



kicad



kicad

IDF Exporter

2 marzo 2016

Indice

1	Introduzione all' esportatore di IDFv3	2
2	Specifica dei modelli di componenti per l' uso da parte dell' esportatore	2
3	Creazione di un file contorno componente	4
4	Linee guida per la creazione dei contorni	6
4.1	Denominazione dei pacchetti	6
4.2	Commenti	6
4.3	Voci di geometria e numero di parte	7
4.4	Orientamento e posizionamento pin	7
4.5	Suggerimenti sulle dimensioni	8
5	IDF Component Outline Tools	8
5.1	idfcyl	9
5.2	idfrect	10
5.3	dx2idf	11
6	idf2vrml	12

Manuale di riferimento

Copyright

Questo documento è coperto dal Copyright © 2014-2015 dei suoi autori come elencati in seguito. È possibile distribuirlo e/o modificarlo nei termini sia della GNU General Public License (<http://www.gnu.org/licenses/gpl.html>), versione 3 o successive, che della Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), versione 3.0 o successive.

Tutti i marchi registrati all' interno di questa guida appartengono ai loro legittimi proprietari.

Collaboratori

Cirilo Bernardo

Traduzione

Marco Ciampa <ciampix@libero.it>, 2014-2015.

Feedback

Please direct any bug reports, suggestions or new versions to here:

- About KiCad document: <https://github.com/KiCad/kicad-doc/issues>
- About KiCad software: <https://bugs.launchpad.net/kicad>
- About KiCad software i18n: <https://github.com/KiCad/kicad-i18n/issues>

Data di pubblicazione e versione del software

Pubblicato il 26 gennaio, 2014.

1 Introduzione all' esportatore di IDFv3

L' IDF exporter, esporta file scheda (.emn) e libreria (.emp) IDFv3 conformi ¹ per trasmettere le dimensioni meccaniche per poter essere usate da un CAD meccanico. L' esportatore attualmente esporta i limiti e le specifiche di taglio scheda, tutte le piazzole e i fori passanti di montaggio inclusi i fori ad asola, e i contorni dei componenti; questo è l' insieme più basilare dei dati meccanici necessari per l' interazione con i progettisti meccanici. Tutte le altre entità di cui alla specifica IDFv3 non sono attualmente esportate.

2 Specifica dei modelli di componenti per l' uso da parte dell' esportatore

L' IDF exporter fa uso degli attributi dei file dei modelli 3D che erano originariamente usati dal visualizzatore 3D. Dal momento che il visualizzatore 3D, IDF, e i possibili futuri esportatori per CAD meccanici sono generalmente interessati a diversi tipi di formati di file, è possibile usare gli attributi dei file dei modelli 3D per specificare i modelli per più esportatori.

Dall' interno dell' editor delle impronte o da pcbnew, modificare i parametri dell' impronta e fare clic sulla scheda delle impostazioni 3D (vedere [figura 1](#)), fare clic su aggiungi forma 3D, e selezionare il filtro File componenti IDFv3 (*.idf) (vedere [figura 2](#)). Selezionare il file di contorno desiderato e inserire ogni valore necessario per lo spostamento e la rotazione. Si noti che solo i valori di spostamento e di rotazione Z vengono usati dall' esportatore IDF; tutti gli altri valori vengono ignorati. Gli spostamenti devono essere specificati usando le unità di uscita delle schede IDF (mm o mils) e nel sistema di coordinate IDF, che è un sistema di coordinate destrorso con l' asse +Z che si sposta verso l' osservatore, +X è alla destra dell' osservatore, e +Y è verso l' alto. La rotazione deve essere in gradi e una rotazione positiva è una rotazione antioraria come descritto nelle specifiche IDFv3. Più contorni possono essere combinati con spostamenti appositi per rappresentare semplici assemblamenti come un dispositivo in contenitore DIP in uno zoccolo. [BUG: è stato deciso che l' unità di misura dello spostamento Z debba essere in pollici, che è consistente con le unità di spostamento del modello VRML. Può tornare utile non ignorare i valori di spostamento (X,Y). Il comportamento qui menzionato è probabile che cambierà in futuro.]

Una volta che i modelli sono stati specificati per tutti i componenti desiderati, dall' interno di pcbnew, selezionare il menu **File**, e poi **Esporta**, e infine **Esportazione IDFv3**. Una finestra di dialogo si aprirà (vedere [figura 3](#)) che permetterà di impostare il nome del file e le unità di misura (mm o mils) IDF del file in uscita. I file IDF esportati possono essere visualizzati con il CAD meccanico **FreeCAD** o convertiti in VRML usando lo strumento idf2vrml e visualizzati con qualsiasi visualizzatore VRML adatto.

¹ http://www.simplifiedsolutionsinc.com/images/idf_v30_spec.pdf

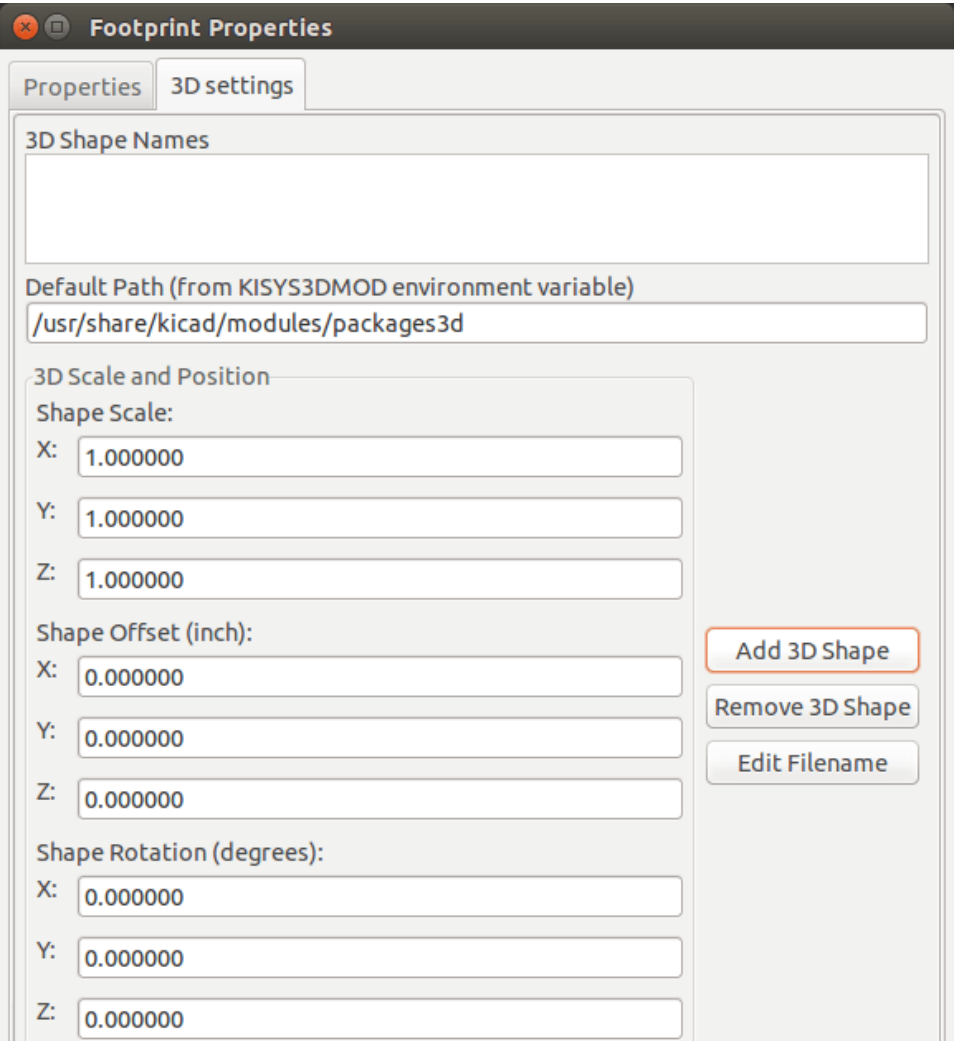


Figura 1: Proprietà modulo, impostazioni 3D

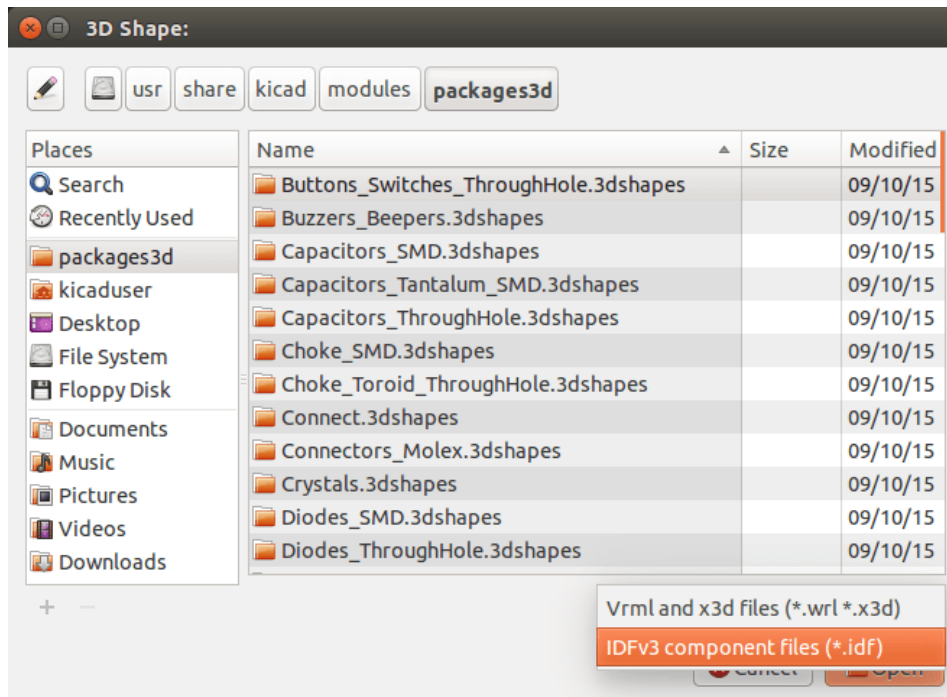


Figura 2: Selezione contorno componente IDF

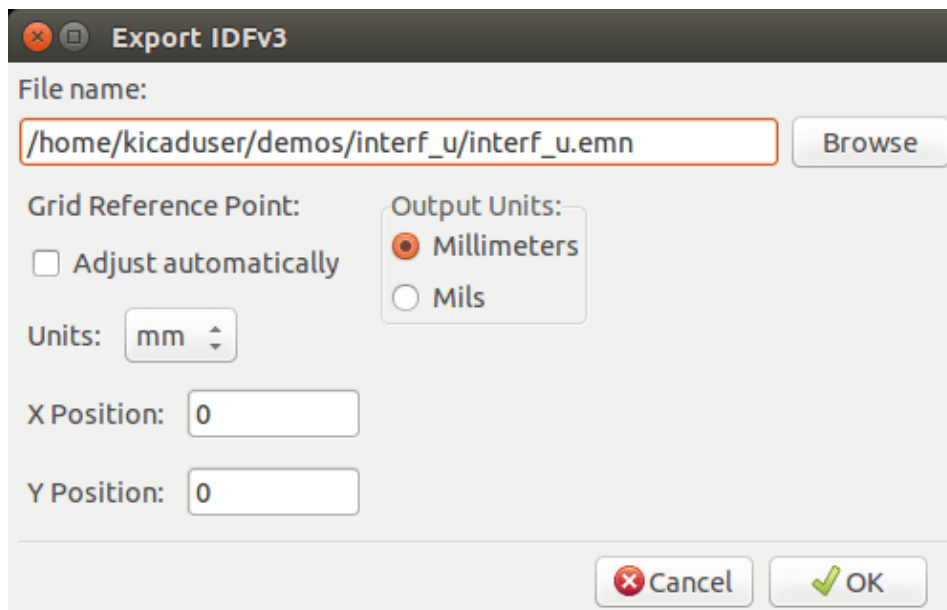


Figura 3: Impostazioni risultati IDF

3 Creazione di un file contorno componente

Il file del contorno del componente (*.idf) consiste in un'unica sezione `.ELECTRICAL` o `.MECHANICAL` come descritto nel documento delle specifiche. La sezione può essere preceduta da un numero qualsiasi di righe di commento;

le righe di commento vengono copiate dall' esportatore nel file delle librerie e possono essere usate per tenere traccia dei metadati come i riferimenti ai documenti usati per determinare il contorno e le dimensioni del componente.

La sezione del contorno del componente contiene campi consistenti in stringhe, numeri interi, o in virgola mobile. Una stringa è un insieme di caratteri che può includere gli spazi; se una stringa contiene spazi, deve essere virgolettata. Le virgolette non possono apparire all' interno di una stringa. I numeri in virgola mobile possono essere rappresentati usando le notazioni decimale o esponenziale ma la notazione decimale è la preferita per questioni di leggibilità. Il punto decimale deve essere un punto e non una virgola. Il file IDF deve consistere solo di caratteri standard ASCII a 7-bit; l' uso di caratteri a 8-bit provocherà un comportamento imprevedibile.

Un file IDF è composto di diverse SECTION (sezioni) che a loro volta sono composte da RECORD che sono composti da FIELD (campi). Per i file contorno IDF, può esistere un solo tipo di sezione che deve essere di tipo .ELECTRICAL o .MECHANICAL. Un record è una singola riga di testo e può contenere uno o più field (campi). I field (campi) sono sequenze di caratteri separate da uno o più spazi che non appaiono tra virgolette. Tutti i campi di un record devono stare su una sola riga; i record non possono distribuirsi su più righe.

Le intestazioni di sezione (.ELECTRICAL o .MECHANICAL) sono considerate il primo record (Record 1) della sezione. Record 1 deve essere seguito da Record 2 che possiede quattro campi:

1. Geometry Name: una stringa che, assieme al Part Number deve formare un identificatore univoco per il contorno del componente. Per package standard, il nome del package è un buona scelta come valore del geometry name (nome geometria), per esempio SOT-23. Per package specifici, il codice parte del costruttore è una buona scelta per il geometry name.
2. Part Number: malgrado sia stato ovviamente pensato per il codice parte, per esempio BS107, è meglio usare questa stringa come aiuto alla descrizione del package. Per esempio se il geometry name è TO-92, la voce part number può essere usata per descrivere la disposizione delle piazzole o l' orientamento di questo particolare file di contorno TO-92.
3. IDF Unit: (unità di misura IDF) questa deve essere MM o THOU e si applica solo alle unità che descrivono il contorno di questo singolo componente.
4. Height: (altezza) è un numero in virgola mobile che rappresenta l' altezza nominale del componente usando le unità specificate nel campo 3.

Record 2 deve essere seguito da un numero di voci Record 3 che specificano il contorno del componente. Record 3 consiste di quattro campi:

1. Loop Index: 0 (i punti del contorno sono specificati in ordine antiorario) o 1 (i punti del contorno sono specificati in ordine orario)
 2. Coordinata X: un numero in virgola mobile
 3. Coordinata Y: un numero in virgola mobile
 4. Included Angle: a floating point number. If the value is 0 then a straight line segment is drawn from the previous point to this point. If the value is 360 then the previous point specifies the center of a circle and this point specifies a point on the circle; never specify a circle using a value of -360 as at least one major mechanical CAD package does not behave well in that situation. If the value is negative then a clockwise arc is drawn from the previous point to this point and if the value is positive then a counter-clockwise arc is drawn.
-

Only one closed loop is permitted and it is not possible to specify a cutout. The last point specified must be the same as the first point unless the outline is a circle.

Esempio IDF file 1:

```
# un semplice cilindro - potrebbe rappresentare un condensatore elettrolitico
.ELECTRICAL
    "cylinder" "5mm OD, 5mm height" MM 5
    0 0 0 0
    0 2.5 0 360
.END_ELECTRICAL
```

Esempio file IDF 2:

```
# una T rovesciata
# commento aggiunto per amore dei commenti
.ELECTRICAL
    "T Maiuscola" "5x8x10mm, rovesciata" 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 Linee guida per la creazione dei contorni

Quando si creano contorni, in special modo quando si condivide il proprio lavoro con altri, la consistenza nella progettazione e nella scelta dei nomi dei file aiuta le persone a localizzare i file e a piazzare i componenti più velocemente e con minimo sforzo.

4.1 Denominazione dei pacchetti

Try to make some information about the outline available in the filename to give the user a general idea of what the outline is. For example axial leaded cylindrical packages may represent some types of capacitors as well as some types of resistors, so it makes sense to identify an outline as a horizontal or vertical axial leaded device and to add some extra information on the relevant dimensions: diameter, length, and pitch are the most important. If a device has a unique outline, the manufacturer's part number and a prefix to indicate the class of device are adequate.

4.2 Commenti

Use comments in the IDF file to give users more information about the outline, for example a reference to the source used for dimensional information.

4.3 Voci di geometria e numero di parte

Think carefully about the values to give to the Geometry and Part Number entries. Taken together, these strings act as a unique identifier for the MCAD system. The values of the strings will ideally have some meaning to a user, but this is not necessary: the values are primarily intended for the MCAD system to use as a unique ID. Ideally the values chosen will be unique within any large collection of outlines; choosing values well will result in fewer clashes especially in complex boards.

4.4 Orientamento e posizionamento pin

For through-hole components there are no widely accepted schemes for determining the pin orientation and component center in 3D models. For consistency, if there are only 2 pins they must be in a horizontal arrangement (see [figure 4](#)) along the X axis and for 3 pins try to keep 2 in a horizontal arrangement on the X axis. Polarized devices such as electrolytic or tantalum capacitors must have the positive lead on Pin 1 and diodes must have the cathode on Pin 1; this is to maintain compatibility of the schematic symbols with the orientation defined for SMT devices; however, note that many existing KiCad schematics and footprints place the anode at Pin 1.

Nota

In the latest revision of the KiCad footprints on github the anode is now Pin 2 for THT as well as SMT components.

For DIP devices the center of the outline must be at the center of the rectangle described by the pin locations and Pin 1 is preferably at the top left corner; this will maintain some consistency with the standardized orientation of SMT components; however, such a model will be rotated -90 degrees relative to most existing KiCad component footprints and VRML models. For items such as a horizontal radial leaded capacitor or a horizontal TO-220 package, prefer to place the leads in a row on the X axis and with the body of the device extending upwards (see [figure 4](#)). Non-polarized vertical axial leaded components must have the wire on the right hand side; polarized vertical axial leaded components may have the wire on either side, depending on whether Pin 1 is on the lower end (wire on right) or on the upper end (wire on left).

Nota

In the current revision of the KiCad footprint modules the THT components are being organized with pins along the Y axis rather than the X axis and Pin 1 of the device is at the origin rather than at the center of the package. Orient and position the component outline to suit your specific footprints; this will avoid the need to specify a non-zero rotation for the IDF component outlines. Since the IDF exporter currently ignores the (X,Y) offset values it is vital that you use the correct origin in the IDF component outline.

For SMT components the orientation, package center, and outline are defined by various standards. Use the standard appropriate to your work. Also keep in mind that many devices do not conform to any standard; in such cases the offending device is probably best identified by using the manufacturer's part number in the outline file name. In general, an SMT outline is a rectangle encompassing the component package and including the leads; the package is oriented such that Pin 1 is as close as possible to the top left corner and the top left corner is usually chamfered for visual reference.

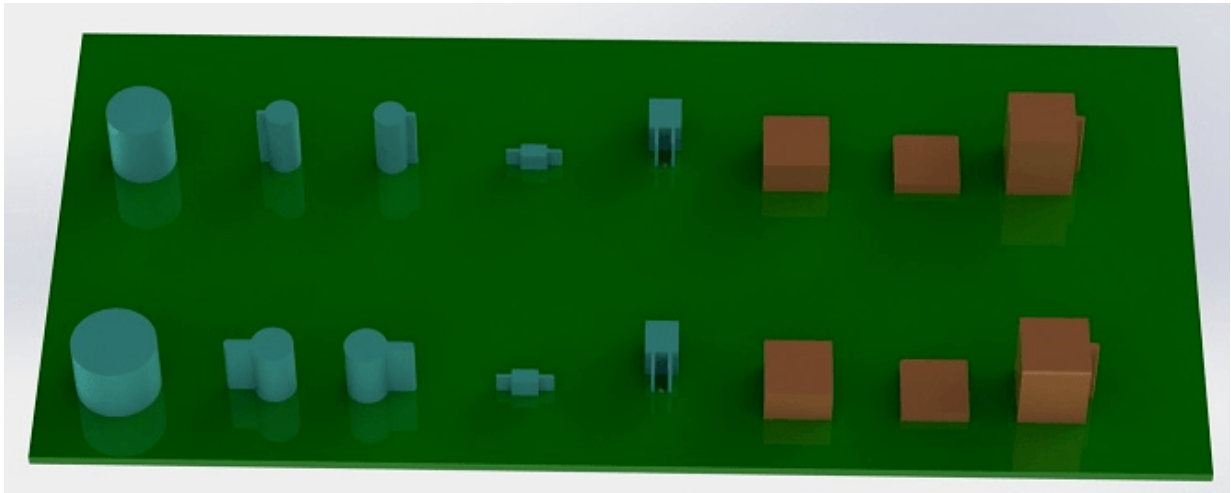


Figura 4: Sample outlines generated by the programs `idfcyl` and `idfrect` and rendered by SolidWorks.

From left to right are (a) vertical radial leaded cylinder, (b) vertical axial leaded cylinder with wire on left, (c) vertical axial leaded cylinder with wire on right, (d) horizontal axial leaded cylinder, (e) horizontal radial leaded cylinder, (f) square outline, plain, (g) square outline with chamfer, (h) square outline with axial lead on right. The top outlines were specified in units of millimeters while the bottom outlines were specified in units of inches.

4.5 Suggerimenti sulle dimensioni

The purpose served by the extruded outlines is to give the mechanical designer some idea of the location and physical space occupied by each component. In a typical scenario the mechanical designer will replace some of the crude outlines with more detailed mechanical models, for example when checking to ensure that a right-angle mounted LED will fit into a hole on a panel. In most situations the accuracy of an outline doesn't matter, but it is good practice to create outlines which convey the best mechanical information possible. In a few instances a user may wish to fit the component into a case with very little excess space, for example in a portable music player. In such a situation, if most extruded outlines are a good enough representation of components then the mechanical designer may only have to replace very few models while designing the case. If the outlines are not a reliable reflection of reality then the mechanical designer will waste a lot of time replacing models to ensure a good fit. After all, if you put garbage in you can expect garbage to come out. If you put in good information, you can be confident of good results.

5 IDF Component Outline Tools

A number of command-line tools are available to help generate IDF component outlines. The tools are:

1. **idfcyl**: creates an outline of a cylinder in vertical or horizontal orientation and with axial or radial leads
2. **idfrect**: creates an outline of a rectangle which may have either an axial lead or a chamfer in the top left corner
3. **dx2idf**: converts a drawing in DXF format into an IDF component outline

5.1 idfcyl

When **idfcyl** is invoked with no arguments it prints out a usage note and a summary of its inputs:

```
idfcyl: This program generates an outline for a cylindrical component.
```

```
    The cylinder may be horizontal or vertical.
```

```
    A horizontal cylinder may have wires at one or both ends.
```

```
    A vertical cylinder may have at most one wire which may be
    placed on the left or right side.
```

```
Input:
```

```
    Unit: mm, in (millimeters or inches)
```

```
    Orientation: V (vertical)
```

```
    Lead type: X, R (axial, radial)
```

```
    Diameter of body
```

```
    Length of body
```

```
    Board offset
```

```
    *   Wire diameter
```

```
    *   Pitch
```

```
    **  Wire side: L, R (left, right)
```

```
    *** Lead length
```

```
    File name (must end in *.idf)
```

```
NOTES:
```

```
    *   only required for horizontal orientation or
        vertical orientation with axial leads
```

```
    **  only required for vertical orientation with axial leads
```

```
    *** only required for horizontal orientation with radial leads
```

The notes can be suppressed by entering any arbitrary argument on the command line. A user can manually enter information at the command line or create scripts to generate outlines. The following script creates a single axial leaded outline with the lead on the right hand side:

```
#!/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

When **idfrect** is invoked with no arguments it prints out a usage note and a summary of its inputs:

```
idfrect: This program generates an outline for a rectangular component.
    The component may have a single lead (axial) or a chamfer on the
    upper left corner.
Input:
    Unit: mm, in (millimeters or inches)
    Width:
    Length:
    Height:
    Chamfer: length of the 45 deg. chamfer
    *   Leaded: Y,N (lead is always to the right)
    ** Wire diameter
    ** Pitch
    File name (must end in *.idf)

NOTES:
    *   only required if chamfer = 0

    ** only required for leaded components
```

The notes can be suppressed by entering any arbitrary argument on the command line. A user can manually enter information at the command line or create scripts to generate outlines. The following script creates a chamfered rectangle and an axial leaded outline:

```
#!/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

The DXF file used to specify the component outline can be prepared with the free software [LibreCAD](#) for best compatibility. When **dxf2idf** is invoked with no arguments it prints out a usage note and a summary of its inputs:

```

dxf2idf: this program takes line, arc, and circle segments
        from a DXF file and creates an IDF component outline file.

Input:
  DXF filename: the input file, must end in '.dxf'
  Units: mm, in (millimeters or inches)
  Geometry Name: string, as per IDF version 3.0 specification
  Part Name: as per IDF version 3.0 specification of Part Number
  Height: extruded height of the outline
  Comments: all non-empty lines are comments to be added to
            the IDF file. An empty line signifies the end of
            the comment block.
  File name: output filename, must end in '.idf'

```

The notes can be suppressed by entering any arbitrary argument on the command line. A user can manually enter information at the command line or create scripts to generate outlines. The following script creates a 5mm high outline from a DXF file *test.dxf*:

```

#!/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

The `idf2vrm1` tool reads a set of one IDF Board (*.emn*) and one IDF Component file (*.emp*) and produces a VRML file which can be viewed with a VRML viewer. This feature is useful for visualization of the board assembly in cases where the user does not have access to MCAD software. Invoking `idf2vrm1` without any arguments will result in the display of a usage message:

```
>./idf2vrm1
Usage: idf2vrm1 -f input_file.emn -s scale_factor {-k} {-d} {-z} {-m}
flags:
  -k: produce KiCad-friendly VRML output; default is compact VRML
  -d: suppress substitution of default outlines
  -z: suppress rendering of zero-height outlines
  -m: print object mapping to stdout for debugging purposes
example to produce a model for use by KiCad: idf2vrm1 -f input.emn -s 0.3937008 -k
>
```

[**BUG:** The `idf2vrm1` tool currently does not correctly render **OTHER_OUTLINE** entities in an `emn` file if that entity is specified on the back layer of the PCB; however you will not notice this bug using files exported by KiCad since there is no mechanism to specify such an entity. Essentially this bug is only an issue in rare instances where you might render a third party `emn` file which does employ the entity on the back side of a board.]