・コモンズ・ライセンス ( <http://creativecommons.org/licenses/by/3.0/> ) のバージョン 3.0 以降のいずれかの条件の下で、配布または変更することができます。

このガイドの中のすべての商標は、正当な所有者に帰属します。

#### \* 貢献者 \*

Cirilo Bernardo

### 翻訳

starfort <starfort AT nifty.com>, 2015. Norio Suzuki <nosuzuki AT postcard.st>, 2015. yoneken <yoneken AT kicad.jp>, 2011-2015.

### フィードバック

バグ報告や提案はこちらへお知らせください:

- KiCad のドキュメントについて: <https://github.com/KiCad/kicad-doc/issues>
- KiCad ソフトウェアについて: <https://bugs.launchpad.net/kicad>
- KiCad ソフトウェアの国際化について: <https://github.com/KiCad/kicad-i18n/issues>

### 発行日とバージョン



2014 年 1 月 26 日発行


## 1 IDF Exporter について

IDF Exporter は、機械系 CAD へ寸法情報を伝えるため、IDF V3.0 付加情報:[\[http://www.simplifiedsolutionsinc.com/-images/idf\\_v30\\_spec.pdf\]](http://www.simplifiedsolutionsinc.com/-images/idf_v30_spec.pdf) に準拠した基板 (.emn) とライブラリ (.emp) のファイルを出力します。IDF Exporter は今のところ、基板アウトラインと切削部、パッドとランド（細長い穴のものを含む）、及びコンポーネント・アウトラインに関する情報を出力します。これは、機構設計者とやりとりする必要がある機械的情報の基本部分のほとんどです。IDF V3.0 仕様に記載された他の要素は現在出力されません。

## 2 IDF Exporter で使用するコンポーネントモデルの指定

IDF Exporter は、元々 3D ビューアで使われている 3D モデルファイルのアトリビュートを使用します。3D ビューア、IDF、将来的な機械系 CAD エクスポートは、別の様々な種類のファイルフォーマットにも広く関心を払っており、多くのエクスポートではモデルを指定することで 3D モデルファイルのアトリビュートを使用できるようにしています。

フットプリントエディタや Pcbnew から、フットプリントのパラメータを編集し、3D 設定タブ上でクリック（ 1. 参照）、3D シェイプの追加をクリックし、“IDFv3 コンポーネントファイル (\*.idf)” のフィルタを選択します（ 2. 参照）。希望するアウトラインファイルを選択し、オフセットと回転に必要な値を入力します。オフセット値と Z 軸の回転角のみが IDF Exporter で使われることに注意して下さい。；他の全ての値は無視されます。オフセットは IDF ファイルの書き出しに使用される単位（mm または thou(=mils)）で指定しなければなりません。また IDF の座標系は右手系で、Z 軸が観測者へ向かって正、X 軸が観測者の右へ向かって正、Y 軸が上へ向かって正です。回転角は度で表さなければならず、正の回転方向は IDF V3.0 仕様に記載通り左回りです。複数のアウトラインは、ソケットに挿入された DIP パッケージのような単純な組み立て品を表すために、適切なオフセットを持って結合されることがあります。[バグ情報: 議論を重ねた中で、VRML モデルとの整合性を保つため、Z オフセットはインチ単位でなければならないと決定されました。また、(X,Y) のオフセット値を無視するのは実用的ではないかも知れません。ここで述べた挙動は将来的に変更されるでしょう。]

モデルに必要な全てのコンポーネントを指定した後、Pcbnew から ファイルメニューを選択して エクスポートを選択し、最終的に **IDFv3 Export** します。IDF 出力の単位（mm または mils）と出力ファイル名を指定するダイアログボックスがポップアップ（ 3. 参照）します。エクスポートされた IDF ファイルは無償の機械系 CAD ソフトウェア **FreeCAD** で参照できます。もしくは idf2vrml ツールで VRML 形式へ変換して適切な VRML ビューアでも参照できます。

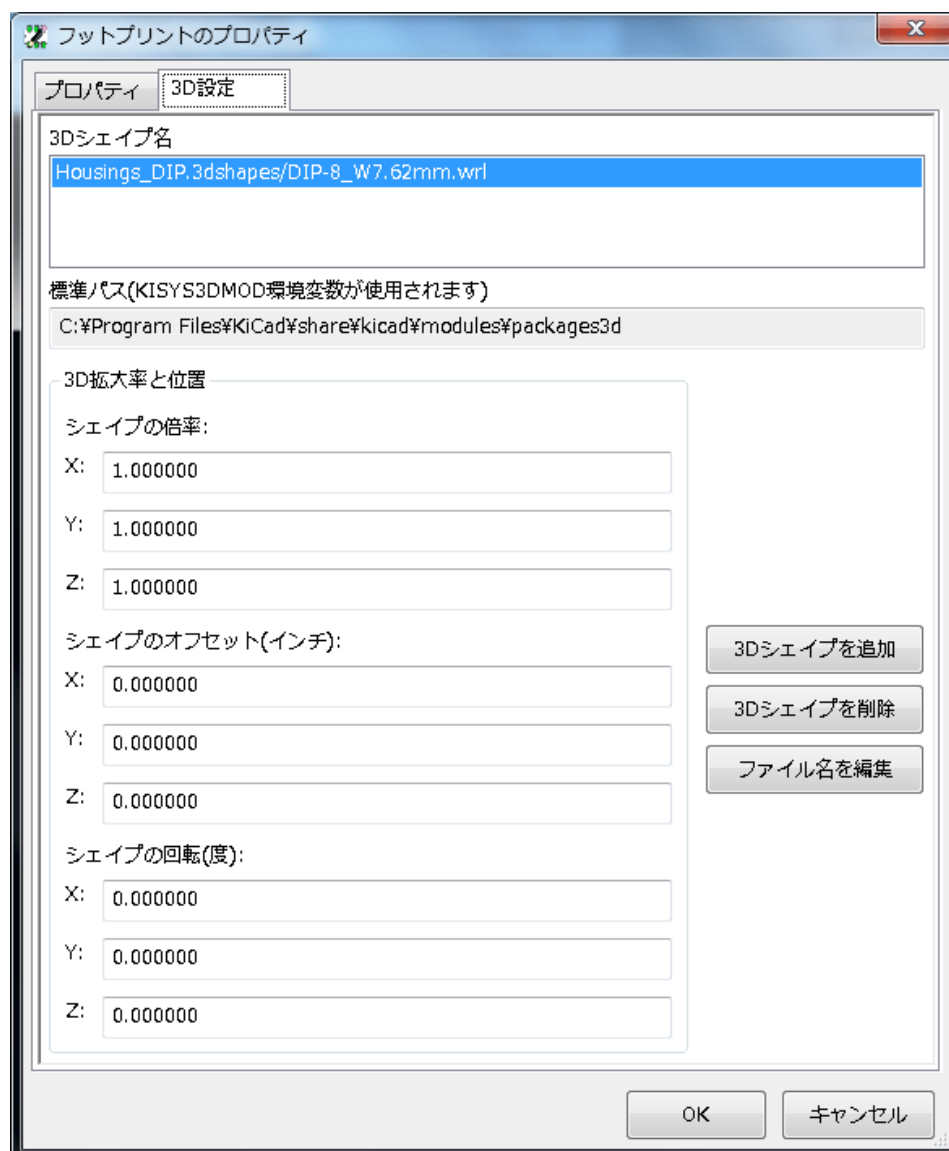


Figure 1: フットプリントのプロパティ、3D 設定

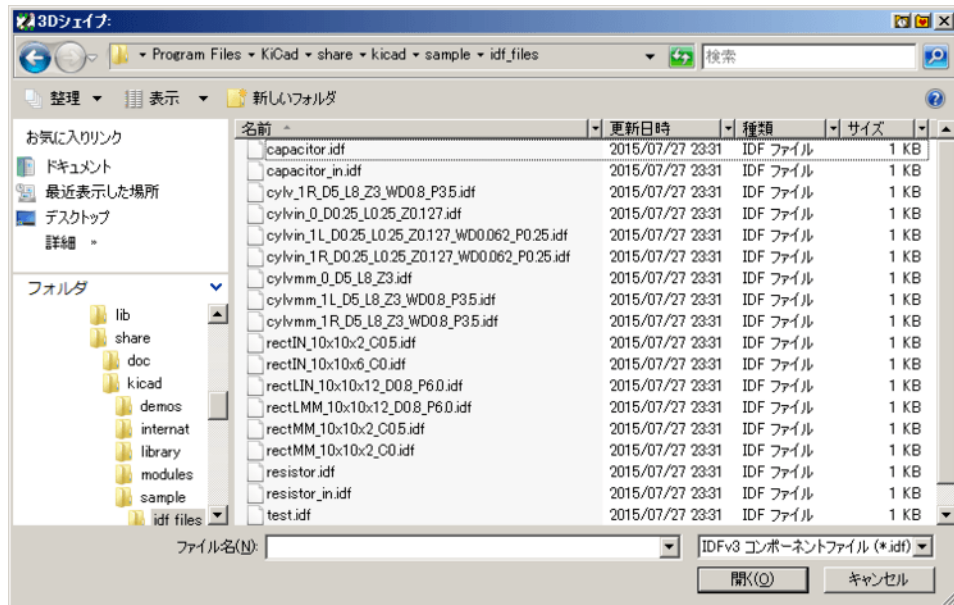


Figure 2: IDF コンポーネント・アウトラインの選択

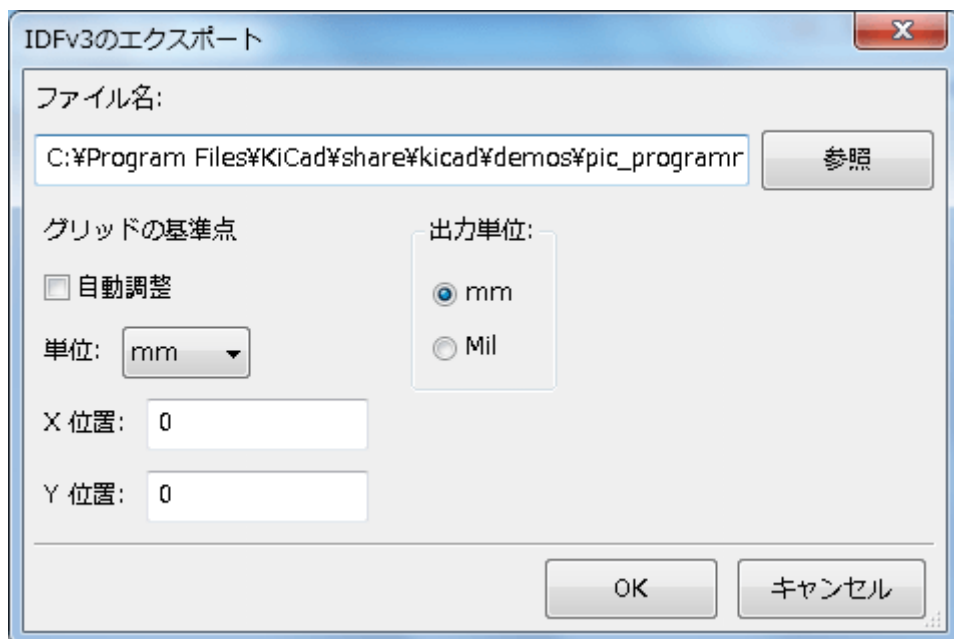


Figure 3: IDFv3 のエクスポートの設定

### 3 コンポーネント・アウトライン・ファイルの生成

コンポーネント・アウトライン・ファイル (\*.idf) は、仕様書に記載されているように単一のELECTRICAL またはMECHANICAL セクションで構成されます。セクションは行数に拘らずコメント行を前に置くことができます。; コメント行は IDF Exporter によってライブラリファイルへコピーされ、コンポーネント・アウトラインと寸法の決定に使われたドキュメントの参照事項のようなメタデータを追跡するために使用できます。

コンポーネント・アウトラインのセクションは、複数の文字列、整数、浮動小数点数のフィールドを含みます。一つの文字列は複数の文字（空白を含む）の組み合わせから成ります。; もし文字列が空白を含む場合は引用符で囲まなければなりません。引用符を文字列の中に入れることはできません。浮動小数点数は十進記数法または指数表記で表すことができますが、可読性から十進記数法を使うことが好ましいでしょう。小数点はカンマ (,) ではなく、ドット (.) でなければなりません。IDF ファイルは 7-bit アスキー文字のみで構成しなければなりません。; 8-bit 文字を使用した場合の挙動は不定となります。

IDF ファイルは、FIELDS からなる RECORDS で構成された SECTIONS で構成されます。IDF アウトラインファイルにはただ一つのセクションタイプだけが存在し、.ELECTRICAL か.MECHANICAL. のどちらかでなければなりません。一つのレコードは単一行のテキストで、一つ以上のフィールドからなります。フィールドは、引用符に囲まれていない一つ以上の空白で区切られた文字列が連続したものです。一つのレコードに含まれる全てのフィールドは単一行でなければなりません。; レコードは複数行に渡ってはいけません。

セクションの表題 (.ELECTRICAL または.MECHANICAL) はセクションの最初の RECORD (Record 1) となるよう考慮して下さい。Record 1 には、4つのフィールドから成る Record 2 が続かなければなりません。

1. ジオメトリ名: 部品番号と結合した文字列はコンポーネント・アウトラインを表す固有識別子となるべきです。標準化されたパッケージでは、ジオメトリ名としてパッケージ名が適切です。(例 “SOT-23”) カスタム品のパッケージには、製造業者の品番がジオメトリ名として良い選択となります。
2. 部品番号: 明らかに部品番号を意図したもの (例 BS107) だとしても、この文字列をパッケージを表すために使ったほうが良いでしょう。例えば、ジオメトリ名が “TO-92” なら、部品番号の記載はパッドのレイアウトや他の TO-92 と異なる向きを示すために使えるかも知れません。
3. IDF 単位: mm または thou(=mils) としなければなりません。また、これは単独のコンポーネント・アウトラインを記述する単位としてのみ適用されます。
4. 高さ: Field 3 で指定された単位を用いた、コンポーネントの図面上の高さを表す浮動小数点数です。

Record 2 には、コンポーネントのアウトラインを指定する Record 3 が続かなければなりません。Record 3 は4つのフィールドからなります。

1. ループインデックス: 0 (外形点は左回り順で指定されます) または 1 (外形点は右回り順で指定されます)
2. X 座標: 浮動小数点数です。
3. Y 座標: 浮動小数点数です。
4. 夾角: 浮動小数点数です。もし値が 0 ならば、一つ前の点からこの点へと描かれたセグメントは直線です。もし値が 360 ならば、一つ前の点が円の中心を指定し、この点は円の点を指定しています。; 少なくとも一つのメジャーな機械系 CAD が対応していないので、値に -360 を用いて円を指定してはいけません。もし値が負ならば一つ前の点からこの点へ右回りの弧が描かれ、値が正ならば左回りの弧が描かれます。

ただ一つの閉じたループのみが許容され、切削部を指定することはできません。アウトラインが円である場合を除き、最後の点は最初の点と等しくなければなりません。

例 IDF ファイル 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
```

例 IDF ファイル 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 アウトライン作成のためのガイドライン

アウトラインを作成するとき（特に複数で作業するような場合）、ファイルの名前と設計における一貫性は、素早くファイルを見つけたり、最少の努力で部品を配置するために役立ちます。

### 4.1 パッケージの命名

ユーザーがアウトラインの概要を把握できるよう、ファイル名にアウトラインについての情報を付加するように努めるべきです。例えば、円筒形のアキシャルリード・パッケージは抵抗だけではなくコンデンサにも使われるので、アキシャルリード部品のアウトラインが横向きなのか若しくは縦向きなのかを識別することは意味があります。また関連する寸法についていくつかの情報（：直径、長さ、ピッチ）を付加することは最も重要です。もし部品がカスタム品で特別なアウトラインを持つような場合、製造業者の品番や種類を表すプレフィックスを使うことが適切です。

### 4.2 コメント

ユーザーにアウトラインについての関連情報（例えば、寸法情報の出典に関する参照事項）を提供するためには IDF ファイル内でコメントを使用します。

### 4.3 ジオメトリと部品番号の入力

ジオメトリと部品番号へ入力する値は慎重に考慮しましょう。まとめて扱うべきです。これらの文字列は機械系 CAD に対して固有識別子として働きます。文字列の値はユーザーにとって意味があることが理想的ですが、必須ではありません。: 値は主として機械系 CAD で固有識別子として使用するためのものです。理想的には、選ばれた値はいかなる多くのアウトラインに対しても固有のものとなるでしょう。; 十分に選ばれた値は、特に複数の基板においても衝突を少なく抑えることにつながります。

### 4.4 ピンの向きと配置

スルーホール・コンポーネントに対しては、3D モデルの中心とピンの向きを決めるために図面を使うことは一般的ではありません。一貫性のため、X 軸に沿って水平に配置 (参照 [figure 4](#)) された 2 つのピンがあれば、3 ピンに対しては X 軸上に水平配置された 2 つのピンを保つようにします。電解コンデンサやタンタルコンデンサのような有極性部品は正極のリードを 1 ピンにする必要があります。またダイオードは 1 ピンをカソードとします。; これは表面実装部品 (SMT デバイス) で定義された方向を持つ回路記号と互換性を保つためです。; しかしながら、多くの既存の KiCad 図面とフットプリントが 1 ピンをアノードとしていることに注意してください。

---

#### 注意

github 上にある最新版の KiCad フットプリントでは、スルーホール・コンポーネントは表面実装部品 (SMT デバイス) 同様に今ではアノードが 2 ピンです。

---

DIP 部品に対しては、アウトラインの中心はピンの位置によって描かれる矩形の中心でなければなりません。また 1 ピンは左上のコーナーとすることが望ましいです。; これは表面実装部品 (SMT デバイス) の標準的な向きとある程度の一貫性を保つでしょう。; しかしながら、このようなモデルは多くの既存の KiCad フットプリントと VRML モデルに対して相対的に -90 度回転するでしょう。横向きのラジアルリード・コンデンサや横向きの TO-220 パッケージに対しては、デバイス本体の上向きに延長された X 軸上の列にリードを配置することが好ましいです ([図 4](#), 参照)。無極性の縦向きアキシャルリード・コンポーネントは右手側に配線されなければなりません。; 有極性の縦向きアキシャルリード・コンポーネントはピン 1 が下端 (右側へ配線) か上端 (左側へ配線) かによって、どちら側へも配線されることがあり得ます。

---

#### 注意

現在のバージョンの KiCad フットプリントでは、スルーホール・コンポーネントは X 軸よりむしろ Y 軸に沿ってピンが編成されており、パッケージの中央よりは 1 ピンにデバイスの原点があります。配置するフットプリントに合わせてコンポーネント・アウトラインの位置と向きを合わせてください。これで、IDF コンポーネント・アウトラインの回転を指定する必要がなくなるでしょう。IDF Exporter は今のところ (X,Y) のオフセット値を無視するので、IDF コンポーネント・アウトラインの原点を正しく扱うことは大変重要です。

---

表面実装部品 (SMT デバイス) に対しては、向き、パッケージの中心、アウトラインは様々な規格で定義されています。あなたの業務に適した規格を使用してください。また多くの部品がいかなる規格にも適合しないことを覚えておきましょう。; このような場合、規格に従わない部品はアウトラインファイルの名前に製造業者の品番を用いることでたぶん最もよく識別されます。一般的には、SMT アウトラインはコンポーネント・パッケージとリードを含む部分を囲むような矩形です。; パッケージはピン 1 が左上のコーナーに最も近くなるような向きとなります。また左上のコーナーは通常、視覚的にはっきりさせるため面取り (角を削る) されます。

---

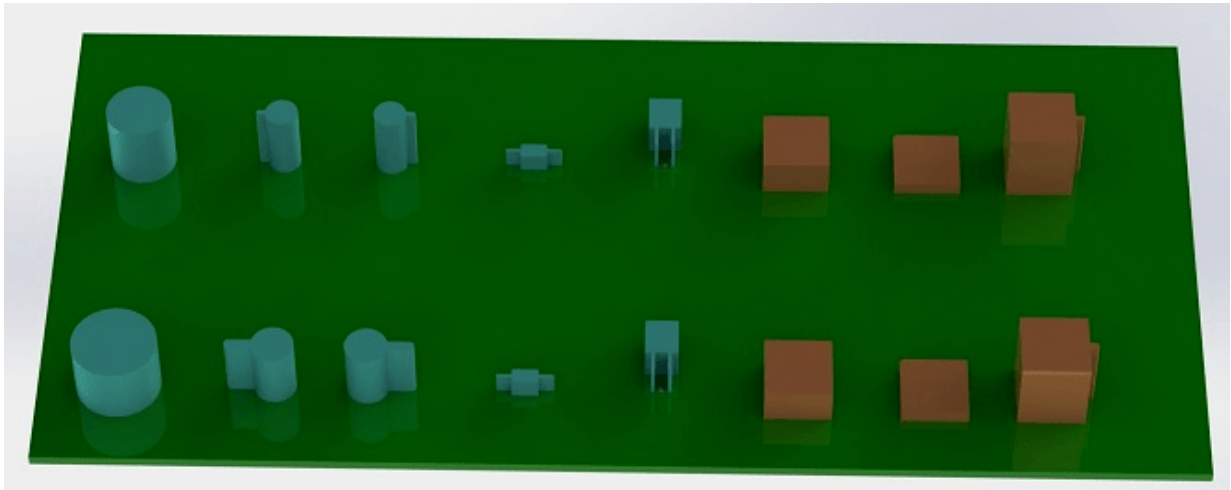


Figure 4: idfcyl と idfrect ツールで生成されたアウトラインのサンプルと SolidWorks によるレンダリング

左から右へ (a) 縦向きラジアルリード円筒部品, (b) 左側配線の縦向きアキシャルリード円筒部品, (c) 右側配線の縦向きアキシャルリード円筒部品, (d) 横向きアキシャルリード円筒部品, (e) 横向きラジアルリード円筒部品, (f) 正方形アウトライン, 平面, (g) 面取りされた正方形アウトライン, (h) 右にアキシャルリードを持つ正方形アウトライン。上のアウトラインは mm 単位で指定、下のアウトラインは inch 単位で指定。

#### 4.5 寸法についての Tips

突き出たアウトラインを使う目的は、機構設計者に各部品が占有する物理的な空間とその位置を認識させることです。典型的なシナリオでは、機構設計者はより詳細な機械的モデルを持つ幾つかの加工していないアウトラインと入れ替えるでしょう。例えば直角に実装された LED がパネルの穴に確実に合うか確認する場合です。多くの場合、アウトラインの正確さは重要ではありませんが、可能な限り正確な機械的情報が伝わるようなアウトラインを作ることは良い慣行です。少ない例ですが、例えば携帯型音楽プレーヤーのように、ユーザーが余分の空間が殆どないケースへコンポーネントを収めることを望むこともあるでしょう。このような場合、最も突き出たアウトラインがコンポーネントの表現として十分ならば、機構設計者はケースを設計する時にほんの少数のモデルと入れ替えるだけで済むでしょう。もしアウトラインが実際の大きさを反映していると信頼できないなら、機構設計者は確実にケースに収めるため（機械的）モデルを置き換えることに多くの時間を取られるでしょう。結局のところ、あなたがゴミを入れたなら、出てくるものはゴミしか期待できません。もしあなたが適切な情報を与えたなら、良い結果を確信できます。

## 5 IDF コンポーネント・アウトライン・ツール

いくつかのコマンドラインツールが IDF コンポーネント・アウトラインの生成を助けます。ツールは次のものとなります：

1. **idfcyl**: アキシャルまたはラジアルのリードを持った縦または横向きになった円筒のアウトラインを作成します。
2. **idfrect**: アキシャルリードか面取りされた左上のコーナーを持つ矩形のアウトラインを作成します。

### 3. dxf2idf: DXF フォーマットの図面を IDF コンポーネント・アウトラインへと変換します

## 5.1 idfcyl

idfcyl は引数なしで呼び出された場合、入力の概要と使用方法を表示します。:

idfcyl: このプログラムは円筒形コンポーネントのアウトラインを作成します。

円筒は横向き又は縦向きです。

横向き円筒は片側または両側に配線できます。

縦向き円筒は左または右側のどちらかに配線できます。

Input:

単位: mm, in (mm または inch)

向き: v (縦向き)

リードの種類: X, R (アキシャル, ラジアル)

本体の直径

本体の長さ

基板のオフセット

\* 配線の直径

\* ピッチ

\*\* 配線面: L, R (左, 右)

\*\*\* リードの長さ

ファイル名 (\*.idf で終わらなければならない)

NOTES:

\* 横向き又は縦向きアキシャルリード  
でのみ必要です

\*\* 縦向きのアキシャルリードでのみ必要です

\*\*\* 横向きのラジアルリードでのみ必要です

NOTES はコマンドラインに任意の引数を入力することで非表示にできます。ユーザーはコマンドラインで情報を手入力するか、アウトラインを生成するスクリプトを作ることができます。次のスクリプトは右手側にリードを持った単独のアキシャルリード・アウトラインを作成します。:

```
#!/bin/bash
# Generate a cylindrical IDF outline for test purposes
# vertical 5mm cylinder, nominal length 8mm + 3mm board offset,
# axial wire on right, 0.8mm wire dia., 3.5mm pitch
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

**idfrect** は引数なしで呼び出された場合、入力の概要と使用方法を表示します。:

**idfrect**: このプログラムは矩形コンポーネントのアウトラインを作成します。

コンポーネントは一つのリード（アキシャル）または面取りされた  
左上のコーナーを持ちます。

Input:

単位: mm, in (mm または inch)  
幅  
長さ  
高さ  
面取り: 45度面取りの長さ  
\* リード: Y,N (リードは常に右)  
\*\* 配線の直径  
\*\* ピッチ  
ファイル名 (\*.idf で終わらなければならない)

NOTES:

\* 面取り = 0 の場合に必要  
  
\*\* リードコンポーネントでのみ必要

NOTES はコマンドラインに任意の引数を入力することで非表示にできます。ユーザーはコマンドラインで情報を手入力するか、アウトラインを生成するスクリプトを作ることができます。次のスクリプトは面取りされた矩形とアキシャルリードのアウトラインを作成します。:

```
#!/bin/bash
# Generate various rectangular IDF outlines for test purposes
# 10x10, 1mm chamfer, 2mm height
idfrect - 1 > /dev/null << _EOF
mm
10
10
2
1
rectMM_10x10x2_C0.5.idf
_EOF
# 10x10x12, 0.8mm lead on 6mm pitch
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

コンポーネントのアウトラインを指定する DXF ファイルは、高い互換性を持つフリーウェア [LibreCAD](#) で準備できます。`dxf2idf` は引数なしで呼び出された場合、入力の概要と使用方法を表示します。:

`dxf2idf`: このプログラムは DXF ファイルから線、弧、円のセグメントを取り出し、  
IDF コンポーネント・アウトライン・ファイルを作成します。

#### Input:

- DXF ファイル名: 入力ファイル, '.dxf' で終わらなければならない
- 単位: mm, in (mm または inch)
- ジオメトリ名: 文字列, IDF version 3.0 仕様に準拠
- パート番号: パート番号の IDF version 3.0 仕様に準拠
- 高さ: アウトラインの突き出た部分の高さ
- コメント: 空ではない全ての行は IDF ファイルへ追加されたコメントです  
空行はコメントブロックの終わりを示します  
ファイル名 (\*.idf で終わらなければならない)

NOTES はコマンドラインに任意の引数を入力することで非表示にできます。ユーザーはコマンドラインで情報を手入力するか、アウトラインを生成するスクリプトを作ることができます。次のスクリプトは DXF ファイル `test.dxf` から高さ 5 mm のアウトラインを作成します。:

```
#!/bin/bash
# Generate an IDF outlines from a DXF file
dxf2idf - 1 > /dev/null << _EOF
test.dxf
mm
DXF TEST GEOMETRY
DXF TEST PART
5
This is an IDF test file produced from the outline 'test.dxf'
This is a second IDF comment to demonstrate multiple comments

test_dxf2idf.idf
_EOF
```

## 6 idf2vrm1

idf2vrm1 ツールは一つの IDF 基板ファイル (*.emn*) と一つの IDF コンポーネントファイル (*.emp*) のセットを読み込み、VRML ビューアで見ることができる VRML ファイルを作成します。この機能は、ユーザーが機械系 CAD ソフトを使用できない場合、実装基板を可視化するのに有用です。idf2vrm1 は引数なしで呼び出された場合、使用方法を表示します。:

```
>./idf2vrm1
使用法: idf2vrm1 -f input_file.emn -s scale_factor {-k} {-d} {-z} {-m}
フラグ:
  -k: KiCad 向きの VRML 出力を作成する; デフォルトは compact VRML
  -d: デフォルトアウトラインの代替を非表示にする
  -z: 高さ0のアウトラインをレンダリングしない
  -m: デバッグ用に stdout ヘマップされたオブジェクトを表示する
KiCad で使用されるモデルを作成する例: idf2vrm1 -f input.emn -s 0.3937008 -k
>
```

[バグ情報: idf2vrm1 ツールは現在、基板の裏面となっている emn ファイル内の **OTHER\_OUTLINE** 部分を正しくレンダリングしません。; しかしながら、このような設定をする方法がないので KiCad がエクスポートしたファイルを使う場合、このバグに気付くことはないでしょう。基本的にこのバグは、(裏面に実体を持ったサードパーティ製の emn ファイルをレンダリングするような) 稀な例でのみ問題となります。]