Intermedius™

INTERMEDIUS DESIGN INTEGRATION, LLC

"Bringing Electrical and Mechanical Design Together"

Intermediate Data Format (IDF)

Version 4.0

DATA EXCHANGE SPECIFICATION FOR THE MECHANICAL DESIGN, ELECTRICAL LAYOUT, AND PHYSICAL ANALYSIS OF PRINTED CIRCUIT ASSEMBLIES

Revision A
(Pre-implementation Draft)

July 17, 1998

Developed By:
Dave Kehmeier & Tom Makoski
Intermedius Design Integration, LLC

License and Warranty

The Intermediate Data Format Specification Version 4.0 (“IDF”) and all patent, copyright, trademark, trade secret, and other intellectual property and proprietary rights (collectively “IPR”) are owned solely by Intermedius Design Integration, LLC (“Intermedius”). Users of the IDF shall have no right, title or interest in or to the IDF or any IPR associated with the IDF except for the limited license granted below.

Limited License

Intermedius shall freely distribute the IDF to any person or organization (“User”) interested in working with the IDF. Intermedius shall not sell or otherwise distribute the IDF for any valuable consideration. By distributing the IDF to any User, Intermedius grants that User a fully paid-up, limited, non-exclusive, irrevocable, perpetual, royalty-free, worldwide license to:

I. Incorporate the IDF into, and use the IDF in the development of User’s software products, and

II. Reproduce, market, and distribute the IDF as incorporated into such software products, including the right to sell, distribute and sublicense such products in any manner in which the User chooses, for valuable consideration or otherwise.

This License is subject to the following restrictions:

I. The User shall not alter the IDF in any material respect,

II. The User shall not sell or otherwise distribute the IDF alone for any valuable consideration, and

III. The User shall place or cause to be placed on all copies of the IDF distributed by the User clear and conspicuous notices of Intermedius’ ownership of the IDF and the associated IPR sufficient to protect the legal and beneficial rights, title, and interests of Intermedius. For example:

Copyright © 1998 by Intermedius Design Integration, LLC - - All rights reserved

Disclaimer of Warranty

The IDF is provided "AS-IS". Intermedius makes no warranty with regard to the IDF either express or implied. Without limiting the generality of the foregoing, INTERMEDIUS SPECIFICALLY DISCLAIMS, AND USER HEREBY WAIVES ALL RIGHTS AND REMEDIES THAT IT MAY HAVE WITH RESPECT TO ALL WARRANTIES OF INTERMEDIUS, EXPRESS OR IMPLIED, ARISING BY LAW OR OTHERWISE, WITH RESPECT TO THE IDF, INCLUDING, BUT NOT LIMITED TO, ANY: (A) IMPLIED WARRANTY OF MERCHANTABILITY; (B) IMPLIED WARRANTY OF FITNESS FOR A PARTICULAR USE OR PURPOSE; AND (C) IMPLIED WARRANTY ARISING OUT OF COURSE OF PERFORMANCE, COURSE OF DEALING OR USAGE OF TRADE. Intermedius shall not be obligated to support the IDF.

Acknowledgments

The development of IDF 4.0 was made possible through the technical and financial contributions of industry-leading CAD vendors and their end users, representing all of the major electronics industry segments: automotive, aerospace, computer, consumer, industrial controls, medical, military, telecom, and test & measurement. Funding was provided by the following CAD vendors:

Cadence Design Systems, Inc.
INCASES Engineering, GmbH
Mentor Graphics Corporation
OrCAD, Inc.
PADS Software, Inc.
VeriBest Incorporated
Redac Systems Limited

Bentley Systems, Inc.
CoCreate Software, GmbH
LTX Software, Inc.
Maya Heat Transfer Technologies, Ltd.
Parmetric Technology Corporation
Structural Dynamics Research Corporation

Additional funding, along with technical requirements, was provided by the following end user sponsors:

Delco Electronics Systems
(Delphi Corporation)
Harris Corporation
Hewlett-Packard Company

Intel Corporation
Square D Corporation
Visteon Company
(Ford Motor Company)

Technical requirements were also provided by the following end-user participants:

Apple
Bosch
Braun
Cisco Systems
Daimler Benz Aerospace
FLIR
Fluke
Guidant/CPI
Kostal

Klockner-Moeller
Lockheed Martin
Matsushita
Nortel
Phoenix Contact
Plexus
Sun
TRW

Change History

The following tables describe the change history for each revision to IDF V4.0

Revision A	Released July 17, 1998	
Date	**Author**	**Description of Change**

Table of Contents

Table of Figures

1 Introduction

This document represents version 4.0 of the Intermediate Data Format (IDF) Specification. IDF 4.0 is the successor to IDF 2.0 and 3.0, which were initially developed and supported by Mentor Graphics Corporation. IDF 4.0 is based on IDF 2.0 and 3.0, but includes significant changes with respect to content, representation, and format. It is neither upward nor backward compatible with IDF 2.0 or 3.0. However, the content of IDF 2.0 and 3.0 can be fully represented in IDF 4.0.

1.1 Purpose of the IDF

The purpose of the Intermediate Data Format (IDF) is to provide a neutral representation for exchanging printed circuit assembly (PCA) data among mechanical design (MCAD), printed circuit layout (ECAD), and physical design analysis (MCAE) applications. The IDF provides the groundwork to enable the users of those applications to integrate the mechanical and electrical design and analysis processes for products containing PCAs.

1.2 Data Scope of the IDF

The data scope of the IDF includes all information that is commonly shared among mechanical design, circuit board layout, and physical analysis during the design and analysis of products containing PCAs. The following list represents the data scope of the IDF:

- Each of the major interconnect technologies (traditional PCB, MCM, hybrid)
- Panel and board assemblies
- Board design variants
- Panel, board, and component parts
- 3D part shapes consisting of extrusions, cutouts, and cavities
- Mounting side and opposite side component part shapes
- Holes (mounting, tooling, pin, via, thermal via)
- Conductors (pads, traces, filled areas)
- Restriction regions (keepins and keepouts)
- Graphics (to represent miscellaneous board features such as fiducials and silkscreen)
- Annotations (to communicate miscellaneous design information between designers)
- Figures, footprints, and sublayouts (to group related features and component instances)
- Properties (thermal and structural)

1.3 Activity Scope of the IDF

The activities that are within scope of the IDF include (but are not necessarily limited to) the transfer of:

- Initial board shape and physical features, critical component locations, and restriction regions from MCAD to ECAD to provide a starting point for PCA layout
- Modifications to the board shape, component locations, and restriction regions between ECAD and MCAD to refine board layout
- Final board shape, component shapes, and component locations from ECAD to MCAD to verify board layout
- Final board shape, component shapes, and component locations from ECAD and MCAD to MCAE for physical design analysis and verification
- Final board shape, component shapes, and component locations from ECAD to MCAD for tooling and fixture design, and manufacturing documentation
- Basic component shapes (outline, height) among ECAD, MCAD, and CAE (library transfer)

The IDF does not provide a full product or design representation of a PCA. As such, *it is not intended to:*

- Provide a full functional or electrical description of the PCA
- Provide a means of converting PCA designs from one ECAD system to another
- Provide a means for archiving PCA designs
- Provide a means for manufacturing, assembly, test, or detailed documentation of a PCA

1.4 Changes from the Previous Version

IDF 4.0 represents a significant change from IDF 3.0 in content, representation, and format.

The data content in IDF 4.0 incorporates that of IDF 3.0, and is expanded to provide a more complete physical and functional representation of PCA designs. Many new features have been introduced, including conductors and miscellaneous graphics. Mechanisms have been provided to group and associate features and parts. Part shapes may consist of multiple extrusions with cutouts and cavities. Miscellaneous properties can be associated with any assembly, part, instance, or feature. In addition, the user can extend the data content, via user-defined properties.

The file format is also new. The format uses a simple syntax, and is based on a context free grammar for ease of parsing. However, additional formatting rules have been added to enhance human readability of the physical file.

Due to the changes in data content and file format, IDF 4.0 is neither upward nor backward compatible with IDF 3.0 or previous versions. However, the content of IDF 2.0 and 3.0 can be fully represented in IDF 4.0.

2 IDF Data Content

This section describes the data content and organization (data model) of the IDF. The purpose of this section is to explain PCA data as it is represented by the IDF – it does not provide a functional, physical, or manufacturing description of a PCA.

All IDF data is represented by entities, most of which are introduced in this section. For convenience, entity names are italicized throughout this section. The complete set of entity definitions can be found in Section 6. In addition, Appendix A contains a graphic representation of the IDF data.

2.1 Data Model Overview

The IDF data model contains the subset of PCA design information that needs to be shared among PCA Layout, Mechanical Design, and Physical Analysis, and takes into account the manner in which these design systems represent and use PCA data. As a result, the IDF provides an approximate physical representation[1] and abbreviated functional representation of single PCAs, and manufacturing panels containing multiple PCAs.

The IDF data model is based on a hierarchy of assemblies, parts, and features. Assemblies are constructed from instances of parts and other assemblies. Parts are constructed from features. Features define the geometric shape and other physical characteristics of parts, and convey functional information as well.

The IDF also provides various means to represent associations among instances and features that are useful to maintain during the design of the PCA. *Figures* represent named collections of related features in board and panel parts (the *Hole*, *Pad*, and *Keepout* for a mounting hole, for example). *Footprints* associate features in a board or panel part with the instances they support in an assembly (the *Holes* and *Pads* associated with the pins of an *Electrical Instance*, for example). *Sublayouts* preserve the relative positions of component instances and *Board Part* features in a *Board Assembly* so that they can be repositioned or removed as a single “block”.

[1] The physical representation is based on a 2½D description of the physical data: all features in parts and all parts in assemblies are located relative to an XY coordinate system with explicit or implicit Z-axis offsets as appropriate. It is assumed that fully detailed 3D representations may be required for some parts. For these parts, detailed 3D models must be separately built and maintained in an MCAD library, and externally referenced from the IDF.

Figure 2.1 illustrates the conceptual view of the IDF data model.

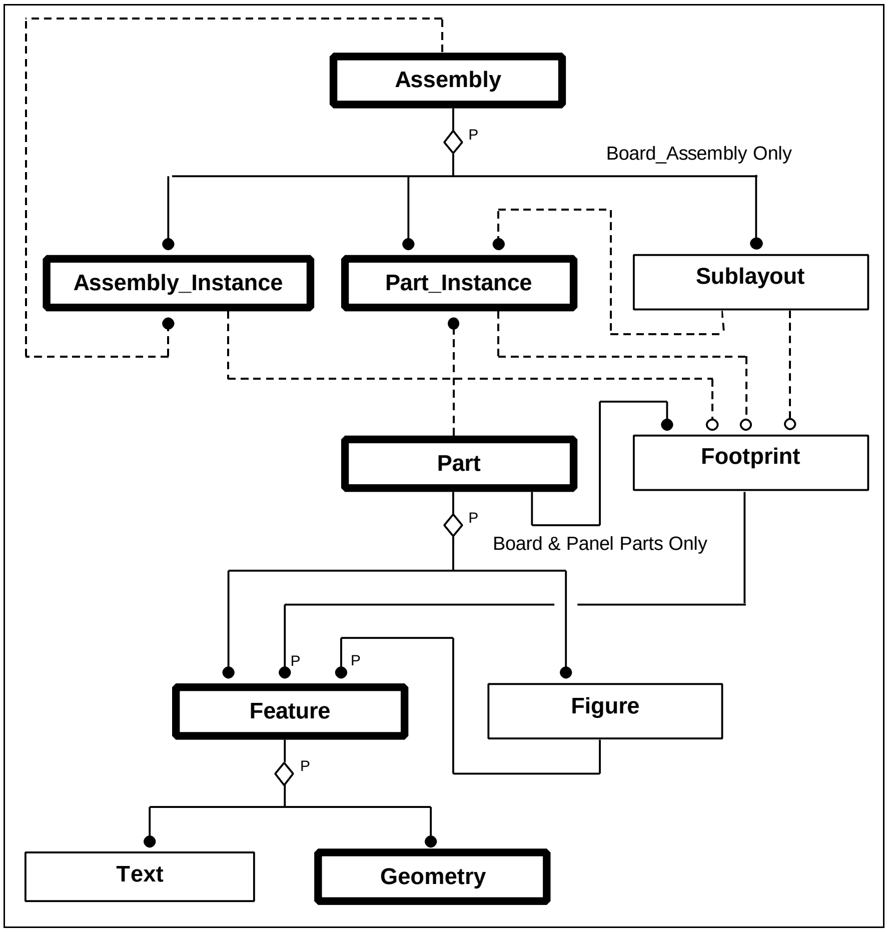


Figure 2.1 Conceptual View of the Data Model

2.2 Parts

In general, an IDF part represents an actual physical part (something that contributes to the actual board or panel assembly). A part is represented in the IDF by a part definition that defines its shape and any other applicable physical features or properties. Every part definition has a name, units of length measurement, and a local origin. All features that comprise a part are defined in the part's units, relative to its origin.

Actual parts that differ from one another in any way, typically require separate part definitions. However, for *Electrical Parts*, the difference may only be functional. (An example is two ICs in identical physical packages that perform different circuit functions.) As a result, a single IDF part definition can be used to represent several actual physical parts.

2.2.1 Panel and Board Parts

Every *Panel Part* and *Board Part* has a shape that is represented by at least one *Extrusion*, and additional features that modify the shape (*Cavities*, *Cutouts*, and *Holes*), enhance the shape (conductors and *Graphics*), or provide additional design information for the part (*Keepins*, *Keepouts*, and *Annotations*).

Typically, the shape of a *Panel Part* or *Board Part* is planar and can be adequately represented by a single *Extrusion*, with *Cutouts* and *Cavities* as required. Rigid-flex *Board Parts*, in which the rigid and flexible portions have different thicknesses, can be represented with multiple *Extrusions*.

Panel Parts and *Board Parts* can include *Figures*, which represent groups of features that can be instanced multiple times, and *Footprints*, which relate the features required to support an instance of a part or assembly.

2.2.2 Component Parts

Parts that are assembled to a *Board Part* (and occasionally, a *Panel Part*) are referred to collectively in this specification as "components". The IDF supports two types of component parts: *Electrical Parts* and *Mechanical Parts*. *Electrical Parts* have *Pins* (also known as "terminals") that are electrically connected to the board. Examples include resistors, ICs, jumpers, connectors, LEDs, and sockets. *Mechanical Parts* do not have pins and are not electrically connected to the board. Examples include card extractors, stiffeners, heatsinks, standoffs, and barcode labels.

As with *Panel Parts* and *Board Parts*, the shape of a component part is represented by one or more *Extrusions*, along with *Cutouts* and *Cavities* as required. It is often useful to represent a component part shape with multiple *Extrusions*. For example, the shape of a surface mount component part can be represented with two *Extrusions* for the space occupied by the pins and another for the body (Figure 2.2)[2].

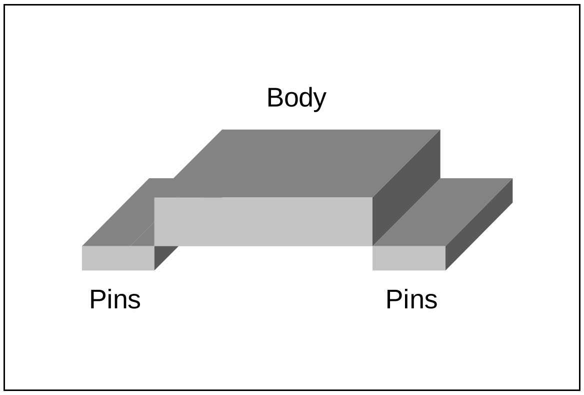


Figure 2.2 Surface Mount Component with Multiple Extrusions

[2] Even this simple 3D representation is more complex than many ECAD systems can fully support. For example, ECAD systems typically represent a component part's shape simply as an outline with a single height (an *Extrusion*).

Component parts can have mounting side and opposite side shapes. An example of an opposite side shape is a separate fastener or retainer that keeps a component in the board.

Most *Electrical Parts* represent actual physical parts that are mounted on either the top or bottom surface of the board. However, some *Electrical Parts* are fabricated into or onto the board itself. These are referred to as “printed” *Electrical Parts*. Examples are screened resistors and finger connectors. Although these parts are not included in a manufacturing Bill of Materials, they are significant from a design standpoint and are represented as *Electrical Parts* in most ECAD systems and in the IDF.

2.3 Assemblies and Instances

The IDF supports both *Panel Assemblies* and *Board Assemblies*. An assembly consists of part instances and (in the case of a *Panel Assembly*) assembly instances. Each instance references a corresponding part or assembly definition (by its name). In this way, parts and assemblies are re-used rather than recreated for each instance.

Instances have unique identifiers called reference designators to differentiate them from other instances in the assembly. Reference designators are required by ECAD systems for *Electrical Part Instances* to represent the electrical connectivity of a PCA design. Reference designators are also required to establish a cross-reference between the instances in the MCAD and ECAD design applications for design updates. For example, if an instance of a resistor in an MCAD assembly is moved to a new location, it is not possible to determine which instance needs to be moved in the ECAD application, without specifying the reference designator.

To support the creation of a Bill of Material (BOM), part numbers are assigned to part instances. Note that because in general, a single part definition can be used to represent more than one actual physical part, the part number cannot be associated with the part definition. Conversely, *Electrical Instances* can have the same part number, but reference different part definitions that differ in mounting style. For example, a resistor may be mounted vertically in one instance and horizontally in another, requiring two different *Electrical Part* definitions.

Figure 2.3 illustrates the relationship of component part definitions and instances with their names, reference designators, and part numbers.

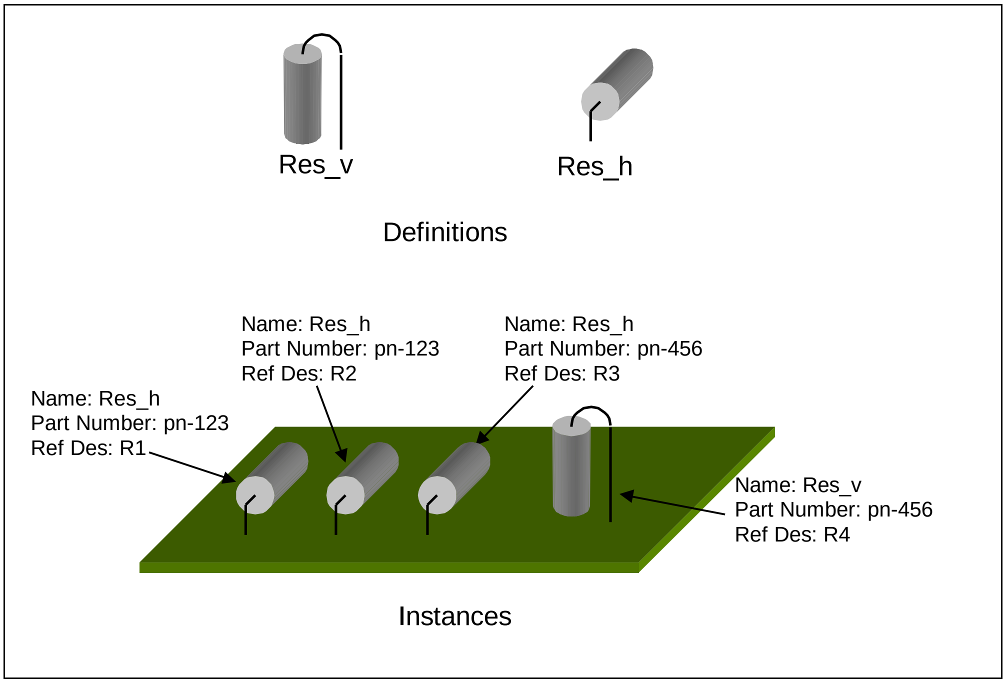


Figure 2.3 Component Definitions and Instances

2.3.1 Board Assembly

A *Board Assembly* consists of a *Board Part Instance* on which *Electrical Part Instances* and *Mechanical Part Instances* are mounted. A component instance's placement in a *Board Assembly* is specified by its XY location, the side of the board, a rotation, and a Z-axis offset.

A group of component instances may contribute a collective *Footprint* to the *Board Part* through a *Sublayout* in the *Board Assembly*.

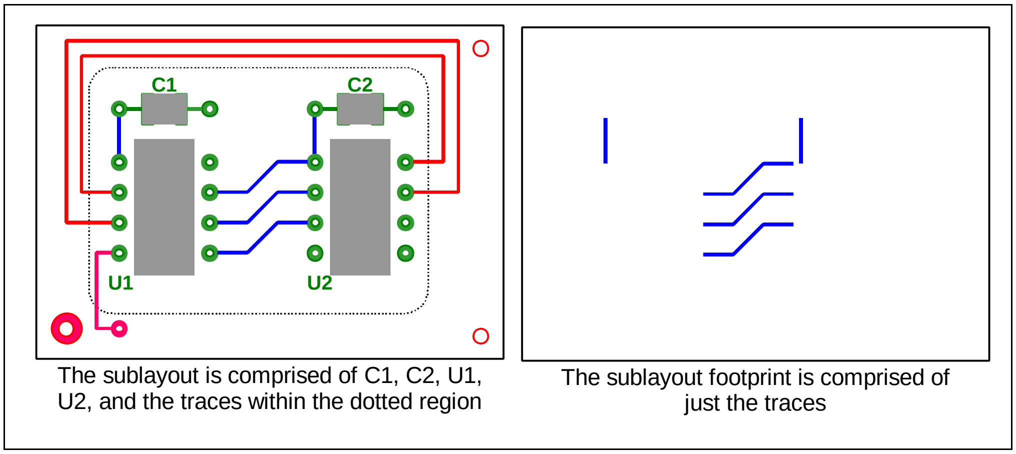
A *Board Assembly* may have design variants. These represent variations of the *Board Assembly* in which certain component instances are not loaded, or alternate components are used (producing a functionally different circuit for each variant). The *Board Part* (and all of its board-related features) remains the same in all *Board Assembly* variants – only the component instances are different.

2.3.2 Panel Assembly

A *Panel Assembly* consists of a *Panel Part Instance* and multiple instances of *Board Assemblies*. A *Panel Assembly* may contain multiple instances of other *Panel Assemblies* (subpanels) and component instances. Whereas component instances are located <u>on</u> the *Panel Part* of a *Panel Assembly*, *Board Assembly Instances* and *Panel Assembly Instances* are located <u>in</u> the *Panel Part* of a *Panel Assembly*.

2.3.3 Sublayouts

A *Sublayout* defines a group of component instances in a *Board Assembly* (Figure 2.4). It represents a “sub-assembly” of components and optionally, features on the board (defined in a *Footprint* for the *Sublayout*) that are physically associated. *Sublayouts* are used to maintain the relative position of component instances and *Footprint* features so that if any of the component instances or *Footprint* features in the *Sublayout* are moved, all of the component instances and *Footprint* features move together.



The sublayout is comprised of C1, C2, U1, U2, and the traces within the dotted region

The sublayout footprint is comprised of just the traces

Figure 2.4 Sublayout in a Board Assembly

2.4 Features

IDF features define the physical shape and appearance of parts, and provide additional information related to the design as well. The following paragraphs describe the features supported by the IDF.

2.4.1 Cavities, Cutouts, and Extrusions

All parts in the IDF have basic 3D shapes that are created from *Extrusions*, *Cutouts*, and *Cavities*.

An *Extrusion* represents a solid shape, which is defined by a linear extrusion (along the Z-axis) of an XY planar outline. The start of the *Extrusion* is the “bottom” surface and the end of the *Extrusion* is the “top” surface. In a component part shape, the bottom of an *Extrusion* is the surface facing the board when the component is mounted, and the top surface of the extrusion faces away from the board[3].

Cutouts and *Cavities* represent the absence (or void) of material in a part. *Cutouts* go all the way through the part. *Cavities* extend into the part a specific depth from its top or bottom surface.

[3] Note that the shape of a component part instance can extend into a *Cavity* or through a *Cutout* in the board. As a result, the bottom surface of an *Extrusion* in a component part shape may actually be inside the board or on the opposite side of the board after the component is mounted.

2.4.2 Holes

Holes are features in *Panel Parts* and *Board Parts* that serve specific purposes such as for fastening components (mounting), aligning the part in a manufacturing fixture (tooling), inserting component pins (pin), or providing electrical connectivity among the conductive layers of a board (via). Like *Cutouts* and *Cavities*, *Holes* affect the physical shape of the *Panel Part* and *Board Part*. However, due to design and manufacturing considerations, they are represented separately in the IDF.

Holes usually have regular shapes. Typically, they are round (produced by drilling) and go all the way through the board. In some designs, *Holes* for pins and vias are "blind", extending from the surface of the board to a conductor layer within the board. *Holes* for vias can be "buried", completely contained within the layers of the board. The span of a blind or buried hole can be fully represented in the IDF by including *Physical Layers* with the *Board Part*. Figure 2.5 illustrates various *Hole* shapes.

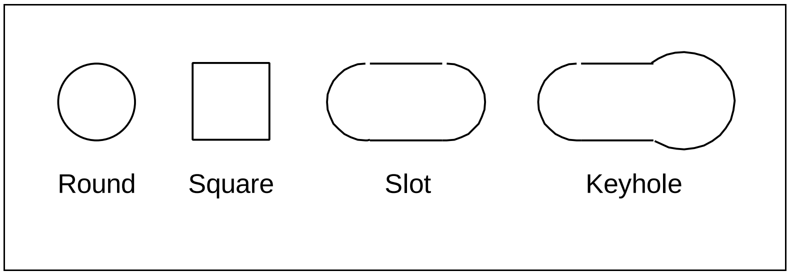


Figure 2.5 Typical Hole Shapes

Plated *Holes* can be electrically connected in the circuit. Consequently, the net name associated with a plated *Hole* can be provided in the IDF.

2.4.3 Conductors

Conductors represent the physical interconnect among the *Electrical Instances* in a *Board Assembly*. They are included in the IDF to enhance the physical shape of the *Board Part* but are not intended to convey the electrical connectivity of the design. Conductors are typically transferred from ECAD to MCAD where they are used as a reference for mechanical design activities such as moving components, locating additional components, or designing *Mechanical Parts* that contact the *Board Part*.

Conductors are represented in the IDF by *Pads*, *Traces*, and *Filled Areas*:

- *Pads* are used to connect *Traces* to component *Pins*, vias, or bond wires. *Pads* may also be used to create contacts on the *Board Part* for an external connection (such as an edge connector), or testing purposes. The shape of a *Pad* can be arbitrarily complex but is generally simple and regular (round, square, rectangular, or oblong).
- *Traces* are piecewise curves consisting of linear or circular arc segments that connect the pins and vias on the Board.

- *Filled Areas* represent regions of conductor. The *Filled Area* can either be solid, or contain voids.

When it is useful to transfer conductors via the IDF, usually only the conductors on the top and bottom surfaces of the board are needed. However, conductors on internal layers can be represented if *Physical Layers* are defined for the *Board Part*.

The net names associated with specific conductors can be provided in the IDF.

2.4.4 Graphics and Annotations

Graphics and *Annotations* are features that provide additional information about the design. *Graphics* are used to represent board and panel features that are not encompassed by the predefined IDF feature set. Examples are fiducials, soldermask shapes, logos, and silkscreen text and graphics.

Annotations, on the other hand, are not features of the board or panel, but simply represent a means to communicate additional information about the design. *Annotations* are used to pass notes between designers, show decomposed dimensions or other drafting-related details, and provide cosmetic detail on parts for visual reference (body outline and pin "whiskers" on an *Electrical Part*, for example).

Both *Graphics* and *Annotations* are constructed from *Text* and geometry. *Leaders* are also provided for use in *Annotations* to visually associate a note with its subject.

2.4.5 Keepins and Keepouts

Keepins and *Keepouts* define regions on a *Board Part* in which restrictions on the placement of component instances and/or features apply. *Keepins* define regions in which component instances or features must be located. *Keepouts* define regions in which component instances or features must not be located. *Keepins* and *Keepouts* are used for design purposes only and do not contribute to the actual PCA product.

The IDF supports several predefined *Keepin* and *Keepout* types. The specific characteristics of a particular type of *Keepin* or *Keepout* are defined through the use of properties. For example, a component placement *Keepout* can have a height specified to restrict the placement of component instances to those below a specific height. The *Keepin* and *Keepout* types can be extended by the user.

The region affected by a *Keepin* or *Keepout* is defined by an outline and zero or more voids. The outline defines the outer boundary of the region. The area inside the outline is affected by the *Keepin* or *Keepout*. A void defines an area inside the outline that is not affected by the *Keepin* or *Keepout*. *Keepins* and *Keepouts* may affect one or more layers of the *Board Part*, based on their type.

2.4.6 Physical Layers

The arrangement (stackup) and general characteristics of the *Physical Layers* of a *Board Part* can be specified to fully support designs where internal features are important. This includes designs with blind pins, blind and buried vias, and embedded components. It also allows the IDF to transfer conductor shapes on internal layers for reference purposes.

Several IDF features appear on, or affect individual conductive layers or combinations of conductive layers of the board or panel, and must include a layer name. When a feature requires a layer name, the name of a *Physical Layer* must be used. However, in the majority of cases, the internal characteristics of the *Board* or *Panel Part* are not significant and the *Physical Layers* do not need to be defined. Consequently, the IDF provides several predefined layer name aliases (Top, Bottom, Both, Inner, and All) to refer to the most common layers, without the need for defining the *Physical Layers*.

2.4.7 Footprints

A *Footprint* represents a set of board or panel features that are required to support the instance[4] of a component part or assembly. As the name implies, a *Footprint* "imprints" the features onto the board or panel that are associated with an instance. For example, the *Pads* to which the *Pins* of an *Electrical Instance* are soldered, exist in the actual *Board Part* and are associated with the *Electrical Instance*, so that if the instance is repositioned or removed, the *Pads* "go with it". Figure 2.6 illustrates *Footprints* contributed to the *Board Part* in a *Board Assembly*.

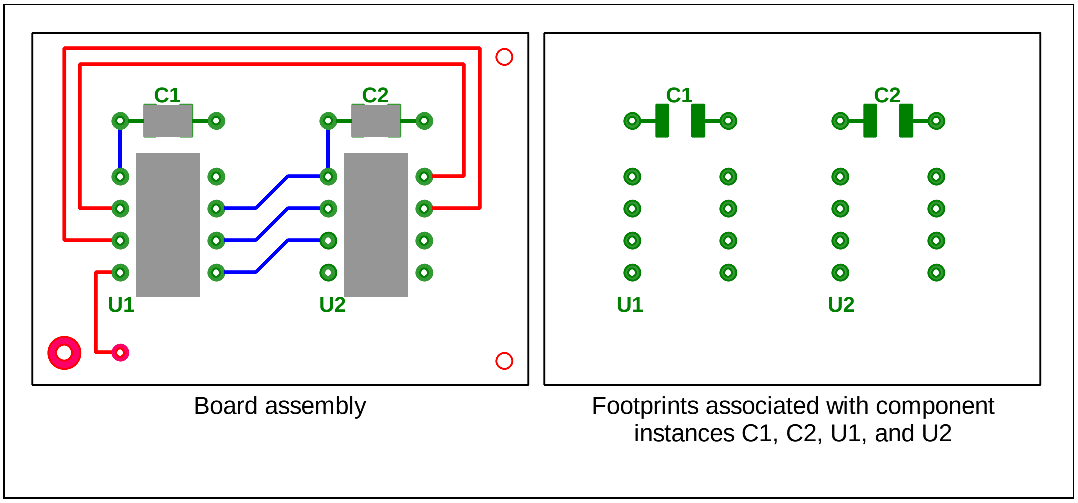

C1
C2
U1
U2
C1
C2
U1
U2
Board assembly
Footprints associated with component instances C1, C2, U1, and U2

Figure 2.6 Footprints in a Board Part

[4] In general, *Footprint* features may vary among instances, even instances of the same component or assembly. For example, a component may require different pad sizes when it is mounted on the top of the board than when it is mounted on the bottom. Therefore, each *Footprint* is associated with an instance in the IDF, as opposed to the part or assembly definition.

2.4.8 Figures

A *Figure* represents a named instance of a set of features that exist in a *Panel Part* or *Board Part*. They are similar to *Footprints* except they are not associated with a part instance. As with part instances in an assembly, *Figures* are located in a *Panel Part* or *Board Part* (via XY location, side, and rotation). Also like part instances, a *Figure* can be substituted for a definition of a corresponding *Figure* in the receiving system, based on the *Figure's* name.

A typical use for a *Figure* is to represent a mounting hole that may include the *Hole* itself, a *Pad*, and a *Keepout*. As illustrated in Figure 2.7, a mounting hole can be represented as a simple *Figure* in MCAD, and reference a complex *Figure* for the mounting hole defined in ECAD.

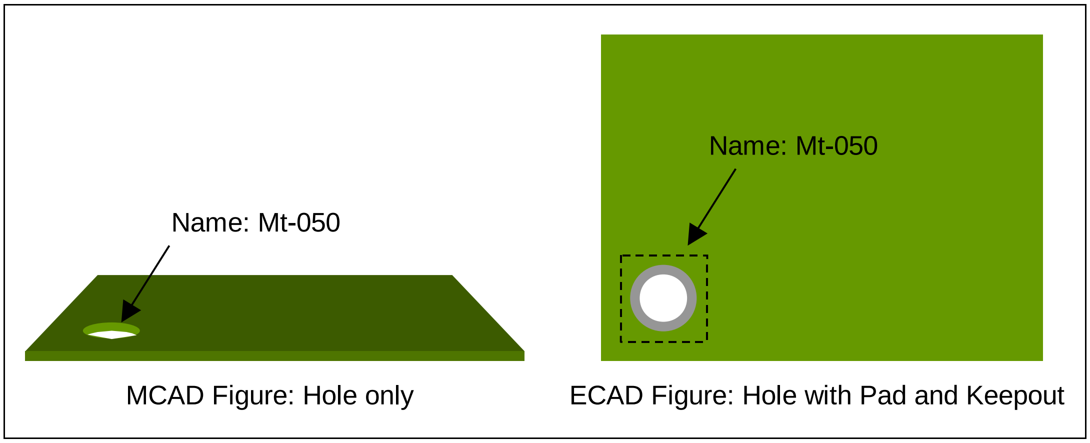


Figure 2.7 Mounting Hole Defined in MCAD and ECAD as a Figure

2.5 Geometry

All features in the IDF provide a graphical or physical representation, constructed from geometry. Geometry supported by the IDF consists of planar curves and planar areas. Curves are used in features such as *Traces*, *Graphics*, and *Annotations*. Areas are used for defining outlines in features such as *Pads*, *Filled Areas*, *Holes*, *Extrusions*, *Cutouts*, *Cavities*, *Keepins* and *Keepouts*.

Curves include *Circular Arc* (a segment of a circle*)*, *Polycurve* (a piecewise curve consisting of linear and circular arc segments), and *Polyline* (a piecewise linear curve). Examples of IDF curves are shown in Figure 2.8.

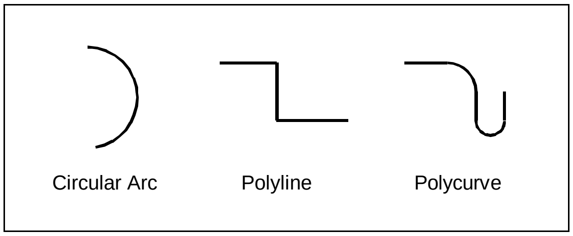


Figure 2.8 Geometric Curves

Areas include *Circle* (an area bounded by a circle), *Polycurve Area* (an area bounded by a closed *Polycurve*), and *Polygon* (an area bounded by a closed *Polyline*). Examples of IDF areas are shown in Figure 2.9.

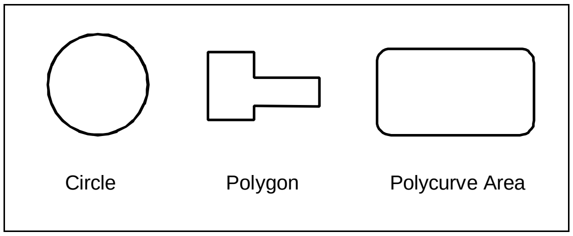


Figure 2.9 Geometric Areas

All IDF geometry entities can have specific line font and line color characteristics. In addition, areas can have fill color and curves can have line width and end styles. Line fonts, color, and end styles are shown in Figure 2.10.

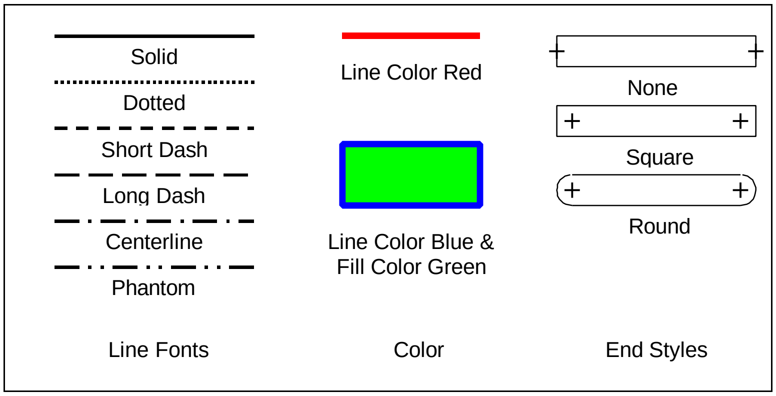


Figure 2.10 Line Fonts, Color, and End Styles

2.6 Analysis

The IDF supports the data that is necessary to interface to both thermal and structural analysis applications. In addition to shape information, which is required for analysis (defined by *Extrusions*, *Cutouts*, *Cavities*, and *Holes*), a part can have an associated *Material* description and/or *Thermal Model*. Properties may be used to augment thermal and electrical characteristics as necessary, to provide the desired level of detail for the analysis.

2.7 Extending the IDF Data Model

The IDF data model has been designed so that it can be extended to include data that is not explicitly defined in the model. This is done through user-extensible attributes, and user-defined properties.

Most parts, assemblies, and many features in the IDF have predefined types that further specify their general characteristics. For example, predefined *Pad* types allow a *Pad* to represent a connection site for a pin, via, bond wire, external connector contact, or a test probe. Each list of predefined types can be extended to include other uses for the entity.

IDF entities may also contain optional properties to further define the characteristics of the entity. Some entities have predefined properties. However, all entities can have user-defined properties as well. For example, if the cost of a part is significant to a particular design organization, a cost property can be defined and added to the part definition.

Extending the IDF data model through the use of user-extensible attributes and user-defined properties requires that individual IDF translators be designed to populate, recognize, and process them appropriately. At a minimum, translators should be designed to ignore user-extended attributes and user-defined properties that they do not recognize.

2.8 Change Control

The IDF provides basic support for “locking” data in a PCA design against modification, and for allowing incremental updates to PCA data between design applications.

To limit the modification of data within a PCA design, individual features and instances may be assigned a lock. The lock may be set by either MCAD or ECAD to indicate that the feature or instance should only be modified in the associated design application. For example, MCAD may lock an instance of a connector in a *Board Assembly* to indicate that ECAD should not move that connector instance. Only the application (MCAD or ECAD) that sets a lock should be able to remove it. If there is no lock on a feature or instance, it is free to be modified or locked by either application.

To enable incremental updates to a PCA design, it is necessary to establish a cross-reference between individual design objects within both the ECAD and MCAD applications. This requires a unique, persistent identifier for each design object, that both applications can use to recognize the object[5]. Most entities in the IDF have such an identifier. For parts, the identifier is the part name. For part and assembly instances, the identifier is the reference designator. For features, the identifier is the feature ID.

[5] Although the IDF supports the concept of unique identifiers for parts, instances, and features, very few design applications have internal support for unique identifiers other than component reference designators. Consequently, the design applications need to be extended by some type of convention in order to properly support incremental updates.

3 Data Representation

The IDF represents PCA data through a structured relationship of entities, attributes, and values[6]. This section describes the details of this representation.

3.1 Entities, Attributes, and Values

IDF data is described by entities. An entity consists of one or more attributes. An attribute has a corresponding value or set of values. A value can represent any of the data types described in Section 0.

3.1.1 IDs and Names

Every entity has an attribute that represents a unique entity ID. The entity ID uniquely identifies the entity within the scope of all entities in the file. A unique entity ID enables an IDF translator to explicitly identify a specific entity, establish entity relationships, or report on error conditions. Entity IDs must be created by the IDF translator, and are transient (they are not persistent from one translation to the next).

Some entities, including assemblies and parts, have an attribute that represents a user-specified name. The name uniquely identifies the entity within the scope of all like entities within the design (two component parts can't have the same name, for example). Entity names are obtained from the design application, and are persistent.

3.1.2 Contained and Referenced Entities

Most entities contain or reference other entities (via attribute values). Contained entities are "embedded" in the parent entity – the attribute that represents a contained entity includes all of the data for the contained entity as its value. Referenced entities are external to the parent entity – the attribute value that represents a referenced entity is a "pointer" (entity ID or name[7]) to the referenced entity.

A contained relationship is used when the entity is used exclusively by the parent entity (the extrusion for a board part, for example). A referenced relationship is used when the entity can be used by multiple entities (a component part definition, for example).

By definition, a contained entity can only be contained by one and only one parent entity. However, a contained entity <u>may be</u> referenced by one or more other entities.

[6] Every IDF file contains a File Header that provides data about the file. Although the File Header is not considered an entity itself, its representation is similar to that of an entity. The File Header is described in Section 5.

[7] Ideally, all references within the IDF data model should be via entity name (this would enhance the readability of the physical file). However, only certain entities have an entity name. Consequently, some entity references must be by entity ID. Basically, the rule for establishing entity references is that if the referenced entity has a name, it is referenced by the name. Otherwise, it is referenced by the entity ID.

The following sample file illustrates the use of contained and referenced entities.

```
Entity_A (
  Entity_ID (#1),             /* The # sign signifies a unique entity ID */
  Attribute_1 (Value),
  Attribute_2 (
    Entity_B (                                       /* Contained entity */
      Entity_ID (#2),
      . . .                                 /* Other Entity_B attributes */
    ); /* End of Entity_B */
  ),
  Attribute_3 (Value),
  Attribute_4 (#3),                  /* Referenced entity (by entity ID) */
  Attribute_5 ("XYZ")              /* Referenced entity (by entity name) */
); /* End of Entity_A */

Entity_C (
  Entity_ID (#3),
  . . .                                     /* Other Entity_C attributes */
); /* End of Entity_C */

Entity_D (
  Entity_ID (#4),
  Entity_Name ("XYZ"),
  . . .                                     /* Other Entity_D attributes */
); /* End of Entity_D */
```

3.2 Attribute Value Data Types

The following table describes the allowable data types for attribute values.

Data Type	Description
Boolean	The Boolean data type is used to represent an attribute value of True or False.
Integer	The Integer data type is used to represent a signed whole number value. The absolute magnitude of an integer value may not exceed (2^{31} - 1). Examples: 2, 23, 0, -15
Real	The Real data type is used to represent a floating point value. The absolute magnitude of a real value may not exceed the power of 308, given 15 digits of significance. Examples: 2.05, -267.12, 256.06D+4, 76.35D-2
String	The String data type is used to represent an ASCII character string value. A String value may only contain the printable ASCII characters (hexadecimal x20 thru x7E). Examples: “R13”, “PN-AF2”, “This is a note.”
Enum	The Enum data type is used to represent an attribute value that must be one of an enumerated list of possible values. An Enum value must begin with an alpha character (hexadecimal x41 thru x5A and x61 thru x7A) and may only contain alphanumeric characters and the underscore character (ASCII hexadecimal x30 thru x39, x41 thru x5A, x5F, and x61 thru x7A). Example: Units must be either “Inch” or “MM”

Data Type	Description
Ext_Enum	The Ext_Enum data type is used to represent a user (or vendor) extensible enumeration. Each Ext_Enum attribute has a list of predefined values. However, a translator implementation can provide additional values beyond those defined in the IDF Specification. An Ext_Enum value must begin with an alpha character (hexadecimal x41 thru x5A and x61 thru x7A) and may only contain alphanumeric characters and the underscore character (ASCII hexadecimal x30 thru x39, x41 thru x5A, x5F, and x61 thru x7A). Example: A Pin Type may be "Thru", "Blind", "Surface", or "Optical"
Entity	The Entity data type is used to represent a contained (embedded) entity. The attribute value contains all of the contents (attributes) of the contained entity. Example: A Trace entity contains a Polyline or Polycurve entity to represent its physical shape.
Ent_ID	The Ent_ID data type represents a positive integer, used to represent a unique entity ID, within the scope of all entities in the IDF file. The Ent_IDs within an IDF file do not need to be sequential. That is, there may be unused values or "holes". An Ent_ID value may be used as an entity reference. Example: #41
Ent_Name	The Ent_Name data type represents an ASCII character string, used to represent a unique entity name, within the scope of all like entities in the IDF file. An Ent_Name value may only contain the printable ASCII characters (hexadecimal x20 thru x7E). An Ent_Name value may be used as an entity reference. Example: "dip_14"
Ref_ID	The Ref_ID data type is used to represent an entity reference by entity ID. The attribute value contains the referenced entity's ID (Ent_ID data type). Example: #41
Ref_Name	The Ref_Name data type is used to represent an entity reference by entity name. The attribute value contains the referenced entity's name (Ent_Name data type). Example: "dip_14"
NValue	The NValue data type is used to represent a named value, where the value may be either an Integer, Real, String, Enum, Ext_Enum, or list of Integers, Reals, Strings, Enums, or Ext_Enums. The name portion of the NValue must begin with an alpha character (hexadecimal x41 thru x5A and x61 thru x7A) and only contain alphanumeric characters and the underscore character (ASCII hexadecimal x30 thru x39, x41 thru x5A, x5F, or x61 thru x7A). The value portion of the NValue must conform to the rules associated with the Integer, Real String, Enum, and Ext_Enum data types. Example: KI_Comp_Group ("U1", "U2", "U3")

3.3 Scalar, Array, and List Attribute Values

An attribute value may be a scalar value, a fixed array of values, a list of scalar values, or a list of fixed array values. An array is a fixed number of values, whereas a list is a variable number of values. Both arrays and lists are specified using array subscripts.

In Section 5 and 6, the entity attribute tables specify arrays and lists, using subscripts as follows:

- If the number of attribute values is fixed, the attribute represents an array, and an integer constant is used to specify the subscript (for example, Attribute[2]).
- If the number of attribute values is not fixed, the attribute represents a list and the subscript specifies the lower and upper bounds of the list. For example, [0..3] means that there can be zero to three values, [0..?] means that there can be zero or more values, [1..?] means that there must be at least one value.
- A list of arrays is represented with two subscripts – the first subscript defines the number of arrays and the second subscript defines the size of the array. For example, Attribute[1..?][2] specifies one or more two-element arrays.

The following table provides several examples of attribute notation:

Attribute Name	Description
Part_Number	A single String value that represents a part’s part number.
XY_Loc[2]	A fixed array of two real numbers that specify the XY location of an entity.
Features[0..?]	A list of zero or more feature entities.
XY_Pts[2..?][2]	A list of two or more XY points, where each point is defined as a two element array.

3.4 Attribute and Entity Constraints

For most entities, there are constraints on the values that can be assigned to its attributes (attribute constraints), and to the relationship of attribute values within an entity (entity constraints). An example of an attribute constraint is: the default value of the Line_Font attribute for a *Circular_Arc* is Solid. An example of an entity constraint is: a void within a *Filled Area* must be located within the outer boundary of the *Filled Area*.

Attribute and entity constraints are formalized in IDF 4.0. The constraints are defined and described in the following table:

Constraint Name	Applicable Data Types	Usage	Description
Constant	Boolean Integer Real String Enum	Constant (Value)	The attribute value must be equal to the Constant constraint value.
Default	Boolean Integer Real String Enum Ext_Enum	Default (Value)	If the attribute is not present in the IDF file, the attribute value must be set to the Default constraint value.

Constraint Name	Applicable Data Types	Usage	Description
EList	Enum Ext_Enum	EList (Value_1, Value_2, . . ., Value_n)	The acceptable set of values that can be used for an Enum or Ext_Enum attribute value.
Informal	All	Informal (Informal description of the attribute constraint.)	A complex constraint that can't be represented using a combination of the other attribute constraints. The constraint value is represented as an informal set of sentences (No two adjacent points in a polygon can be coincident, for example). Informal can also be used as an entity constraint.
Optional	All	Optional (Interpretation of the missing attribute.)	The attribute is optional, it is not a required part of the entity definition. The constraint value is used to further define how to interpret the missing attribute. If an attribute has an optional constraint, it can not have a default constraint. Optional implies that if the attribute is not present, its value is to be ignored (there is no default value).
NList	NValue	NList (Name, Name, . . .)	The list of predefined names (and corresponding values) that may be contained by the attribute.
Range	Integer Real Entity Constraint	Range (Relational Statement)	The attribute value must be within the range specified by the relational statement, which is constructed as an ANSI C relational statement. Operator precedence is as defined in ANSI C. The following operators are allowed: > Greater than >= Greater than or equal to < Less than <= Less than or equal to == Equal to != Not equal to && And !! Or (Left parenthesis) Right parenthesis Range can also be used as an entity constraint.
RList	Entity Ref_ID Ref_Name	RList (Entity, Entity, . . .)	The valid list of entity types for either contained or referenced entities. The constraint value must be a list of one or more entity types, as defined in Section 6.
Where_Used	None	Where_Used (Entity, Entity, . . .)	The list of entity types that may contain or reference the entity. The value of “Independent” indicates that the entity may exist in the IDF file without any other entities containing or referencing it. If the parent to child relationship is referenced, as opposed to contained, it will be noted as such. Where_Used can only be used as an entity constraint.

4 File Format

The IDF file format is an ASCII format, based on a context free grammar, and designed to be suitable for processing via C or C++ translator implementations. Although in general, the IDF file has a free format[8], specific formatting rules have been defined to enhance the human readability of the file, and facilitate processing by a script language such as PERL.

4.1 Organization

The IDF file is organized into the following four sections:

Section Name	Description
IDF_Header	The IDF_Header is the first section in the IDF file and is always required. It consists of attributes containing information about the IDF file itself – when it was created, by whom, and a summary of the contents.
Assemblies	The Assemblies section contains all the assembly entities in the IDF file, and follows the IDF_Header section. It is only required if assembly entities are present. If an IDF file contains an Assemblies section, it represents a board or panel design. Typically, the Assemblies section will only contain a single board assembly.
Parts	The Parts section contains all the part entities in the IDF file, and follows the Assemblies section, if it exists. It is only required if part entities are present. If an IDF file contains a Parts section, but no Assemblies section, it represents a parts library.
Ref_Entities	The Ref_Entities section contains all of the entities that are referenced from the Assemblies and Parts sections (with the exception of the part and assembly entities that are referenced by the instance entities contained in an assembly or part). It is always the last section in the file, and is only required if the file contains referenced entities.

If there are no entities for a particular section, the entire section may be omitted from the file, with the exception of the IDF_Header section, which must always be contained in the file.

[8] A free format means that white space (blanks, tabs, new lines, and comments) is ignored, and can occur anywhere in the file.

4.2 Tokens

An IDF file is comprised of the following tokens:

Token	Description
Keywords	Keywords are predefined section, entity, and attribute names, plus the name portion of a named value attribute. All keywords in the IDF are comprised of a combination of alphanumeric characters and the underscore character (ASCII hexadecimal x30 thru x39, x41 thru x5A, x5F, and x61 thru x7A).
Values	Values represent the actual IDF data, and are contained within an attribute. The allowable attribute value data types are defined in Section 0.
Comma	The comma is used to delimit attributes within an entity, and values within an attribute.
Semicolon	The semicolon is used to specify the end of a section or an entity.
Parentheses	Opening (left) and closing (right) parentheses are used to enclose entities, attributes, and attribute values.
New Line	The new line character is used to enhance the readability of the file.
Comments	Comments are enclosed by opening (/*) and closing (*/) comment characters.

The following example illustrates how these tokens are used in an IDF file:

```
Section (
  Entity (
    Attribute_A (Value),
    Attribute_B (Value_1, Value_2),
    Attribute_C (
      Value_1, Value_2,
      Value_3, Value_4
    )
  ); /* End Entity */
); /* End Section */
```

4.3 Format

The format is defined by the rules for how the tokens can be arranged (grammar).

4.3.1 Formatting Rules

The following represents the general formatting rules for the IDF file:

Blanks, Tabs, and Comments

- Blanks, tabs, and comments may appear anywhere in the file, except within a token. However, blanks are allowed within a comment or a String attribute value.
- Comments can span multiple lines, but may not be nested.

Parenthesis

- All of the entities for a section, attributes for an entity, values for an attribute, and values within the value portion of a named value attribute, must be enclosed within parenthesis.

Delimiters

- A comma must be used to delimit each attribute within an entity, with the exception of the last attribute in the entity.
- A comma must be used to delimit each value within an attribute, with the exception of the last value in the attribute.
- A comma must be used to delimit each value within the value portion of a named value attribute, with the exception of the last value in the list.
- A semicolon must follow the closing parenthesis of a section and an entity. This "ends" the section or entity.

New Lines

- A new line must follow both a section name and its corresponding left parenthesis, and an entity name and its corresponding left parenthesis. This constitutes the "opening" line of the section or entity.
- A new line must follow the semicolon ending both a section and an entity. This constitutes the "closing" line of the section or entity.
- If an attribute represents a contained entity(s), a named value, or contains more than one value, and more than one line is required to format the attribute and its corresponding values[9], the attribute must be opened and closed similar to a section or entity. In this case:
  - – A new line must follow the attribute name and its corresponding left parenthesis.
  - – A new line must follow the last value.
- A new line must follow the comma delimiting an attribute within an entity.

[9] Technically, the attribute name and corresponding values of a multi-value attribute can be formatted on a single line. However, to make the file more human-readable, new lines may be inserted after the comma delimiters. The following guidelines should be used to maintain consistency for IDF files:

- If the attribute value represents a fixed array (XY location of a part instance, for example), such that the total number of characters required to format the attribute name and values will fit on a single printed line, the attribute name and corresponding values should be formatted on a single line.
- If the attribute value represents a list of Strings, one String should be formatted per line.
- If the attribute represents a list of named values, each name and value pair should be formatted on a separate line (if the name value pair has multiple values, it may require multiple lines to be formatted).
- If the attribute value represents a list of fixed arrays (XY points of a polyline, for example), the values for each array in the list, should be formatted per line.

- A new line must follow the closing right parenthesis of the last attribute within an entity.
- A new line must follow the semicolon ending both a section and an entity.
- A new line may appear after a new line (resulting in a blank line).
- A new line may appear after the comma delimiting a value in an attribute, or a value in the value portion of a named value attribute. However, a new line may not appear between a value and the comma delimiter.
- A new line may not appear between a section, entity, attribute, or named value keyword and its corresponding left parenthesis.
- A new line may not appear between a closing right parenthesis and comma delimiter.
- A new line may not appear between a closing right parenthesis and an ending semicolon.
- A new line may not appear within a token, other than a comment.

Case Sensitivity

- Section, Entity, and Attribute keywords, Boolean, Enum, and Ext_Enum attribute values, and the name portion of a NValue attribute are case insensitive.
- String, Ent_Name, and Ref_Name attribute values are case sensitive.

4.3.2 Formatting Example

The following example illustrates the formatting rules.

```
Section (
  Entity_1 (
    Attribute_A (Value),                       /* Single value attribute */
    Attribute_B (Value_1, ..., Value_n),        /* Multi-value attribute */
    Attribute_C (          /* Multi-value attribute - one value per line */
      Value_1,
      . . .,
      Value_n
    ),

    Attribute_D (                               /* Multi-value attribute */
      Value_1A, Value 1B,                   /* List of two-element array */
      . . .,
      Value_nA, Value_nB
    ),

    Attribute_E (                          /* Contained entity attribute */
      Entity_2 (
      . . .,                                      /* Entity_2 attributes */
      );                                              /* End of Entity_2 */
    ),
    Attribute_F (                               /* Named value attribute */
      Name_1 (Value),                                    /* Single value */
      Name_2 (Value)
    ),

    Attribute_G (                               /* Named value attribute */
      Name (Value_1, ..., Value_n)                        /* Multi-value */
    ),

    Attribute_H (                               /* Named value attribute */
      Name (                         /* Multi-value – one value per line */
        Value_1,
        . . .,
        Value_N
      )
    ),

    Attribute_I (                               /* Named value attribute */
      Name (                   /* Multi-value – multiple values per line */
        Value_1, Value_2,
        . . .,
        Value_m, Value_n
      )
    )
  );                                                  /* End of Entity_1 */
);                                                     /* End of Section */
```

4.3.3 Attribute Data Type Formatting Rules

The following table defines the specific formatting rules for each attribute data type.

Data Type	Example	Description
Boolean	Attribute (“True”)	A Boolean value must be either “True” or “False”. The value must be enclosed in double quotes.
Integer	Attribute (23)	An Integer value is an encoded integer.
Real	Attribute (0.25) Attribute (2.50D+02)	A Real value is an encoded real number. A formatted Real value must preserve at least six significant digits. Exponential notation is allowed.
String	Attribute ("Text string") Attribute ("Embedded ""Double Quotes""") Attribute (“Text with left (or right) parenthesis”)	A String value must be enclosed in double quotes. If the text string contains a double quote, it must be escaped by a double quote.
Enum	Attribute (“Inch”)	An Enum value must be represented exactly as it is defined in the EList attribute constraint. The value must be enclosed in double quotes.
Ext_Enum	Attribute (“Optical”)	An Ext_Enum value is formatted as an Enum value.
Entity		A contained entity is formatted according to the rules described above.
Ent_ID	Attribute (#45)	The Ent_ID value is formatted as an Integer data type, with a preceding # sign.

Data Type	Example	Description
Ent_Name	Attribute ("DIP_14")	The Ent_Name value is formatted as a String data type.
Ref_ID	Attribute (#45)	The Ref_ID value is an entity ID and is formatted as an Ent_ID value.
Ref_Name	Attribute ("DIP_14")	The Ref_Name value is an entity name and is formatted as an Ent_Name value.
NValue	Attribute (Name_1 (Value), Name_2 (Value_1, Value_2, . . ., Value_n))	An NValue value has two components, a name and a value. The name is formatted as a section, entity, or attribute name. The value(s) is formatted according to its data type (values are formatted as either, Integer, Real, String, Enum, or Ext_Enum) and is enclosed within parenthesis. Multiple values are delimited with a comma.

5 IDF_Header

The IDF_Header contains information about the IDF file and the PCA design data it contains. It consists of a list of attributes defined in the following table:

Attribute Name	Type	Constraints	Descripton
Version	String	Constant (“V4.0”)	IDF Version of the file.
Creation_Date_Time	String	- - -	Date and time the file was created, in the format (yyyy/mm/dd.hh:mm:ss). The value is specified in UTC (Coordinated Universal Time). For example: “1998/02/13.02:42:34”
Owner_Name	String	Optional (Owner_Name is provided at the discretion of the sending user.)	Name of the person who created the file, or the person who is the point of contact for questions regarding the content of the file.
Owner_Phone	String	Optional (Owner_Phone is provided at the discretion of the sending user.)	Phone number for the Owner.
Owner_EMail	String	Optional (Owner_EMail is provided at the discretion of the sending user.)	E-mail address for the Owner.
Source_App_Type	Ext_Enum	EList(“ECAD”, “MCAD”, “MCAE”, “Dwg”, “Mfg”)	Type of application that created the file.
Source_App_Vendor	String	- - -	Vendor of the Source Application. For example: “Mentor Graphics Corporation” “Parametric Technology Corporation”
Source_App_Name	String	- - -	Name of the Source Application that created the design data. For example: “Board Station” “Pro/Engineer”
Source_App_Version	String	- - -	Version of the Source Application. For example: “V8.2.1”
IDF_Tx_Name	String	- - -	Name of the IDF translator that created the file.
IDF_Tx_Version	String	- - -	Version of the IDF translator.

Attribute Name	Type	Constraints	Descripton
Entity_Count[12]	NValue	NList (Elec_Part_Defs, Elec_Part_Insts, Mech_Part_Defs, Mech_Part_Insts, Board_Part_Defs, Board_Part_Insts, Board_Assy_Defs, Board_Assy_Insts, Panel_Part_Defs, Panel_Part_Insts, Panel_Assy_Defs, Panel_Assy_Insts)	Summary of part and assembly definition and instance counts for the IDF file. The predefined names represent each type of part and assembly definition and instance that is to be counted. Each predefined name and value must be provided, even if the count is equal to zero. The corresponding values are Integers that represent the count for the associated definition or instance.
Comp_Part[0..19]	Enum	EList (“Annotation”, “Cavity”, “Circle”, “Circular_Arc”, “Cutout”, “Electrical_Part”, “Extrusion”, “Leader”, “Material”, “Mechanical_Part”, “Pin”, “Polycurve”, “Polycurve_Area”, “Polygon”, “Polyline”, “Text”, “Thermal_Model”, “Thermal_CV”, “Thermal_RV”) Optional (No Comp_Part content in the file.)	Content summary for the Comp_Part content category. See Section 5.1 below.
Board_Part[0..24]	Enum	EList (“Annotation”, “Board_Part”, “Cavity”, “Circle”, “Circular_Arc”, “Cutout”, “Extrusion”, “Figure”, “Filled_Area”, “Footprint”, “Graphic”, “Hole”, “Keepin”, “Keepout”, “Leader”, “Material”, “Pad”, “Physical_Layer”, “Polycurve”, “Polycurve_Area”, “Polygon”,	Content summary for the Board_Part content category. See Section 5.1 below.

Attribute Name	Type	Constraints	Descripton
		“Polyline”, “Text”, “Trace”) Optional (No Board_Part content in the file.)	
Board_Assy[0..5]	Enum	EList (“Board_Assembly”, “Board_Part_Instance”, “Electrical_Part_Instance”, “Mechanical_Part_Instance”, “Sublayout”) Optional (No Board_Assy content in the file.)	Content summary for the Board_Assy content category. See Section 5.1 below.
Panel_Part[0..23]	Enum	EList (“Annotation”, “Cavity”, “Circle”, “Circular_Arc”, “Cutout”, “Extrusion”, “Figure”, “Filled_Area”, “Footprint”, “Graphic”, “Hole”, “Keepin”, “Keepout”, “Leader”, “Material”, “Pad”, “Panel_Part”, “Polycurve”, “Polycurve_Area”, “Polygon”, “Polyline”, “Text”, “Trace”) Optional (No Panel_Part content in the file.)	Content summary for the Panel_Part content category. See Section 5.1 below.
Panel_Assy[0..6]	Enum	EList (“Board_Assembly_Instance”, “Electrical_Part_Instance”, “Mechanical_Part_Instance”, “Panel_Assembly”, “Panel_Assembly_Instance”, “Panel_Part_Instance”) Optional (No Panel_Assy content in the file.)	Content summary for the Panel_Assy content category. See Section 5.1 below.
Default_Units	Enum	EList (“Inch”, “MM”)	The default length units for the file. If a part, assembly, figure, or footprint does not have a units attribute, the default units should be used.

Attribute Name	Type	Constraints	Descripton
Min_Res	Real	Range (Min_Res > 0.0)	The minimum resolution (in the default units) for the geometric content of the file. If the distance between two points is less than or equal to the Min_Res value, the two points are to be considered coincident.
Notes[0..?]	String	Optional ()	Miscellaneous notes that describe the contents of the file, or issues surrounding why the file needs to be exchanged.

5.1 Content

The content of an IDF file is summarized in the IDF Header, according to a set of five content categories:

Content Category	Description
Comp_Part	All of the entities that are subordinate[10] to electrical and/or mechanical parts.
Board_Part	All of the entities that are subordinate to the board part(s).
Board_Assy	All of the entities that are subordinate to the board assembly(s).
Panel_Part	All of the entities that are subordinate to the panel part(s).
Panel_Assy	All of the entities that are subordinate to the panel assembly(s).

Each content category has an associated content summary attribute (Comp_Part, Board_Part, Board_Assy, Panel_Part, and Panel_Assy) that specifies the type of entities that are included in the associated content category. Each attribute represents an enumerated list of entities that can be used in that content category. The value is the list of entity keywords that are actually used for that content category in the file.

5.2 Sample IDF Header

```
IDF_Header (                                /* Start IDF Header Section */

  Version (“V4.0”),
  Creation_Date_Time (“1998/06/05.10:00:00”),
  Owner_Name (“Tom Makoski”),
  Owner_Phone (“513-528-5059”),
  Owner_EMail (“Tom.Makoski@Intermedius.com”),
  Source_App_Type (“ECAD”),
```

[10] A content category includes all of the entities that are either contained or referenced from the part or assembly entity(s), and all of the entities that are contained or referenced by those entities, and so on. For example, if a board part has a figure that contains a hole, the Board_Part content category includes both the figure and the hole. An exception to this rule is that the instance to part relationship and instance to assembly relationship is not traversed when determining the content.

```
  Source_App_Vendor (“XYZ Design Co.”),
  Source_App_Name (“PCB DesignWorks”),
  Source_App_Version (“V3.2”),
  IDF_Tx_Name (“PCB Mechanical Interface”),
  IDF_Tx_Version (“V2.1”),
  Entity_Count (
    Elec_Part_Defs (3),
    Elec_Part_Insts (5),
    Mech_Part_Defs (1),
    Mech_Part_Insts (1),
    Board_Part_Defs (1),
    Board_Part_Insts (1),
    Board_Assy_Defs (1),
    Board_Assy_Insts (0),
    Panel_Part_Defs (0),
    Panel_Part_Insts (0),
    Panel_Assy_Defs (0),
    Panel_Assy_Insts (0)
  ),
  Comp_Part (
    “Annotation”,
    “Extrusion”,
    “Pin”,
    “Polygon”,
    “Polyline”,
    “Text
),
  Board_Part (
    “Annotation”,
    “Cavity”,
    “Circle”,
    “Cutout”,
    “Extrusion”,
    “Figure”,
    “Filled_Area”,
    “Footprint”,
    “Graphic”,
    “Hole”,
    “Keepin”,
    “Keepout”,
    “Leader”,
    “Pad”,
    “Physical_Layer”,
    “Polycurve_Area”,
    ‘Polygon”,
    “Polyline”,
    “Text”,
    “Trace”
  ),
  Board_Assy (
    “Board_Part_Instance”,
    “Electrical_Part_Instance”,
    “Mechanical_Part_Instance”,
    “Sublayout”
  ),
Default_Units (“Inch”),
  Min_Res (0.0005),
  Notes (
    “This IDF file represents a traditional PCA design, including ”,
    “the board part definition and the component part definitions.”
  )
); /* End IDF_Header Section */
```

6 Entity Descriptions

This section provides a detailed description for each IDF entity. Each entity description begins with a brief overview of the entity, followed by an attribute table listing the entity's attribute names, data types, constraints, and description. Entity constraints are included at the bottom of the attribute table. Usage notes follow the attribute table.

The entities are organized alphabetically by name within this section.

6.1 Common Entity Attributes

There are several attributes (Entity_ID, Feature_ID, Lock, Layer, Type, and Properties) that are common to many or all entities. The following paragraphs describe each of these attributes in detail. The attributes are only briefly described in the entity descriptions.

6.1.1 Entity ID

Every IDF entity has a unique entity identifier, represented by the Entity_ID attribute. The Entity_ID attribute is always the first attribute in the entity. It must be created by the IDF translator, and be unique within the scope of all entities in the IDF file. However, they do not need to be sequential (holes are allowed). It is not intended to be a persistent identifier. It is used to establish entity references within the file, and can also be used to create traceable error messages.

6.1.2 Feature ID

Each IDF entity that represents a product or design feature (Annotation, Cavity, Cutout, Extrusion, Filled_Area, Graphic, Hole, Keepin, Keepout, Pad, Pin, or Trace) has a unique feature identifier, represented by the Feature_ID attribute. The feature ID is a global, unique (within the scope of all like features within the design), and persistent identifier, that is used by each design application, to identify each occurrence of a feature.

The feature ID allows for the incremental update of a PCA design via IDF translation. For example, when a tooling hole is moved in MCAD, an IDF file can be created with the board part and the tooling hole. On input to ECAD, the translator can match the IDF tooling hole with the corresponding ECAD tooling hole (via the feature ID), and make the necessary modifications. Without the feature ID, the translator would not know whether the tooling hole is new, or simply a modification of an existing tooling hole.

The feature ID is a required attribute. However, there will be cases in which a design application is unable to assign unique feature IDs. In this case, the default value of "Unassigned" must be used, which simply denotes that the feature has not been assigned a feature ID.

Similar to feature IDs, Figure entities have Figure IDs and Sublayouts have Sublayout IDs so that each occurrence of a figure or sublayout can be uniquely identified among design applications.

6.1.3 Lock

The Lock attribute provides a mechanism for a design organization to specify who has the authority to change an existing entity[11].

The value of the Lock attribute is an extended enumeration (Ext_Enum) that specifies which organization has locked the entity from being modified. The predefined values are:

Value	Description
ECAD	The electrical design organization has locked the entity.
MCAD	The mechanical design organization has locked the entity.
None	The entity is not locked (the default value). It is free to be modified by any design application.

6.1.4 Layer

There are many IDF entities that either appear on, or affect one or more physical layers of the board or panel. In each case, there is an attribute (typically named Layer, but occasionally named something else such as Span) to specify the layer or layers associated with the entity. The value of this attribute is a layer name.

If it is necessary to represent information on internal layers of the board (embedded printed components, for example), a stackup of physical layers must be defined and associated with the board. Layer names are defined in the Physical_Layer entities. Generally, however, a detailed representation of the internal layers of the board is not required, and the physical layers are not included. In this case, the pre-defined layer names (Top, Bottom, Both, Inner, and All) can be used.

Consequently, there are two ways to specify the layer or layers associated with an entity:

1. The layer name defined in a Physical_Layer entity.
2. The name of a physical layer alias.

6.1.5 Type

The Type attribute is used to further specify the entity type (a pin is either a thru, blind, or surface type, for example). If an entity has a Type attribute, it is represented as an extended enumeration (Ext_Enum), so that its use can be extended by the user, if necessary.

[11] Not all design applications can support entity locking. Some applications may have the ability to set and/or store a lock status with a design object, but are not able to prevent it from being modified in the application.

6.1.6 Properties

The Properties attribute is used to further define the meaning or characteristics of an entity. It is defined as a list of named values (NValue attribute data type). Each entity may have predefined properties associated with it (as specified by the NList attribute constraint). Every entity may also have one or more user-defined properties associated with it.

The IDF has a set of predefined property names and corresponding values (see table below). This set can be extended, simply by providing additional property names and their corresponding values, within the restrictions of the NValue data type.

The following table defines the list of predefined IDF properties.

Property Name	Data Type	Applicable Entities	Description / Usage
Electrical Component Characteristics			
EL_Capacitance	Real	Electrical_Part, Electrical_Part_Instance	Electrical capacitance (in farads).
EL_Power_Nominal	Real	Electrical_Part, Electrical_Part_Instance	Nominal electrical power rating (in watts).
EL_Power_Min_Max[3]	Real	Electrical_Part, Electrical_Part_Instance	Minimum, nominal, and maximum electrical power rating (in watts). All three values are required. The values are constrained as follows: Minimum <= Nominal <= Maximim If only the nominal power rating is available, the EL_Power_Nominal property value should be used.
EL_Resistance	Real	Electrical_Part, Electrical_Part_Instance	Electrical resistance (in ohms).
EL_Tolerance	Real	Electrical_Part, Electrical_Part_Instance	Tolerance for the nominal power rating, expressed as a percent (0.10, for example). The EL_Tolerance property should only be used in conjunction with the EL_Power_Nominal property.
Keepin Values			
KI_Comp_Group[0..?]	String	Keepin	List of component reference designators that specifies the components that must be placed in the Keepin region.
KI_Comp_Height	Real	Keepin	A maximum component placement height value for the Keepin region (in the units defined for the associated part). Components whose heights are greater than this value cannot be placed within the region.

Property Name	Data Type	Applicable Entities	Description / Usage
KI_Comp_Rot[0..?]	Integer	Keepin	List of component placement rotation values (in degrees) that specify the allowable component placement rotation, for components that are placed within the Keepin region.
Keepout Values			
KO_Comp_Height	Real	Keepout	A maximum component placement height value for the Keepout region (in the units defined for the associated part). Components whose heights are greater than this value cannot be placed within the region.
Pin Characteristics			
Pin_Area	Real	Electrical_Part, Electrical_Part_Instance	Area (in the part units^2) of the cross section for a single pin. It is assumed that all pins on an electrical part have the same (or approximately the same) cross section.
Pin_Count	Integer	Electrical_Part, Electrical_Part_Instance	The number of pins for the electrical part. This value may be redundant given that the electrical part may contain the set of pin entities. However, the pin entities are optional, and may not be present in the file.
Pin_Layout	Ext_Enum	Electrical_Part, Electrical_Part_Instance	Pin layout or configuration with respect to the electrical part. Acceptable values are: “DIP” Dual in-line pins (on two sides of the component) “Edge” Pins are only on one side “Grid” Pins represent a grid on the bottom of the component “Quad” Pins are on four sides
Pin_Length	Real	Electrical_Part, Electrical_Part_Instance	Length (in the part units) of a single pin. It is assumed that all pins on an electrical part have the same (or approximately the same) length.
Thermal Analysis Properties			
TH_Air_Gap	Real	Electrical_Part, Electrical_Part_Instance, Mechanical_Part, Mechanical_Part_Instance	Thermal air gap (in the part units). The distance between the mounting surface of the board or panel, and the bottom surface of the component body.
TH_Capacitance	Real	Electrical_Part, Electrical_Part_Instance	Thermal capacitance (in kJ/°C).
TH_Heat_Dis_Nominal	Real	Electrical_Part, Electrical_Part_Instance, Mechanical_Part, Mechanical_Part_Instance	Nominal thermal heat dissipation (in watts).

Property Name	Data Type	Applicable Entities	Description / Usage
TH_Heat_Dis_Min_Max[3]	Real	Electrical_Part, Electrical_Part_Instance, Mechanical_Part, Mechanical_Part_Instance	Minimum, nominal, and maximum thermal heat dissipation (in watts). All three values are required. The values are constrained as follows: Minimum <= Nominal <= Maximim If only the nominal heat dissipation is available, the TH_Heat_Dis_Nominal property value should be used.
TH_C2B_Resistance	Real	Electrical_Part, Electrical_Part_Instance	Case to board thermal resistance (in °C/watts).
TH_J2B_Resistance	Real	Electrical_Part, Electrical_Part_Instance	Junction to board thermal resistance (in °C/watts).
TH_J2C_Resistance	Real	Electrical_Part, Electrical_Part_Instance	Junction to case thermal resistance (in °C/watts).
Miscellaneous Properties			
Avg_M2V_Ratio	Real	Board_Part, Panel_Part	Average metal to volume ratio for the board or panel.

The following example illustrates the use of the Properties attribute for an electrical part.

```
Electrical_Part (
  Entity_ID (#21),
  Part_Name ("Sample_Part"),
  Units (“Inch”),
  . . .                              /* Other Electrical Part attributes */
  Properties (
    TH_Heat_Dis_Min_Max (0.50, 0.525, 0.55),
    TH_J2B_Resistance (0.05),
    TH_J2C_Resistance (0.025),
    Cost (25.50)                                /* User defined property */
  )
); /* End of Electrical_Part entity */
```

6.2 Coordinate Systems

The following entities have their own coordinate system, in which subordinate entities are defined:

- Board_Assembly
- Board_Part
- Electrical_Part
- Mechanical_Part
- Panel_Assembly
- Panel_Part
- Figure
- Footprint
- Pad
- Hole

Each coordinate system is a right hand coordinate system, as illustrated in Figure 6.1.

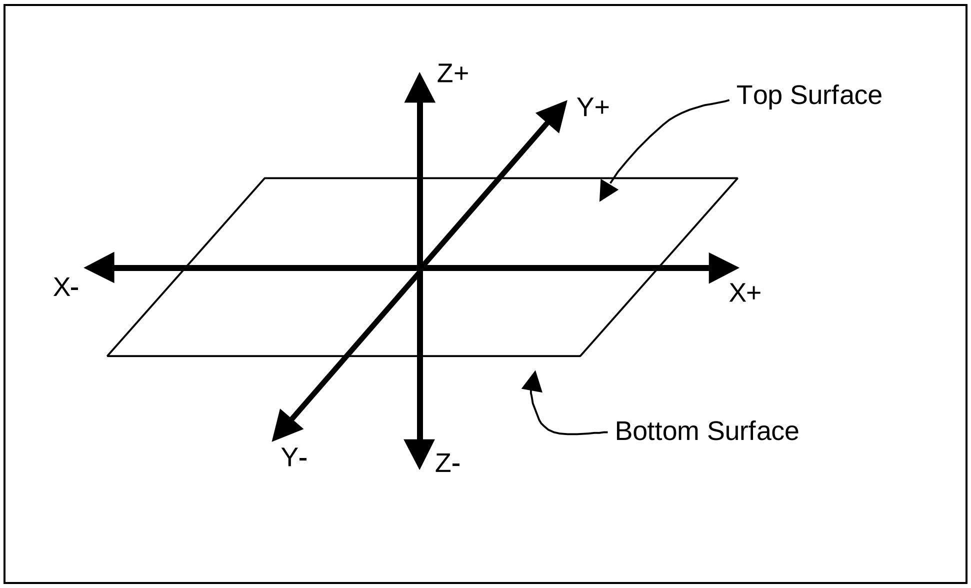


Figure 6.1 IDF Coordinate Systems

6.3 Annotation

The Annotation entity is used to represent design notes and other miscellaneous drawing information. Annotations are not intended to represent physical or design features of a part. They are intended to provide supplemental information to enhance the description of the physical and/or design features of a part.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Feature_ID	String	Default (“Unassigned”)	Unique feature identifier.
Lock	Ext_Enum	EList (“MCAD”, “ECAD”, “None”) Default (“None”)	Feature lock status.
Level	String	Optional (There is no grouping or association for the annotation.)	Level that the annotation is associated with.
Entities[1..?]	Entity	RList (Circle, Circular_Arc, Leader, Polycurve, Polycurve_Area, Polygon, Polyline, Text)	List of entities that represent the annotation.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Board_Part, Figure, Footprint, Panel_Part)			

Usage Notes:

- The entities that represent the annotation are defined in the coordinate system of the entity that contains the annotation.
- The level attribute provides a general mechanism for differentiating annotations. For example, a set of design notes that apply to features on the top of the board may be associated with a level corresponding to a named layer within an ECAD application. This association can be preserved, by assigning the ECAD layer name to the level attribute for each annotation that is associated with the layer.
- It is assumed that in MCAD, an annotation is located at Z=0.0 within the part that contains it. However, the MCAD application may use Z-planes, layers, visibility groups, or other methods to differentiate the different levels of annotations.

6.4 Board_Assembly

The Board_Assembly entity is used to represent a printed circuit assembly (PCA) consisting of a board part, component instances, and sublayouts. In addition, a board assembly may represent a set of one or more design variants.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Assy_Name	Ent_Name	- - -	Name of the board assembly.
Part_Number	String	Optional (Part number is not known.)	Part number for the board assembly.
Units	Enum	EList (“Inch”, “MM”, “Global”) Default (“Global”)	Length units for defining the location of instance entities within the board assembly. Global means that the units are equal to the default units specified in the IDF Header.
Type	Ext_Enum	EList (“Flex”, “MCM”, “Rigid_Flex”, “Traditional”, “Unspecified”)	Board assembly type.
Variants[0..?]	String	Optional (Board assembly represents a single design variant.)	List of design variants.
Board_Inst	Entity	RList (Board_Part_Instance) Optional (Board instance is not required when only updating component instance locations.)	Board part instance.
Comp_Insts[1..?]	Entity	RList (Electrical_Part_Instance, Mechanical_Part_Instance)	List of electrical and mechanical part instances.
Sublayouts[0..?]	Entity	Optional (There are either no sublayouts, or they are not required for the translation.)	List of sublayouts.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Informal (The board part instance must be located at XY_Loc = 0.0, 0.0 and Rotation = 0.0.) Informal (If the board assembly has a list of design variants, each component instance contained within the board assembly must specify the variants that it is used in.) Where_Used (*Independent,* Board_Assembly_Instance (*Ref. by Name*))			

Usage Notes:

- The board assembly has its own coordinate system in which its part instances are located.
- If a board assembly within an ECAD application has a set of design variants, but only one variant needs to be output to the IDF, the IDF board assembly can either specify a single variant (and ensure that each component instance is included in that variant), or not specify any variants.

- The board types are defined as follows:

Type	Description
Flex	A PCA whose substrate is flexible.
MCM	A PCA that is a packaged multichip module.
Rigid_Flex	A PCA whose substrate has both rigid and flexible portions.
Traditional	A PCA whose substrate is an etched laminate.
Unspecified	PCA type is unkown or not required.

6.5 Board_Assembly_Instance

The Board_Assemby_Instance entity is used to represent the occurrence of a board assembly in a panel assembly.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Assy_Name	Ref_Name	RList (Board_Assembly)	Name of the board assembly that is instanced.
Refdes	String	Default (“Unassigned”)	Reference designator for the board assembly instance.
Old_Refdes	String	Optional (Refdes has not been changed by the sending application.)	Previous reference designator for the board assembly instance.
Lock	Ext_Enum	EList (“MCAD”, “ECAD”, “None”) Default (“None”)	Instance lock status.
XY_Loc[2]	Real	- - -	XY location of the board assembly instance.
Rotation	Real	Range (0.0 <= Rotation < 360.0)	Rotation (in degrees) of the board assembly instance, about the Z-axis, at the instance location. Rotation is counter-clockwise with respect to the instance location.
Mirror	Boolean	Default (“False”)	Mirror flag for the board assembly instance. A value of true means that the board assembly instance is rotated 180 degrees about the Y-axis, at the instance location.
Footprint	Ref_ID	Optional (Footprint is not required for the translation.) Informal (The footprint must be contained in the panel part.)	Footprint that is associated with the board assembly instance.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Panel_Assembly)			

Usage Notes:

- The board assembly instance is located in the coordinate system of the panel assembly that contains it.
- If the Old_Refdes attribute is present, it means that the sending application changed the Refdes for this instance. The receiving application must use the Old_Refdes value to find the corresponding assembly instance, and modify its reference designator accordingly.

6.6 Board_Part

The Board_Part entity is used to represent the "bare" printed circuit board (without any components assembled to it). It includes the set of product and design features that are directly associated with the bare board.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Part_Name	Ent_Name	- - -	Name of the board part.
Units	Enum	EList ("Inch", "MM", "Global") Default ("Global")	Length units for the entities that make up the board part. Global means that the units are equal to the default units specified in the IDF Header.
Type	Ext_Enum	EList ("Unspecified")	Board part type.
Shape[1..?]	Entity	RList (Extrusion) Informal (The bottom surface of the extrusion in which the XY origin is located, must be at Z = 0.0.)	Extrusion(s) that make up the board part's shape.
Phy_Layers[0..?]	Entity	RList (Physical_Layer) Optional (Physical layers are not required for the translation.)	List of physical layer descriptions for the board part.
Features[0..?]	Entity	RList (Annotation, Cavity, Cutout, Figure, Filled_Area, Graphic, Hole, Keepin, Keepout, Pad, Trace) Optional (Board features are not required for the translation.)	List of features contained in the board part.
Footprints[0..?]	Entity	RList (Footprint) Optional (Footprints are not required for the translation.)	List of footprints associated with the board part.
Material	Ref_Name	RList (Material) Optional (Material characteristics are not required for the translation.)	Board part material characteristics.
Properties[0..?]	NValue	NList (Avg_M2V_Ratio, U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Informal (All cutouts, cavities, and holes must be located within the outer boundary of the board part's extrusion(s).) Where_Used (*Independent*, Board_Part_Instance (*Ref. by Name*))			

Usage Notes:

- The board part has its own coordinate system in which its features are defined.

6.7 Board_Part_Instance

The Board_Part_Instance entity is used to represent the occurrence of a board part within a board assembly.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Part_Name	Ref_Name	RList (Board_Part)	Name of the board part that is instanced.
Part_Number	String	Optional (Part number is not known.)	Part number for the board part instance.
Refdes	String	Default (“Unassigned”)	Reference designator for the board part instance.
Old_Refdes	String	Optional (Refdes has not been changed by the sending application.)	Previous reference designator for the board part instance.
XY_Loc[2]	Real	- - -	XY location of the board part instance.
Rotation	Real	Range (0.0 <= Rotation < 360.0)	Rotation (in degrees) of the board part instance, about the Z-axis, at the instance location. Rotation is counter-clockwise with respect to the instance location.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Informal (The board part instance must be located at XY_Loc = 0.0, 0.0 and Rotation = 0.0 in the board assembly.) Where_Used (Board_Assembly)			

Usage Notes:

- The board part instance is located in the coordinate system of the board assembly that contains it.
- If the Old_Refdes attribute is present, it means that the sending application changed the Refdes for this instance. The receiving application must use the Old_Refdes value to find the corresponding part instance, and modify its reference designator accordingly.

6.8 Cavity

The Cavity entity is used to represent a partial void (absence of material) in an extrusion. The cavity is defined by an outline, a depth, and the surface (top or bottom) from which it extends into the extrusion.

Attribute Name	Type	Constraints	Descripton
Entity_ID	Ent_ID	- - -	IDF entity ID.
Feature_ID	String	Default (“Unassigned”)	Unique feature identifier.
Lock	Ext_Enum	EList (“MCAD”, “ECAD”, “None”) Default (“None”)	Feature lock status.
Surface	Enum	EList (“Top”, “Bottom”)	Surface from which the cavity extends into the extrusion.
Depth	Real	Range (Depth > 0.0)	Depth of the cavity, from the surface of the extrusion.
Outline	Entity	RList (Circle, Polycurve_Area, Polygon)	Outer boundary of the cavity.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Informal (The outline of a cavity must exist completely within the outline of an extrusion within the part where it is used.) Informal (The outline of a cavity may not intersect the outline of the extrusion, or another cavity or cutout within the part. However, it may share one or more edges (coincident curves) with the extrusion or other cavities or cutouts.) Where_Used (Board_Part, Electrical_Part, Figure, Footprint, Mechanical_Part, Panel_Part)			

Usage Notes:

- The cavity is defined in the coordinate system of the entity that contains it.
- The association between a cavity and an extrusion is by location only. The cavity affects the extrusion whose outline encompasses the outline of the cavity.
- A cavity may be contained within another cavity, producing a “countersunk” cavity. The depth of a cavity is always from the surface of the extrusion, so the depth of the inner cavity must be greater than the depth of the outer cavity.

- Figure 6.2 illustrates a set of valid and invalid cavities

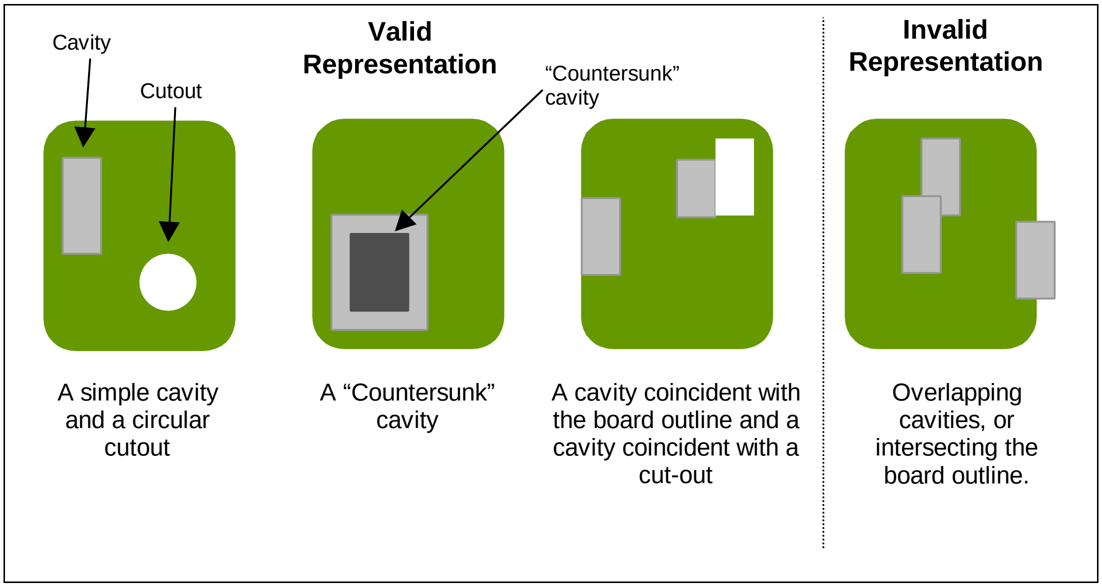

Valid Representation
Invalid Representation
Cavity
Cutout
"Countersunk" cavity
A simple cavity and a circular cutout
A "Countersunk" cavity
A cavity coincident with the board outline and a cavity coincident with a cut-out
Overlapping cavities, or intersecting the board outline.

Figure 6.2 Valid and Invalid Cavity Representation

6.9 Circle

The Circle entity is used to represent an XY planar area, bounded by a circle.

Attribute Name	Type	Constraints	Descripton
Entity_ID	Ent_ID	- - -	IDF entity ID.
Line_Font	Enum	EList (“Solid”, “Center”, “Phantom”, “Dashed_Long”, “Dashed_Short”, “Dotted”) Default (“Solid”)	Line font of the boundary curve.
Line_Color[3]	Real	Range (0.0 <= Line_Color[0..2] <= 100.0) Optional (Line color is not required for the translation.)	Color of the boundary curve, expressed as percentages of intensity of red, green, and blue (RGB), respectively.
Fill_Color[3]	Real	Range (0.0 <= Fill_Color[0..2] <= 100.0) Optional (Fill color is not required for the translation.)	Fill color of the bounded area, expressed as percentages of intensity of red, green, and blue (RGB), respectively.
XY_Loc[2]	Real	- - -	XY location of the center of the circle.
Radius	Real	Range (Radius >= 0.0)	Radius of the circle.
Properties[0..?]	NValue	NList (*User-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Annotation, Cavity, Cutout, Extrusion, Filled_Area, Graphic, Hole (*Ref. by ID*), Keepin, Keepout, Pad (*Ref. by ID*))			

Usage Notes:

- The circle is defined in the coordinate system of the part, figure, footprint, pad, or hole where it is used.

6.10 Circular_Arc

The Circular_Arc entity is used to represent an XY planar circular arc (open or closed).

Attribute Name	Type	Constraints	Descripton
Entity_ID	Ent_ID	- - -	IDF entity ID.
Line_Font	Enum	EList (“Solid”, “Center”, “Phantom”, “Dashed_Long”, “Dashed_Short”, “Dotted”) Default (“Solid”)	Line font of the curve.
Line_Color[3]	Real	Range (0.0 <= Line_Color[0..2] <= 100.0) Optional (Line color is not required for the translation.)	Color of the curve, expressed as percentages of intensity of red, green, and blue (RGB), respectively.
Width	Real	Range (Width > 0.0) Optional (Line width is not required for the translation.)	Line width for the curve.
Start_Style	Enum	EList (“Round”, “Square”, “None”) Default (“None”)	End condition for the first point of the curve.
End_Style	Enum	EList (“Round”, “Square”, “None”) Default (“None”)	End condition for the second point of the curve.
XY_Start[2]	Real	- - -	XY location for the start point.
XY_Mid[2]	Real	- - -	XY location for the mid point on the circular arc.
XY_End[2]	Real	- - -	XY location for the end point.
Properties[0..?]	NValue	NList (*User-defined*) Optional ()	Associated property values.
Entity Constraints: Informal (If XY_Start and XY_End are equivalent, the circular arc is closed.) Informal (The arc length of the circular arc must be greater than the file header value Min_Res.) Where_Used (Annotation, Graphic, Leader, Pad (*Ref. by ID*))			

Usage Notes:

- The circular arc is defined in the coordinate system of the part, figure, footprint, pad, or hole where it is used.
- A closed circular arc represents a circular ring. It does not represent an area.

- Figure 6.3 illustrates the representation of a circular arc.

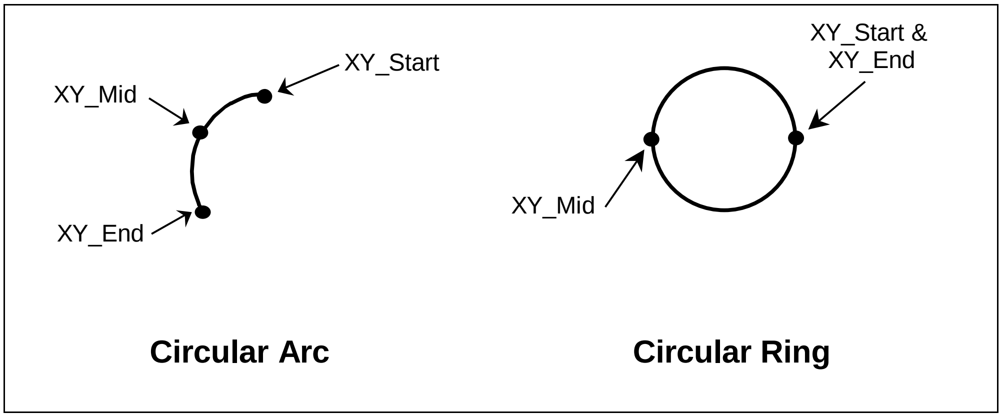


Figure 6.3 Circular Arc Representation

6.11 Cutout

The Cutout entity is used to represent a void (absence of material) in an extrusion.

Attribute Name	Type	Constraints	Descripton
Entity_ID	Ent_ID	- - -	IDF entity ID.
Feature_ID	String	Default (“Unassigned”)	Unique feature identifier.
Lock	Ext_Enum	EList (“MCAD”, “ECAD”, “None”) Default (“None”)	Feature lock status.
Outline	Entity	RList (Circle, Polycurve_Area, Polygon)	Outer boundary of the cutout.
Properties[0..?]	NValue	NList (*User-defined*) Optional ()	Associated property values.

Entity Constraints:

Informal (The outline of a cutout must exist completely within the outline of an extrusion within the part, where it is used.)

Informal (The outline of a cutout may not intersect the outline of an extrusion or another cutout or cavity, nor may it become tangent with the outline of an extrusion or cutout. However, it may share one or more edges (coincident curves) with a cavity.)

Where_Used (Board_Part, Electrical_Part, Figure, Footprint, Mechanical_Part, Panel_Part)

Usage Notes:

- A cutout is defined in the coordinate system of the entity that contains it.
- The association between a cutout and an extrusion is by location only. The cutout affects the extrusion whose outline encompasses the outline of the cutout.
- Figure 6.4 illustrates valid and invalid cutouts.

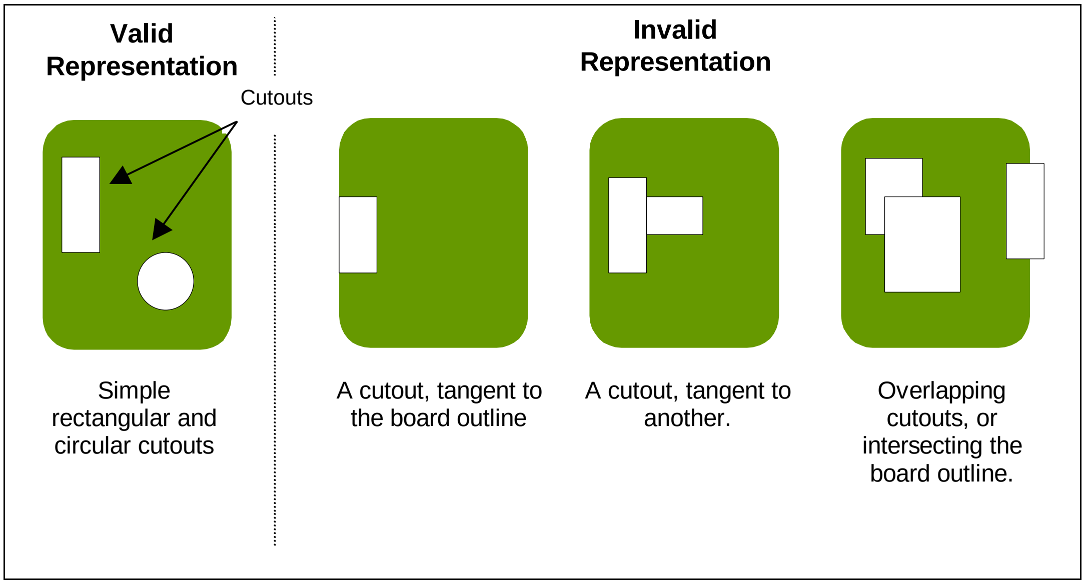

Valid Representation
Invalid Representation
Cutouts
Simple rectangular and circular cutouts
A cutout, tangent to the board outline
A cutout, tangent to another.
Overlapping cutouts, or intersecting the board outline.

Figure 6.4 Valid and Invalid Cutout Representation

6.12 Electrical_Part

The Electrical_Part entity is used to represent the physical characteristics of an electrical part. The difference between an electrical part and a mechanical part is that an electrical part may contain a set of pins, and is electrically connected to the board or panel part.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Part_Name	Ent_Name	- - -	Name of the electrical part.
Units	Enum	EList (“Inch”, “MM”, “Global”) Default (“Global”)	Length units for the entities that make up the electrical part. Global means that the units are equal to the default units specified in the IDF Header.
Type	Ext_Enum	EList (“Printed”, “Surface”, “Thru”, “Unspecified”)	Electrical part type.
Mnt_Shape[1..?]	Entity	RList (Cavity, Cutout, Extrusion) Informal (At least one Extrusion entity is required.)	Mounting side shape of the electrical part.
Opp_Shape[0..?]	Entity	RList (Cavity, Cutout, Extrusion) Informal (If there is an opposite side shape, at least one Extrusion entity is required.) Optional (There is no opposite side shape.)	Opposite side shape of the electrical part.
Pins[0..?]	Entity	RList (Pin) Optional (Pins are not required for the translation.)	List of pins contained in the electrical part.
Annos[0..?]	Entity	RList (Annotation) Optional (Annotations are either not in the part, or are not required for the translation.)	List of annotations contained in the electrical part.
Pin_Material	Ref_Name	RList (Material) Optional (Pin material characteristics are not required for the translation.)	Pin material characteristics.
Comp_Material	Ref_Name	RList (Material) Optional (Component material characteristics are not required for the translation.)	Electrical part material characteristics.
TH_Model	Ref_Name	RList (Thermal_Model) Optional (Thermal model is not required for the translation.)	Thermal model for the electrical part.

Attribute Name	Type	Constraints	Description
Properties[0..?]	NValue	NList (EL_Capacitance, EL_Power_Nominal, EL_Power_Min_Max, EL_Resistance, EL_Tolerance, Pin_Area, Pin_Count, Pin_Layout, Pin_Length, TH_Air_Gap TH_Capacitance, TH_Heat_Dis_Nominal, TH_Heat_Dis_Min_Max, TH_C2B_Resistance, TH_J2B_Resistance, TH_J2C_Resistance, *User-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (*Independent*, Electrical_Part_Instance (*Ref. by Name*))			

Usage Notes:

- The electrical part has its own coordinate system in which its features are defined.
- An electrical part must have a mounting side shape. In addition, it may optionally have an opposite side shape (to represent a fastener or retainer, for example).
- The shape of a printed electrical part does not have any height (it is simply a planar outline). It is represented by an extrusion with height equal to 0.0.
- The extrusions that make up either the mounting side shape or the opposite side shape may or may not be physically joined (share a common boundary curve).
- Figure 6.5 illustrates typical component shapes.

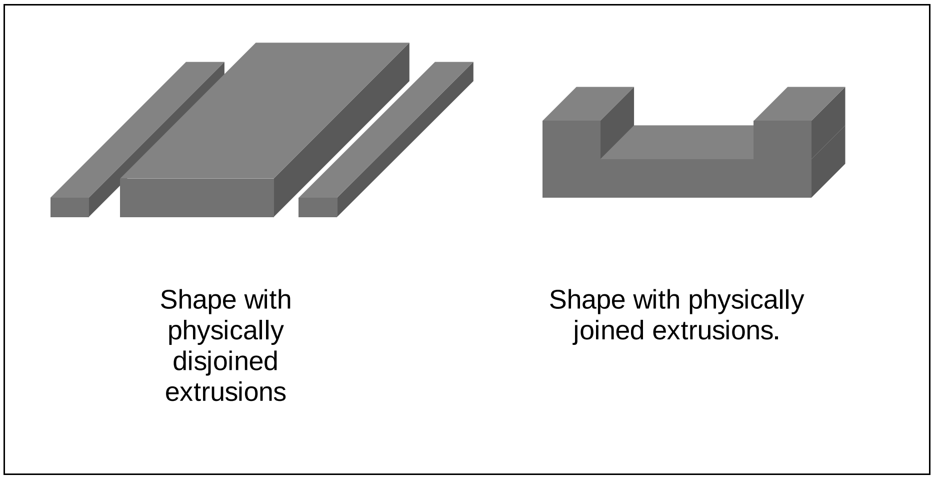

Shape with physically disjoined extrusions
Shape with physically joined extrusions.

Figure 6.5 Typical Component Shapes

- Board or panel features that are associated with the electrical part (pads, holes, etc.) are defined within a footprint, which is associated with each electrical part instance. Consequently, the board or panel features are not defined in the electrical part itself.

- There is no physical relationship between the pins of an electrical part and the pads and holes in the corresponding footprint. If the pads or holes are moved in the footprint (the result of a moved pin on an electrical part instance, for example) the pin location on the electrical part is not changed.
- The electrical part types are defined as follows:

Type	Description
Printed	Component printed on a conductive layer.
Surface	Component mounted on surface pads.
Thru	Component mounted in thru holes.
Unspecified	Component type is unkown or not required.

- An electrical part that uses blind pins is considered to be of type “surface”.

6.13 Electrical_Part_Instance

The Electrical_Part_Instance entity is used to represent the occurrence of an electrical part, and optionally an associated footprint, in a board or panel assembly. Electrical part instances are uniquely identified in the design by their reference designators.

Attribute Name	Type	Constraints	Descripton
Entity_ID	Ent_ID	- - -	IDF entity ID.
Part_Name	Ref_Name	RList (Electrial_Part)	Name of the electrical part that is instanced.
Part_Number	String	Optional (Part number is not known or there is no part number.)	Part number for the electrical part instance.
In_BOM	Boolean	Default (“True”)	Bill of material flag. True means that the part instance is included in the bill of material.
Refdes	String	Default (“Unassigned”)	Reference designator for the instance.
Old_Refdes	String	Optional (Refdes has not been changed by the sending application.)	Previous reference designator for the instance.
Lock	Ext_Enum	EList (“MCAD”, “ECAD”, “None”) Default (“None”)	Instance lock status.
XY_Loc[2]	Real	- - -	XY location of the instance.
Side	Enum	EList (“Top”, “Bottom”)	Side of the board or panel on which the instance is mounted. Bottom side instances are rotated 180 degrees about the Y-axis, at the instance location.
Embedded	Boolean	Default (“False”) Informal (If True, the layer attribute must be used to specify the internal layer on which the instance is located.)	Specifies whether or not the instance is embedded on an internal layer or not. A value of true means that the instance is embedded.
Rotation	Real	Range (0.0 <= Rotation < 360.0)	Rotation (in degrees) of the electrical part instance, about the Z-axis, at the instance location. Rotation is counter-clockwise with respect to the instance location.
Layer	Ref_Name	RList (Physical_Layer) Optional (Instance is not embedded on an internal layer.)	Conductive layer on which an embedded printed part instance is located.

Attribute Name	Type	Constraints	Descripton
Mnt_Offset[1..2]	Real	Range (Mnt_Offset[0..1] >= 0.0) Default (0.0, 0.0)	Mounting offset from the board or panel (from the nominal mounting height). The first value is for the mounting side shape. The second value is for the opposite side shape, if it exists.
Variant_Use	Enum	EList ("All", "Inclusion", "Exclusion") Optional (There are no design variants for the board assembly.)	Use of the instance with respect to design variants. All Instance is in all design variants. Inclusion Instance is only in the design variants specified in the variants list. Exclusion Instance is in all design variants, except those specified in the vairants list.
Variants[0..?]	String	Optional (There are no design variants, or the variant use value is equal to All.)	List of design variant names that the instance is included in or excluded from, depending on the variant use value.
Footprint	Ref_ID	Optional (Footprint is not required for the translation.) Informal (The footprint must be contained in the board part.)	Footprint that is associated with the instance.
Properties[0..?]	NValue	NList (EL_Capacitance, EL_Power_Nominal, EL_Power_Min_Max, EL_Resistance, EL_Tolerance, Pin_Area, Pin_Count, Pin_Layout, Pin_Length, TH_Air_Gap TH_Capacitance, TH_Heat_Dis_Nominal, TH_Heat_Dis_Min_Max, TH_C2B_Resistance, TH_J2B_Resistance, TH_J2C_Resistance, U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Board_Assembly, Panel_Assembly, Sublayout (*Ref. by ID*))			

Usage Notes:

- The electrical part instance is located in the coordinate system of the assembly that contains it.

- The placement of an electrical part instance is determined as follows:
  - The instance is placed at the XY location, with respect to the assembly's origin and in the assembly's units.
  - The instance origin is aligned (along the Z-axis) with the surface of the board or panel part. The surface may be the top, bottom, or exposed surface of a cavity that extends into the board or panel part from the top or bottom.
  - If the board surface is the bottom or bottom-facing cavity surface, the instance is rotated 180 degrees about the Y-axis, at the instance location.
  - The instance is then rotated about the Z-axis, at the instance location.
  - If a mounting offset is specified, the instance is offset from the surface of the board, the specified amount (in the assembly's units).
- Rotation is always counterclockwise, with respect to the instance location. Figure 6.6 illustrates component instance rotation.

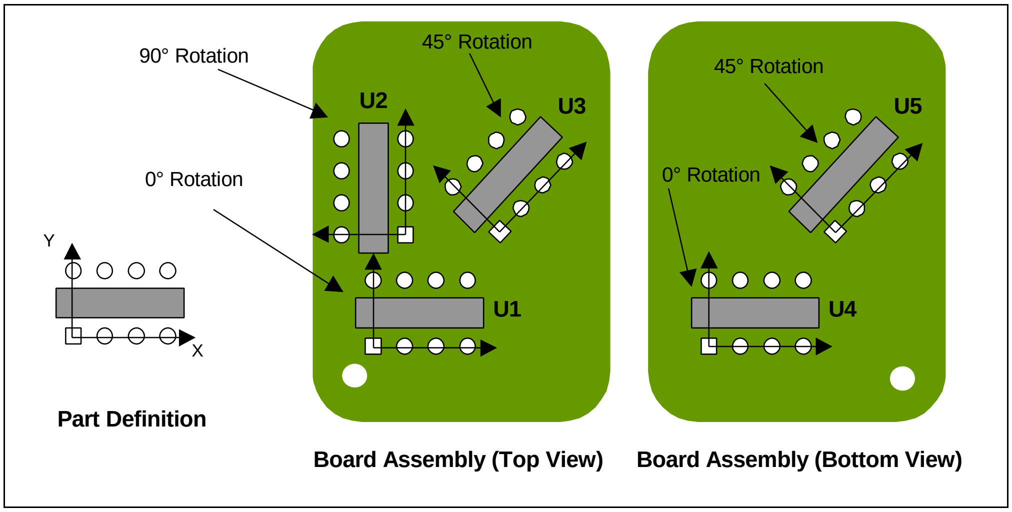


Figure 6.6 Component Instance Rotation

- Figure 6.7 illustrates component instance mounting.

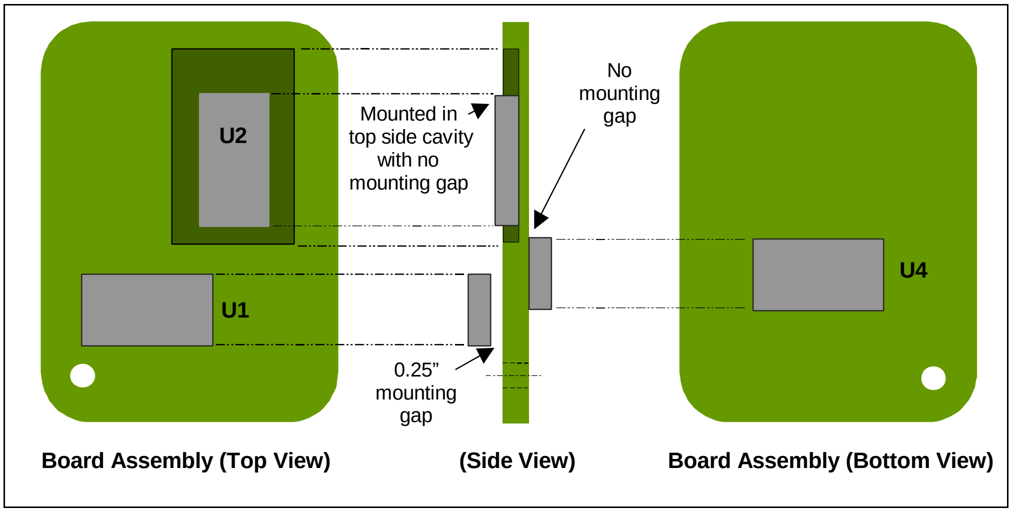

U2
Mounted in top side cavity with no mounting gap
No mounting gap
U4
U1
0.25" mounting gap
Board Assembly (Top View)
(Side View)
Board Assembly (Bottom View)

Figure 6.7 Component Instance Mounting

- Electrical part instance reference designators are required in the ECAD application to represent the electrical connectivity of the circuit. If the Refdes attribute value is equal to "Unassigned", it must be assigned before the electrical part instance is processed by the ECAD application.
- An electrical part instance may be placed on an internal conductor layer (embedded). In this case, the Side attribute is used to specify whether or not the electrical part instance is flipped about the Y-axis or not.
- If the electric part instance is embedded, and there are no physical layers defined, the Inner layer alias must be used to denote that the instance is located on "some" internal layer.
- Each alternate or substitute for a component instance must reference a common footprint.
- If the Old_Refdes attribute is present, it means that the sending application changed the Refdes for this instance. The receiving application must use the Old_Refdes value to find the corresponding part instance, and modify its reference designator accordingly.
- If the same property is on both an electrical part and an instance of that electrical part, the property on the electrical part instance overrides the one on the electrical part.

6.14 Extrusion

The Extrusion entity is used to represent the base shape of a part. The shape is created by extruding an outline, a specified distance (from the Bot_Height to the Top_Height), along the Z-axis of the part that contains the extrusion.

Attribute Name	Type	Constraints	Descripton
Entity_ID	Ent_ID	- - -	IDF entity ID.
Feature_ID	String	Default (“Unassigned”)	Unique feature identifier.
Lock	Ext_Enum	EList (“MCAD”, “ECAD”, “None”) Default (“None”)	Feature lock status.
Top_Height	Real	- - -	Height of the top surface of the extrusion.
Bot_Height	Real	Default (0.0)	Height of the bottom surface of the extrusion.
Outline	Entity	RList (Circle, Polycurve_Area, Polygon)	Boundary outline for the extrusion.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Informal (The extrusion for a board, panel, and mounting side component shape must have a Top_Height that is greater than or equal to the Bot_Height.) Informal (The extrusion for an opposite side component shape must have a Top_Height that is less than or equal to the Bot_Height.) Where_Used (Board_Part, Electrical_Part, Mechanical_Part, Panel_Part)			

Usage Notes:

- An extrusion is defined in the coordinate system of the entity that contains it.
- The Top_Height and Bot_Height are expressed as values along the Z-axis.
- For component part shapes, the Z-origin is equal to the mounting surface (or the opposite mounting surface for opposite side mounting shapes).
- An extrusion may have a height equal to 0.0 (to support printed components or because the height is unknown) and should be represented in MCAD as an outline, a face, or a cosmetic feature.
- For component parts, the bottom of an extrusion is the surface that faces the board or panel when the component is mounted. The top of an extrusion is the surface that faces away from the board. This is true of both mounting side and opposite side shape extrusions.
- For component parts, a negative bottom height for a mounting side shape implies that the component shape extends into or through the board or panel (presumably into a cavity or through a cutout), at its nominal mounting height. Similarly, a positive bottom height for an opposite side shape extends into or through the board or panel.

- Figure 6.8 illustrates the use of extrusions for mounting side and opposite side component shapes.

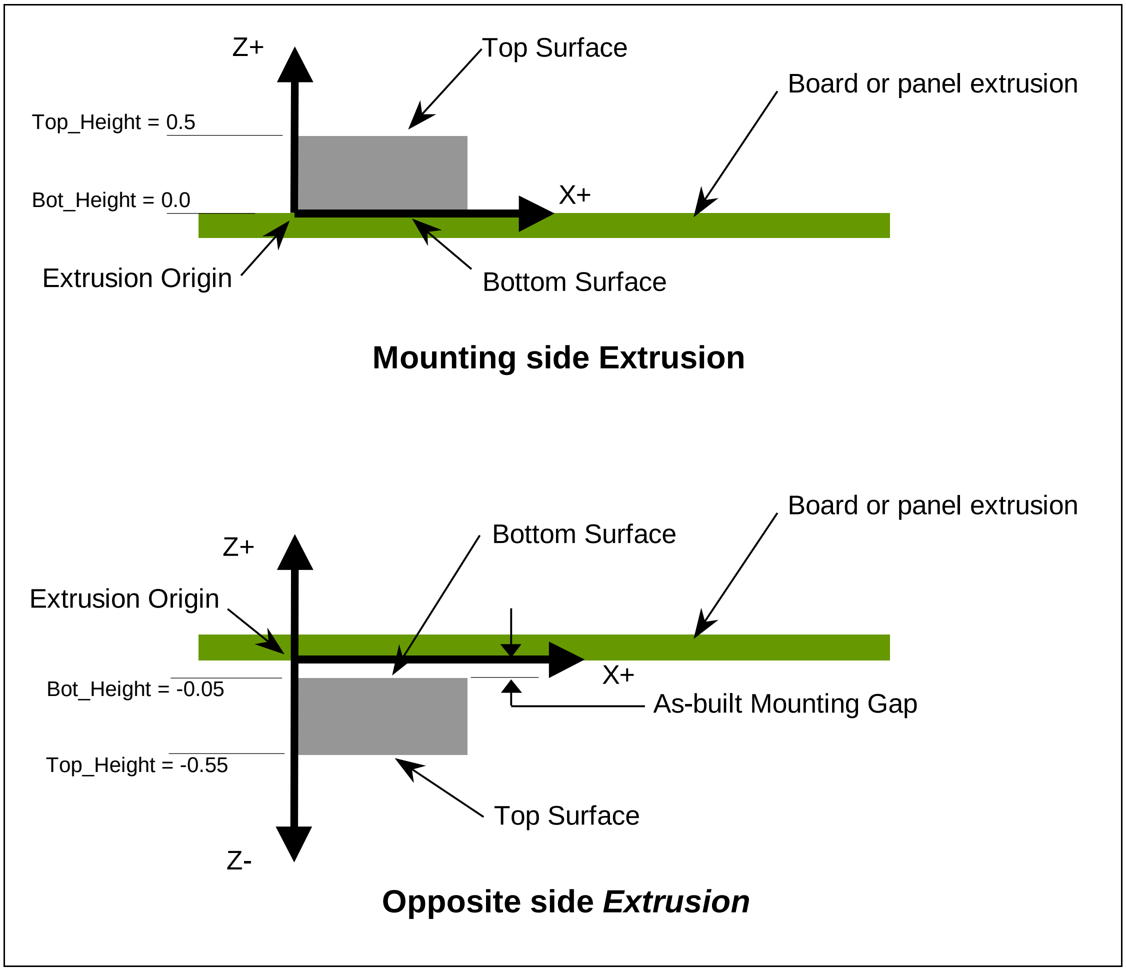


Figure 6.8 Mounting and Opposite Side Component Shape Extrusions

6.15 Figure

A Figure entity is used to represent a set of board or panel features that can be instanced at one or more locations.

Attribute Name	Type	Constraints	Descripton
Entity_ID	Ent_ID	- - -	IDF entity ID.
Figure_ID	String	Default (“Unassigned”)	Unique Figure instance identifier.
Lock	Ext_Enum	EList (“MCAD”, “ECAD”, “None”) Default (“None”)	Feature lock status.
Figure_Name	Ref_Name	- - -	Name of the figure.
Units	Enum	EList (“Inch”, “MM”, “Global”) Default (“Global”)	Length units for the entities that make up the figure. Global means that the units are equal to the default units specified in the IDF Header.
XY_Loc[2]	Real	- - -	XY location of the figure.
Side	Enum	EList (“Top”, “Bottom”)	Side of the board or panel on which the figure is located. Bottom side figures are rotated 180 degrees about the Y-axis, at the instance location.
Rotation	Real	Range (0.0 <= Rotation < 360.0)	Rotation (in degrees) of the figure, about the Z-axis, at the instance location. Rotation is counter-clockwise with respect to the instance location.
Features[0..?]	Entity	RList (Annotation, Cavity, Cutout, Filled_Area, Graphic, Hole, Keepout, Pad, Trace) Optional (Receiving application must substitute a corresponding figure at the specified location.)	List of features that are contained in the figure.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Board_Part, Panel_Part)			

Usage Notes:

- The figure is located in the coordinate system of the part that contains it.
- The figure has its own coordinate system in which its features are defined. The features must be transformed according to the location, side, and rotation of the figure, to be properly located.
- Layer based features contained within the figure are resolved to the appropriate layer.
- A typical use for a figure is to associate a mounting or tooling hole with pads and keepouts.

- Another typical use for a figure is to indicate the name of a predefined hole symbol or graphic symbol (such as a logo) to be used in the receiving application.
- Figure 6.9 illustrates the representation of a figure.

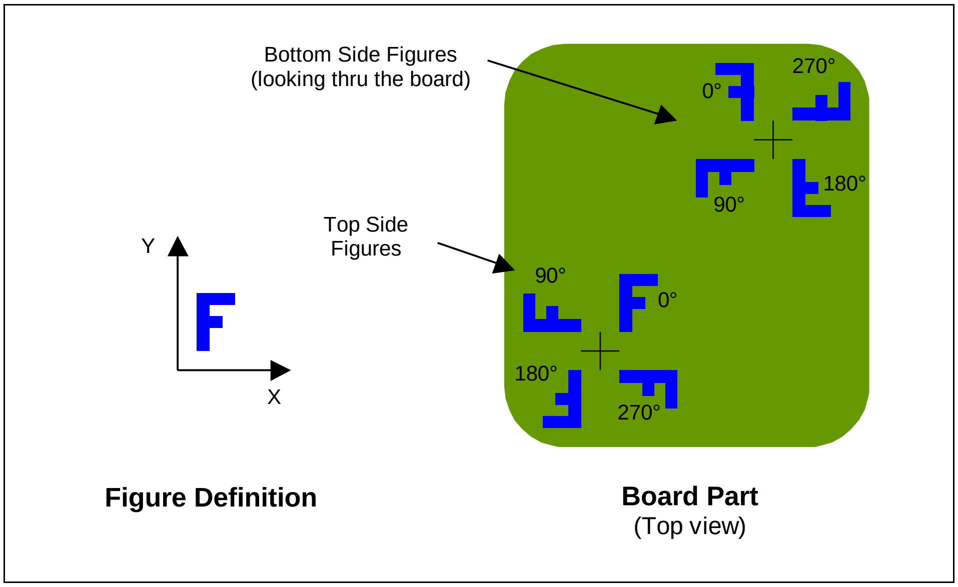


Figure 6.9 Figure Representation

6.16 Filled_Area

The Filled_Area entity is used to represent a conductive area, where the area is defined by an outline with one or more voids. In addition, a filled area may optionally have an associated net name.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Feature_ID	String	Default (“Unassigned”)	Unique feature identifier.
Lock	Ext_Enum	EList (“MCAD”, “ECAD”, “None”) Default (“None”)	Feature lock status.
Layer	Ref_Name	RList (Physical_Layer) Informal (Layer attribute must specify a physical layer name or a predefined layer alias.)	Board or panel layer on which the filled area is located.
Outline	Entity	RList (Circle, Polycurve_Area, Polygon)	Outer boundary of the filled area.
Voids[0..?]	Entity	RList (Circle, Polycurve_Area, Polygon) Informal (The boundary of a void may not intersect or become coincident with the boundary of another void.)	List of voids for the filled area.
Net_Name	String	Optional (Net name is not required for the translation.)	Net name that is associated with the filled area.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Informal (The boundary of each void must exist completely within the outline of the filled area.) Where_Used (Board_Part, Figure, Footprint, Panel_Part)			

Usage Notes:

- A filled area is defined in the coordinate system of the entity that contains it.
- Figure 6.10 illustrates a valid and invalid filled area.

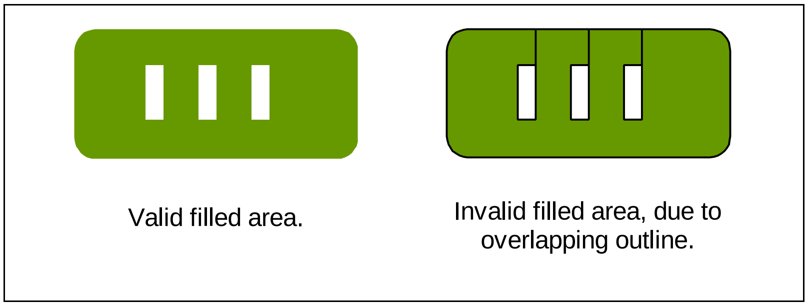


Figure 6.10 Valid and Invalid Filled Areas

6.17 Footprint

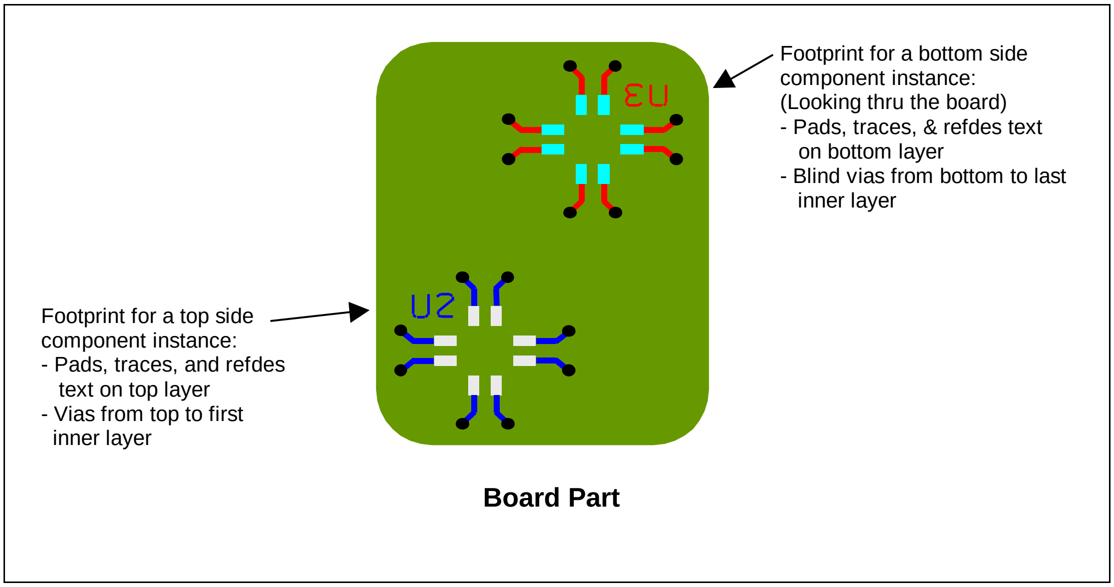
The Footprint entity is used to represent a set of features that are contributed to a board or panel part, via a component instance, sublayout, board assembly instance, or panel assembly instance. As the name implies, a footprint "imprints" features into the board or panel, that are associated with the part or sublayout that is being instanced.

Attribute Name	Type	Constraints	Descripton
Entity_ID	Ent_ID	- - -	IDF entity ID.
Assoc_Instance	String	Default ("Unassigned")	Associated instance reference designator or sublayout ID that the footprint is associated with.
Units	Enum	EList ("Inch", "MM", "Global") Default ("Global")	Length units for the entities that make up the footprint. Global means that the units are equal to the default units specified in the IDF Header.
Lock	Ext_Enum	EList ("MCAD", "ECAD", "None") Default ("None")	Footprint lock status.
XY_Loc[2]	Real	- - -	XY location of the footprint.
Side	Enum	EList ("Top", "Bottom")	Side of the board or panel in which the footrint is instanced. Bottom side footprints are rotated 180 degrees about the Y-axis, at the instance location.
Rotation	Real	Range (0.0 <= Rotation < 360.0)	Rotation (in degrees) of the footprint, about the Z-axis, at the instance location. Rotation is counter-clockwise with respect to the instance location.
Features[1..?]	Entity	RList (Annotation, Cavity, Cutout, Filled_Area, Graphic, Hole, Keepout, Pad, Trace)	List of features that are contained in the footprint.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Informal (The XY_Loc, Side, and Rotation must be the same as that defined for the corresponding part or assembly instance.) Where_Used (Board_Assembly_Instance (*Ref. by ID*), Board_Part, Electrical_Part_Instance (*Ref. by ID*), Mechanical_Part_Instance (*Ref. by ID*), Panel_Assembly_Instance (*Ref. by ID*), Panel_Part, Sublayout (*Ref. by ID*))			

Usage Notes:

- The footprint is located in the coordinate system of the part that contains it.
- The footprint has its own coordinate system in which its features are defined. The features must be transformed according to the location, side, and rotation of the footprint, to be properly located.

- Layer based features contained within the footprint are resolved to the appropriate layer.
- Footprints are only allowed in board or panel parts. However, they can be contributed by, and therefore associated with, component instances, board assembly instances, panel assembly instances, or sublayouts.
- Figure 6.11 illustrates the representation of a footprint.



Footprint for a bottom side component instance:
(Looking thru the board)
- Pads, traces, & refdes text on bottom layer
- Blind vias from bottom to last inner layer

Footprint for a top side component instance:
- Pads, traces, and refdes text on top layer
- Vias from top to first inner layer

Figure 6.11 Footprint Representation

6.18 Graphic

The Graphic entity is used to represent miscellaneous physical features of a board or panel part. It is comprised of a set of text and/or geometry entities.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Feature_ID	String	Default (“Unassigned”)	Unique feature identifier.
Lock	Ext_Enum	EList (“MCAD”, “ECAD”, “None”) Default (“None”)	Feature lock status.
Type	Ext_Enum	EList (“Fiducial”, “Glue_Mask”, “Milling_Path”, “Paste_Mask”, “Silkscreen”, “Solder_Mask”, “Unspecified”)	Graphic type. See detailed descriptions below.
Layer	Ref_Name	RList (Physical_Layer) Informal (Layer attribute must specify a physical layer name or a predefined layer alias.)	Board or panel layer on which the graphic is located.
Entities[1..?]	Entity	RList (Circle, Circular_Arc, Polycurve, Polycurve_Area, Polygon, Polyline, Text)	List of entities that represent the graphic.
Properties[0..?]	NValue	NList (*User-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Board_Part, Figure, Footprint, Panel_Part)			

Usage Notes:

- The entities that represent the graphic are defined in the coordinate system of the entity that contains the graphic.
- The graphic types are defined as follows:

Type	Description
Fiducial	A graphic used to align of component insertion equipment.
Glue_Mask	A graphic that defines a glue mask shape.
Milling_Path	A graphic that represents the centerline tool path for milling.
Paste_Mask	A graphic that defines a solder paste mask shape.
Silkscreen	A graphic that defines a silkscreen shape.
Solder_Mask	A graphic that defines a solder mask shape.
Unspecified	Graphic type is unkown or not required.

6.19 Hole

The Hole entity is used to represent a passage that is either partially or completely through a board or panel part. Holes serve specific purposes such as for fastening components to the board (mounting), aligning the board in a manufacturing fixture (tooling), inserting component pins into the board (pin), providing electrical connectivity among the board's conductive layers (via), or providing heat dissipation through the board (thermal_via).

Attribute Name	Type	Constraints	Descripton
Entity_ID	Ent_ID	- - -	IDF entity ID.
Feature_ID	String	Default ("Unassigned")	Unique feature identifier.
Lock	Ext_Enum	EList ("MCAD", "ECAD", "None") Default ("None")	Feature lock status.
Type	Ext_Enum	EList ("Blind_Pin", "Blind_Via", "Buried_Via", "Mounting", "Mount_Tool", "Thermal_Via", "Tooling", "Thru_Pin", "Thru_Via", "Unspecified")	Hole type.
Assoc_Pin_ID	String	Optional (The hole either does not have an associated pin, or it is not required for the translation.)	Associated pin ID.
Side	Enum	EList ("Top", "Bottom", "Both") Default ("Both")	Side of the board or panel part where the hole is located. Both means that the hole goes thru both sides of the board or panel. Top or bottom only applies to holes that go partially thru the board or panel.
Shape_Type	Ext_Enum	EList ("Key", "Round", "Square", "Slot", "Unspecified")	Geometric shape of the hole.
Outline	Ref_ID	RList (Circle, Polycurve_Area, Polygon) Informal (Geometry that represents the outline of the hole, must correspond to the Shape_Type value.)	Outline of the hole.
XY_Loc[2]	Real	- - -	XY location of the hole.
Rotation	Real	Range (0.0 <= Rotation < 360.0)	Rotation (in degrees) of the hole's outline, about the Z-axis, at the hole's location. Rotation is counter-clockwise with respect to the instance location.
Span[2]	Ref_Name	RList (Physical_Layer) Optional (Hole represents a thru hole.)	Start and stop layer for the hole.

Attribute Name	Type	Constraints	Descripton
Plated	Boolean	Default ("False")	Plating status for the hole. A value of true means that the hole is plated.
Net_Name	String	Optional (Hole is either not electrically connected, or the net name is not required for the translation.)	Net name that is associated with the hole.
Properties[0..?]	NValue	NList (*User-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Board_Part, Figure, Footprint, Panel_Part)			

Usage Notes:

- The hole is located in the coordinate system of the part that contains it.
- The outline referenced by the hole, is defined in the coordinate system of the hole. Consequently, the outline must be transformed according to the location, side, and rotation of the hole, to be properly located.
- The outline of a hole is a referenced entity. Therefore, it can be re-used by other holes. Typically, there are many holes in a board or panel that can use the same outline geometry.
- The outline of a round hole must be a circle. The outline of a square hole must be a polygon. The outline of a slot hole must be a polycurve area.
- Figure 6.12 illustrates the representation of a hole.

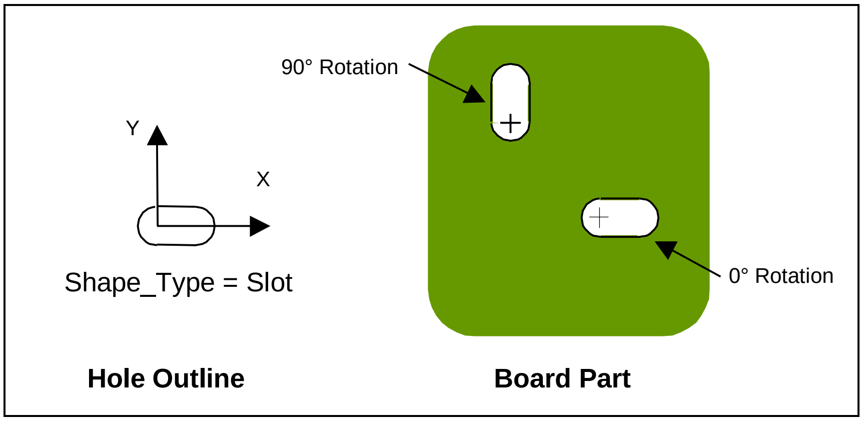


Figure 6.12 Hole Representation

- The hole types are defined as follows:

Type	Description
Blind_Pin	Component pin blind (partial) hole.
Blind_Via	Via blind (partial) hole.
Buried_Via	Via buried (from one internal layer to another internal layer) hole.
Mounting	Board or panel component hole.
Mount_Tool	Combination mounting and tooling hole.
Thermal_Via	Thermal via hole.
Thru_Pin	Component pin thru hole.
Thru_Via	Via thru hole.
Tooling	Manufacturing tooling hole.
Unspecified	Hole type is unkown or not required.

6.20 Keepin

The Keepin entity is used to represent an area on the board or panel in which component instances or board features must be located.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Feature_ID	String	Default (“Unassigned”)	Unique feature identifier.
Lock	Ext_Enum	EList (“MCAD”, “ECAD”, “None”) Default (“None”)	Feature lock status.
Type	Ext_Enum	EList (“Comp_All”, “Comp_Group”, “Comp_Rotation”, “Route_All”)	Keepin type.
Layers[1..?]	Ref_Name	RList (Physical_Layer) Informal (Layers attribute must specify a physical layer name(s) or a predefined layer alias.)	Board or panel layer(s) that are affected by the keepin.
Outline	Entity	RList (Circle, Polycurve_Area, Polygon)	Outer boundary of the keepin area.
Voids[0..?]	Entity	RList (Circle, Polycurve_Area, Polygon) Informal (The boundary of a void may not intersect or become coincident with the boundary of another void.) Optional (There are no voids in the keepin.)	List of voids for the keepin area.
Properties[0..?]	NValue	NList (KI_Comp_Group, KI_Comp_Height, KI_Comp_Rot, U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Informal (The boundary of each void must exist completely within the outline of the keepin.) Where_Used (Board_Part, Panel_Part)			

Usage Notes:

- A keepin is defined in the coordinate system of the entity that contains it.

- There are four predefined keepin types, which are described in the following table:

Keepin Type	Description / Usage
Comp_All	An area within which all component instances must be placed. • The Comp_All keepin may affect any valid component placement layer(s). • The Comp_All keepin optionally uses the KI_Comp_Height property value to specify a maximum placement height value. All components with a mounted height less than this value must be placed within the area. If the attribute value is not present, all component instances must be placed within the area.
Comp_Group	An area within which a specific set of component instances must be placed. • The Comp_Group keepin may affect any valid component placement layer(s). • The Comp_Group keepin uses the KI_Comp_Group property value to specify a list of component reference designators of the component instances that are to be placed within the area.
Comp_Rotation	An area within which component instances must be placed according to a set of predefined orientations. • The Comp_Rotation keepin may affect any valid component placement layer(s). • The Comp_Rotation Keepin uses the KI_Comp_Rot property value to specify a set of acceptable rotation values for the component placement direction (0 and 180 degrees, for example).
Route_All	An area within which all routing must occur. • The Route_All keepin may affect any valid routing layer(s). • The Route_All keepin does not use any property values.

6.21 Keepout

The Keepout entity is used to represent an area on the board or panel in which component instances or board features may not be located.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Feature_ID	String	Default (“Unassigned”)	Unique feature identifier.
Lock	Ext_Enum	EList (“MCAD”, “ECAD”, “None”) Default (“None”)	Feature lock status.
Type	Ext_Enum	EList (“Comp_By_Height”, “Route_All”, “Trace_All”, “Via_All”, “Testpoint_All”, “Silkscreen_All”, “Hole_All”)	Keepout type.
Layers[1..?]	Ref_Name	RList (Physical_Layer) Informal (Layers attribute must specify a physical layer name(s) or a predefined layer alias.)	Board or panel layer(s) that are affected by the keepout.
Outline	Entity	RList (Circle, Polycurve_Area, Polygon)	Outer boundary of the keepout area.
Voids[0..?]	Entity	RList (Circle, Polycurve_Area, Polygon) Informal (The boundary of a void may not intersect or become coincident with the boundary of another void.) Optional (There are no voids in the keepout.)	List of voids for the keepout area.
Properties[0..?]	NValue	NList (KO_Comp_Height, U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Informal (The boundary of each void must exist completely within the outline of the keepout.) Where_Used (Board_Part, Figure, Footprint, Panel_Part)			

Usage Notes:

- A keepout is defined in the coordinate system of the entity that contains it.

- There are seven predefined keepout types, which are described in the following table:

Keepout Type	Description / Usage
Comp_By_Height	An area in which no component instance having a mounted height greater than that specified for the keepout, is allowed. The component mounted height is equal to the maximum height of the shape of the component part plus the mounting offset of the component instance. • The Comp_By_Height keepout may affect any valid component placement layer(s). • The Comp_By_Height keepout uses the KO_Comp_Height property value to specify a maximum placement height value. If the attribute value equals 0.0 or is not present, the restriction applies to all component instances.
Route_All	An area in which no routing (traces and vias) is allowed. • The Route_All keepout may affect any valid routing layer(s). • The Route_All keepout does not use any property values.
Trace_All	An area in which no traces are allowed. However, vias are allowed. • The Trace_All keepout may affect any valid routing layer(s). • The Trace_All keepout does not use any property values.
Via_All	An area in which no vias are allowed. However, traces are allowed. • The Via_All keepout may affect any valid routing layer(s). • The Via_All keepout does not use any property values.
Testpoint_All	An area in which no testpoints are allowed. • The Testpoint_All keepout may affect either the Top, Bottom, or Both layers. • The Testpoint_All keepout does not use any property values.
Silkscreen_All	An area in which no silkscreen text or graphics are allowed. • The Silkscreen_All keepout may affect either the Top, Bottom, or Both layers. • The Silkscreen_All keepout does not use any property values.
Hole_All	An area in which no Holes are allowed. • The Hole_All keepout affects all layers. • The Hole_All keepout does not use any property values.

6.22 Leader

The Leader entity is used to represent a curve with an arrowhead on either the start and/or end of the curve. A leader is used to visually associate an annotation to its subject.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Arrow_Start	Boolean	- - -	Start arrowhead flag. A value of true means that an arrowhead is at the start point of the curve.
Arrow_End	Boolean	- - -	End arrowhead flag. A value of true means that an arrowhead is at the end point of the curve.
Curve	Entity	RList (Circular_Arc, Polyline)	Leader curve.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Annotation)			

Usage Notes:

- Figure 6.13 illustrates several different types of leaders.

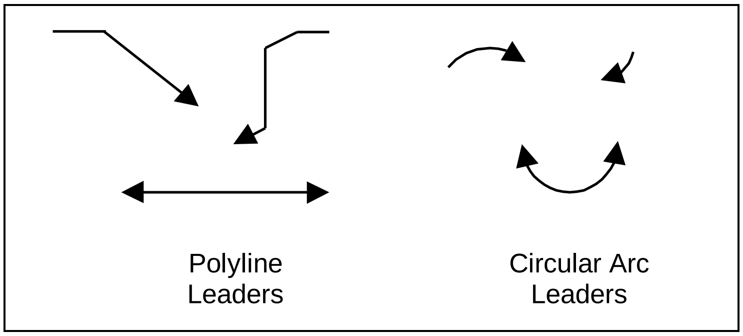


Figure 6.13 Polyline and Circular Arc Leaders

- The Leader entity does not specify an arrowhead style. The receiving application needs to apply the appropriate arrowhead style.

6.23 Material

The Material entity is used to represent the material characteristics of a part, which are required to conduct a thermal or structural analysis.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Material_Name	String	- - -	Name of the material.
ST_Mass_Density	Real	Range (ST_Mass_Density >= 0.0) Optional (Mass density is either unkown or not required for the translation.)	Structural mass density (in Kg/m^3).
ST_Mod_Elasticity	Real	Range (ST_Mod_Elasticity >= 0.0) Optional (Modulus of elasticity is either unkown or not required for the translation.)	Structural modulus of elasticity (expressed as a ratio of unit stress to unit strain within the proportional limit of the material in tension or compression).
TH_Iso_Cond	Real	Range (TH_Iso_Cond >= 0.0) Optional (Isotropic thermal conductivity is either unkown or not required for the translation.)	Isotropic thermal conductivity of the material (in watts/m °C).
TH_Ortho_Cond[3]	Real	Range (TH_Ortho_Cond[0..2] >= 0.0) Optional (orthotropic thermal conductivity is either unkown or not required for the translation.)	Orthotropic thermal conductivity of the material (in watts/m °C).
TH_Specific_Heat	Real	Range (TH_Specific_Heat >= 0.0) Optional (Specific heat is either unkown or not required for the translation.)	Specific heat for the material (in kJ/kg °C).
TH_Surf_Emissivity	Real	Range (TH_Surf_Emissivity >= 0.0) Optional (Surface emissitivity is either unkown or not required for the translation.)	Thermal surface emissivity of the material, defined as the emissive power of the surface divided by the emissive power of a black body at the same temperature.
TH_Surf_Roughness	Real	Range (TH_Surf_Roughness >= 0.0) Optional (Surface roughness is either unkown or not required for the translation.)	Surface roughness, specified as the average roughness height (in the part units).
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Board_Part (*Ref. by Name*), Electrical_Part (*Ref. by Name*), Mechanical_Part (*Ref. by Name*), Panel_Part (*Ref. by Name*))			

6.24 Mechanical_Part

The Mechanical_Part entity is used to represent the physical characteristics of a mechanical part. The difference between an electrical part and a mechanical part is that an electrical part may contain a set of pins, and is electrically connected to the board or panel part.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Part_Name	Ent_Name	- - -	Name of the mechanical part.
Units	Enum	EList ("Inch", "MM", "Global") Default ("Global")	Length units for the entities that make up the mechanical part. Global means that the units are equal to the default units specified in the IDF Header.
Type	Ext_Enum	EList ("Unspecified")	Mechanical part type.
Mnt_Shape[1..?]	Entity	RList (Cavity, Cutout, Extrusion) Informal (At least one extrusion entitiy is required.)	Mounting side shape of the mechanical part.
Opp_Shape[0..?]	Entity	RList (Cavity, Cutout, Extrusion) Informal (If there is an opposite side shape, at least one extrusion entitiy is required.) Optional (There is no opposite side shape.)	Opposite side shape of the mechanical part.
Annos[0..?]	Entity	RList (Annotation) Optional (Annotations are either not in the part, or are not required for the translation.)	List of annotations contained in the mechanical part.
Material	Ref_Name	RList (Material) Optional (Material characteristics are not required for the translation.)	Mechanical part material characteristics.
TH_Model	Ref_Name	RList (Thermal_Model) Optional (Thermal model is not required for the translation.)	Thermal model for the mechanical part.
Properties[0..?]	NValue	NList (TH_Air_Gap, TH_Heat_Dis_Nominal, TH_Heat_Dis_Min_Max, U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (*Independent*, Mechanical_Part_Instance (*Ref. by Name*))			

Usage Notes:

- The mechanical part has its own coordinate system in which its features are defined.

- A mechanical part must have a mounting side shape. In addition, it may optionally have an opposite side shape (to represent a fastener or retainer, for example).
- A printed board or panel feature (a logo, for example) is not considered a mechanical part. It should be represented as a graphic or a figure.
- The extrusions that make up either the mounting side shape or the opposite side shape may or may not be physically joined (share a common boundary curve). See Figure 6.5.
- Board or panel features that are associated with the mechanical part (holes and keepouts, for example) are defined within a footprint, which is associated with the each mechanical part instance. Consequently, the board or panel features are not defined in the mechanical part itself.

6.25 Mechanical_Part_Instance

The Mechanical_Part_Instance entity is used to represent the occurrence of a mechanical part, and optionally an associated footprint, in a board or panel assembly. Mechanical part instances are optionally uniquely identified in the design by their reference designators.

Attribute Name	Type	Constraints	Descripton
Entity_ID	Ent_ID	- - -	IDF entity ID.
Part_Name	Ref_Name	RList (Mechanical_Part)	Name of the mechanical part that is instanced.
Part_Number	String	Optional (Part number is not known or there is no part number.)	Part number for the instance.
In_BOM	Boolean	Default (“True”)	Bill of material flag. True means that the part instance is included in the bill of material.
Refdes	String	Default (“Unassigned”)	Reference designator for the instance.
Old_Refdes	String	Optional (Refdes has not been changed by the sending application.)	Previous reference designator for the instance.
Lock	Ext_Enum	EList (“MCAD”, “ECAD”, “None”) Default (“None”)	Instance lock status.
XY_Loc[2]	Real	- - -	XY location of the instance.
Side	Enum	EList (“Top”, “Bottom”)	Side of the board or panel on which the instance is mounted. Bottom side instances are rotated 180 degrees about the Y-axis, at the instance location.
Rotation	Real	Range (0.0 <= Rotation < 360.0)	Rotation (in degrees) of the mechanical part instance, about the Z-axis, at the instance location. Rotation is counter-clockwise with respect to the instance location.
Mnt_Offset[1..2]	Real	Range (Mnt_Offset[0..1] >= 0.0) Default (0.0, 0.0)	Mounting offset from the board or panel (from the nominal mounting height). The first value is for the mounting side shape. The second value is for the opposite side shape, if it exists.

Attribute Name	Type	Constraints	Descripton
Variant_Use	Enum	EList (“All”, “Inclusion”, “Exclusion”) Optional (There are no design variants for the board assembly.)	Use of the instance with respect to design variants. All — Instance is in all design variants. Inclusion — Instance is only in the design variants specified in the variants list. Exclusion — Instance is in all design variants, except those specified in the vairants list.
Variants[0..?]	String	Optional (There are no design variants, or the variant use value is equal to All.)	List of design variant names that the instance is included in or excluded from, depending on the variant use value.
Footprint	Ref_ID	Optional (Footprint is not required for the translation.) Informal (The footprint must be contained in the board part.)	Footprint that is associated with the instance.
Properties[0..?]	NValue	NList (TH_Air_Gap, TH_Heat_Dis_Nominal, TH_Heat_Dis_Min_Max, U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Board_Assembly, Panel_Assembly, Sublayout (*Ref. by ID*))			

Usage Notes:

- The mechanical part instance is located in the coordinate system of the assembly that contains it.
- The placement of an mechanical part instance is determined as follows:
  - The instance is placed at the XY location, with respect to the assembly’s origin and in the assembly’s units.
  - The instance origin is aligned (along the Z-axis) with the surface of the board or panel part. The surface may be the top, bottom, or exposed surface of a cavity that extends into the board or panel part from the top or bottom.
  - If the board surface is the bottom or bottom-facing cavity surface, the instance is rotated 180 degrees about the Y-axis, at the instance location.
  - The instance is then rotated about the Z-axis, at the instance location.
  - If a mounting offset is specified, the instance is offset from the surface of the board, the specified amount (in the assembly’s units).
- Rotation is always counterclockwise, with respect to the instance location. See Figure 6.6, which illustrates several instance rotations.
- Each alternate or substitute for a component instance must reference a common footprint.
- If the Old_Refdes attribute is present, it means that the sending application changed the Refdes for this instance. The receiving application must use the Old_Refdes value to find the corresponding part instance, and modify its reference designator accordingly.

6.26 Pad

The Pad entity is used to represent a set of one or more conductor shapes for connecting a trace to a component pin, via, bond wire, testpoint, or external connector contact. In addition, a pad may be used as a contact point for a test probe.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Feature_ID	String	Default (“Unassigned”)	Unique feature identifier.
Lock	Ext_Enum	EList (“MCAD”, “ECAD”, “None”) Default (“None”)	Feature lock status.
Type	Ext_Enum	EList (“Bondwire”, “Connector”, “Pin”, “Test”, “Unspecified”, “Via”)	Pad type.
Assoc_Pin_ID	String	Optional (The pad either does not have an associated pin, or it is not required for the translation.)	Associated pin ID.
Testpoint	Boolean	Default (“False”)	Testpoint flag. A value of true means that the pad is used as a testpoint.
XY_Loc[2]	Real	- - -	XY location of the pad.
Rotation	Real	Range (0.0 <= Rotation < 360.0)	Rotation (in degrees) of the pad’s geometry, about the Z-axis, at the pad’s location. Rotation is counter-clockwise with respect to the instance location.
Layer	Ref_Name	RList (Physical_Layer) Informal (Layer attribute must specify a physical layer name or a predefined layer alias.)	Board or panel layer on which the pad is located.
Geometry[1..?]	Ref_ID	RList (Circle, Circular_Arc, Polycurve, Polycurve_Area, Polygon, Polyline)	List of entities that represent the pad geometry.
Net_Name	String	Optional (Pad is either not electrically connected, or the net name is not required for the translation.)	Net name that is associated with the pad.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Board_Part, Figure, Footprint, Panel_Part)			

Usage Notes:

- The pad is located in the coordinate system of the part that contains it.

- Typically, a pad's shape is represented by either a circle (represented by a Circle entity), a square or rectangle (represented by a Polygon entity), or an oval (represented by a Polycurve_Area entity). However, the pad shape may be complex and need to be represented by a set of geometric entities.
- The geometry referenced by the pad, is defined in the coordinate system of the pad. Consequently, the geometry must be transformed according to the location, side, and rotation of the pad, to be properly located.
- The geometry entities of a pad are referenced entities. Therefore, they can be re-used by other pads. Typically, there are many pads on a board or panel that can use the same geometry.
- Figure 6.14 illustrates the representation of a pad.

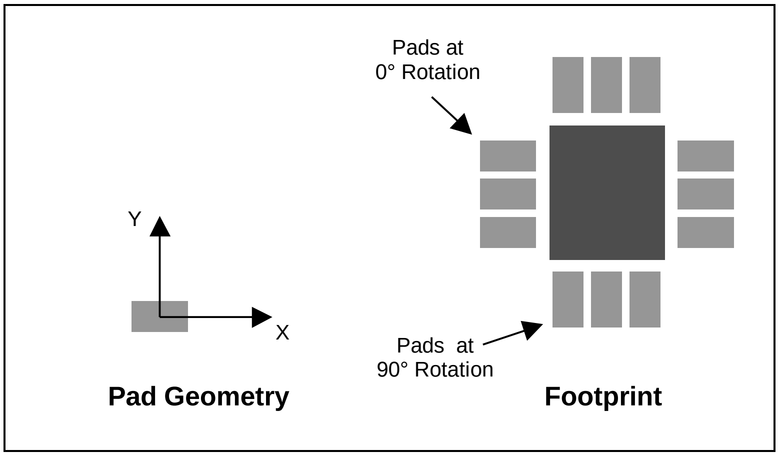


Figure 6.14 Pad Representation

- The pad types are defined as follows:

Type	Description
Bondwire	Component bond wire pad.
Connector	Printed connector pad.
Pin	Component pin pad.
Test	Test probe pad.
Unspecified	Pad type is unkown or not required.
Via	Via pad.

6.27 Panel_Assembly

The Panel_Assembly entity is used to represent a manufacturing panel consisting of a panel part, one or more board assembly instances (of the same or different board assemblies), and optionally component instances.

A panel assembly may also represent a subpanel, in which case it can be instantiated in a panel assembly via a panel assembly instance.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Assy_Name	Ent_Name	- - -	Name of the panel assembly.
Part_Number	String	Optional (Part number is not known.)	Part number for the panel assembly.
Units	Enum	EList (“Inch”, “MM”, “Global”) Default (“Global”)	Length units for defining the location of instance entities within the panel assembly. Global means that the units are equal to the default units specified in the IDF Header.
Type	Ext_Enum	EList (“Combi”, “Family”, “Step_Repeat”, “Unspecified”)	Panel assembly type. See detailed descriptions below.
Panel_Inst	Entity	RList (Panel_Part_Instance) Optional (Panel instance is not required when updating board assembly instance or component instance locations.)	Panel part instance.
Panel_Assys[0..?]	Entity	RList (Panel_Assembly_Instance) Optional (Panel assembly either does not have any subpanels, or they are not required for the translation.)	List of panel assembly instances (subpanels).
Board_Assys[0..?]	Entity	RList (Board_Assembly_Instance) Optional (Board assembly instances are not required for the translation.)	List of board assembly instances.
Comp_Insts[0..?]	Entity	RList (Electrical_Part_Instance, Mechanical_Part_Instance) Optional (Panel assembly either does not have any component instances, or they are not required for the translation.)	List of electrical and mechanical part instances (other than those on board assemblies in the panel).
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.

Attribute Name	Type	Constraints	Description
Entity Constraints:			
Informal (The panel part instance must be located at XY_Loc = 0.0, 0.0 and Rotation = 0.0)			
Informal (Panel parts and board parts that are part of a panel assembly, must have identical layer stackup and thickness, and be positioned such that their corresponding surfaces are coplanar.)			
Informal (The panel assembly must include at least one panel assembly instance, board assembly instance, or component instance.)			
Where_Used (*Independent,* Panel_Assembly_Instance (*Ref. by Name*))			

Usage Notes:

- The panel assembly has its own coordinate system in which its part and assembly instances are located.
- The panel assembly types are defined as follows:

Type	Description
Combi	A combination of step-and-repeat and family, often with subpanels.
Family	The board assembly instances reference a family of board assemblies.
Step_Repeat	All board assembly instances reference the same board assembly.
Unspecified	Panel type is unkown or not required.

6.28 Panel_Assembly_Instance

The Panel_Assemby_Instance entity is used to represent the occurrence of a panel assembly (subpanel) in a panel assembly.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Assy_Name	Ref_Name	RList (Panel_Assembly)	Name of the panel assembly that is instanced.
Refdes	String	Default (“Unassigned”)	Reference designator for the panel assembly instance.
Old_Refdes	String	Optional (Refdes has not been changed by the sending application.)	Previous reference designator for the panel assembly instance.
Lock	Ext_Enum	EList (“MCAD”, “ECAD”, “None”) Default (“None”)	Instance lock status.
XY_Loc[2]	Real	- - -	XY location of the panel assembly instance.
Rotation	Real	Range (0.0 <= Rotation < 360.0)	Rotation (in degrees) of the panel assembly instance, about the Z-axis, at the instance location. Rotation is counter-clockwise with respect to the instance location.
Mirror	Boolean	Default (“False”)	Mirror flag for the panel assembly instance. A value of true means that the panel assembly instance is rotated 180 degrees about the Y-axis, at the instance location.
Footprint	Ref_ID	Optional (Footprint is not required for the translation.) Informal (The footprint must be contained in the panel part.)	Footprint that is associated with the panel assembly instance.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Panel_Assembly)			

Usage Notes:

- The panel assembly instance is located in the coordinate system of the panel assembly that contains it.
- If the Old_Refdes attribute is present, it means that the sending application changed the Refdes for this instance. The receiving application must use the Old_Refdes value to find the corresponding assembly instance, and modify its reference designator accordingly.

6.29 Panel_Part

The Panel_Part entity is used to represent the "bare" panel (without any components assembled to it). It includes the set of product and design features that are directly associated with the "bare" panel.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Part_Name	Ent_Name	- - -	Name of the panel part.
Units	Enum	EList ("Inch", "MM", "Global") Default ("Global")	Length units for the entities that make up the panel part. Global means that the units are equal to the default units specified in the IDF Header.
Type	Ext_Enum	EList ("Unspecified")	Panel part type.
Shape[1..?]	Entity	RList (Extrusion) Informal (The bottom surface of the extrusion in which the XY origin is located, must be at Z = 0.0.)	Extrusion(s) that make up the panel part's shape
Features[0..?]	Entity	RList (Annotation, Cavity, Cutout, Figure, Filled_Area, Graphic, Hole, Keepin, Keepout, Pad, Trace) Optional (Panel features are not required for the translation.)	List of features that are contained in the panel part.
Footprints[0..?]	Entity	RList (Footprint) Optional (Footprints are not required for the translation.)	List of footprints that are associated with the panel part.
Material	Ref_Name	RList (Material) Optional (Material characteristics are not required for the translation.)	Panel part material characteristics.
Properties[0..?]	NValue	NList (Avg_M2V_Ratio, U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Informal (All cutouts, cavities, and holes must be located within the outer boundary of the panel part's extrusion(s).) Where_Used (*Independent*, Panel_Part_Instance (*Ref. by Name*))			

Usage Notes:

- The panel part has its own coordinate system in which its features are defined.
- It is assumed that a panel part has the same physical layer stackup and thickness as the board part(s) that are internal to it. Therefore, there are no physical layers associated with the panel part.

- Keepins and keepouts that are contained in a panel part, affect the board parts that are located in the panel assembly, as well as the panel part itself. The portion of a keepout that falls on a board assembly in the panel assembly should be “mapped” to the board part for use in that board’s design.

6.30 Panel_Part_Instance

The Panel_Part_Instance entity is used to represent the occurrence of a panel part in a panel assembly.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Part_Name	Ref_Name	RList (Panel_Part)	Name of the panel part that is instanced.
Part_Number	String	Optional (Part number is not known.)	Part number for the panel part instance.
Refdes	String	Default (“Unassigned”)	Reference designator for the panel part instance.
Old_Refdes	String	Optional (Refdes has not been changed by the sending application.)	Previous reference designator for the panel part instance.
XY_Loc[2]	Real	- - -	XY location of the panel part instance.
Rotation	Real	Range (0.0 <= Rotation < 360.0)	Rotation (in degrees) of the panel part instance, about the Z-axis, at the instance location. Rotation is counter-clockwise with respect to the instance location.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Informal (The panel part instance must be located at XY_Loc = 0.0, 0.0 and Rotation = 0.0 in the panel assembly.) Where_Used (Panel_Assembly)			

Usage Notes:

- The panel part instance is located in the coordinate system of the panel assembly that contains it.
- If the Old_Refdes attribute is present, it means that the sending application changed the Refdes for this instance. The receiving application must use the Old_Refdes value to find the corresponding part instance, and modify its reference designator accordingly.

6.31 Physical_Layer

The Physical_Layer entity is used to represent the physical characteristics and arrangement (stackup) of the physical layers that make up a board part.

The details of the physical layer stackup are not always required. Consequently, the Physical_Layer entity is optional. If there are no physical layers associated with the board part, layer based features must refer to one of the predefined layer aliases (defined below).

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Layer_Name	Ent_Name	Informal (Layer names can't be the same as any of predefined layer aliases.)	Name of the physical layer.
Type	Ext_Enum	EList ("Conductive", "Dielectric", "Unspecified")	Physical layer type.
Position	Integer	Range (Position > 0) Informal (Position is defined sequentially from 1 to N, starting with the top layer.)	Position of the layer within the stackup of layers for the board part.
Thickness	Real	Range (Thickness >= 0.0) Informal (The sum of each layer's thickness must equal the overall thickness of the board.)	Thickness (in the length units of the board part) of the layer.
Material	Ref_Name	Optional (Material is not required for the translation.)	Material for the layer.
Properties[0..?]	NValue	NList (*User-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Board_Part, Electrical_Part_Instance (*Ref. by Name*), Filled_Area (*Ref. by Name*), Graphic (*Ref. by Name*), Hole (*Ref. by Name*), Keepin (*Ref. by Name*), Keepout (*Ref. by Name*), Pad (Ref. by Name), Trace (Ref. by Name))			

Usage Notes:

- Physical layers are required to support embedded components, blind pin holes, blind and buried via holes, conductors and graphics on internal layers, and keepins and keepouts that affect internal layers.

- The following table specifies the predefined physical layer aliases that must be used if there are no physical layers associated with the board.

Alias	Description
Top	The top physical layer (or surface) of the board or panel.
Bottom	The bottom physical layer (or surface) of the board or panel.
Both	Both the top and bottom layers of the board or panel.
Inner	All of the inner conductor layers of the board.
All	All of the conductor layers of the board, including the top and bottom surfaces of the board or panel.

- The physical layer types are defined as follows:

Type	Description
Conductive	Physical layer is a conductive material that conducts electrical signals
Dielectric	Physical layer is a non-conductive material that acts as an insulator between conductive layers.
Unspecified	Physical layer type is unkown or not required.

6.32 Pin

The Pin entity is used to represent the location of a physical pin (terminal) on an electrical part.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Pin_ID	String	- - -	Pin identifier (name or number).
Type	Ext_Enum	EList (“Blind”, “Surface”, “Thru”, “Unspecified”)	Pin type.
XY_Loc[2]	Real	- - -	XY location of the pin (with respect to the electrical part’s coordinate system). Location represents the center point, where the pin mates with the board or panel part.
Properties[0..?]	NValue	NList (*User-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Electrical_Part)			

Usage Notes:

- A pin is defined in the coordinate system of the Electrical Part entity that contains it.
- The pin type for a pin on a printed electrical part is surface.
- The receiving application must determine how to display the pin location (using a graphic marker or the pin ID, for example).
- Figure 6.15 illustrates the relationship between a pin, the electrical part and the footprint pad and hole.

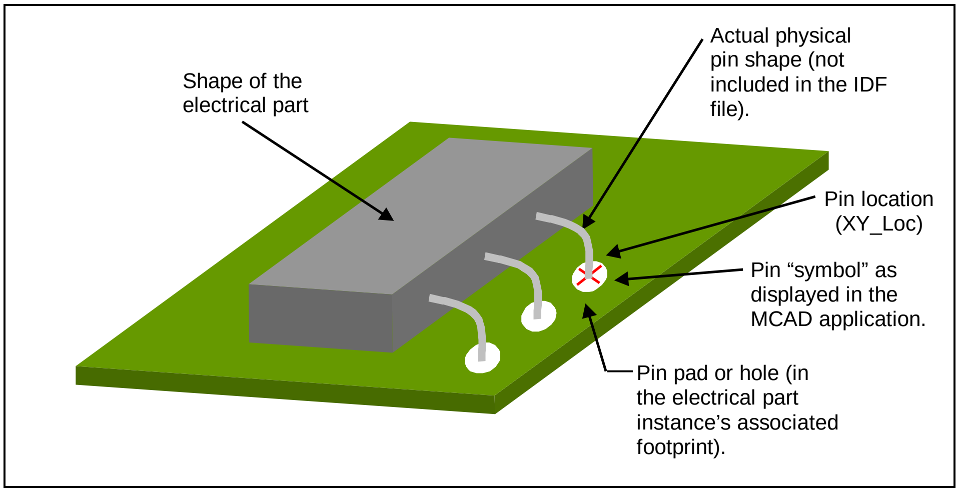

Shape of the electrical part
Actual physical pin shape (not included in the IDF file).
Pin location (XY_Loc)
Pin “symbol” as displayed in the MCAD application.
Pin pad or hole (in the electrical part instance’s associated footprint).

Figure 6.15 Pin Representation

- The pin types are defined as follows:

Type	Description
Blind	Pin is connected to a blind (partial) hole.
Surface	Pin is connected to a pad on the surface of the board or panel.
Thru	Pin is connected to a thru hole.
Unspecified	Pin type is unkown or not required.

6.33 Polycurve

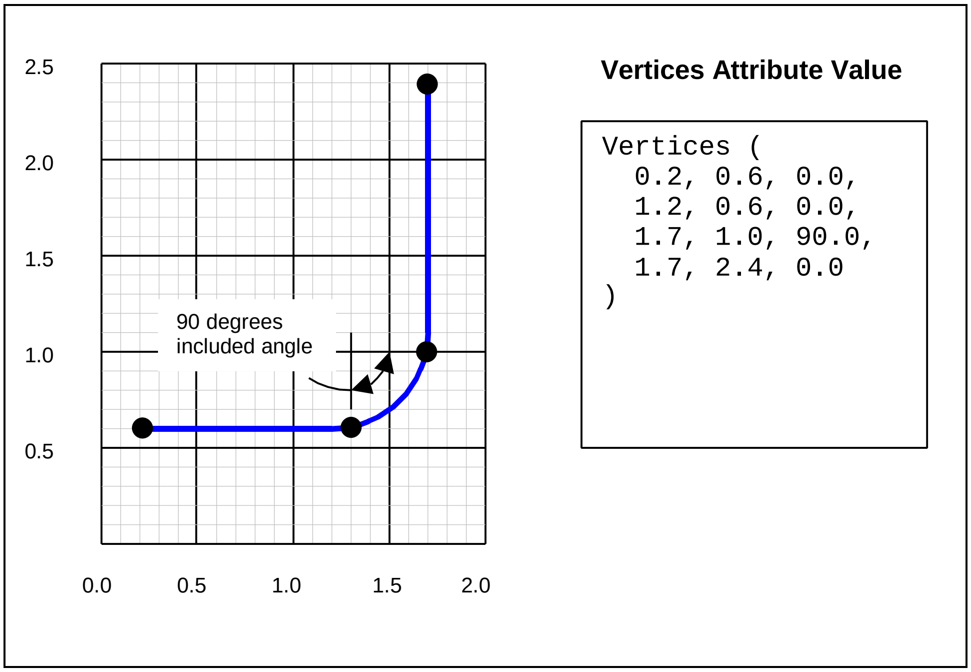
The Polycurve entity is used to represent an XY planar piecewise curve (open or closed), consisting of either linear and/or circular arc segments. It is defined by a series of vertices, where each vertex represents an XY location, and an included angle.

Attribute Name	Type	Constraints	Descripton
Entity_ID	Ent_ID	- - -	IDF entity ID.
Line_Font	Enum	EList (“Solid”, “Center”, “Phantom”, “Dashed_Long”, “Dashed_Short”, “Dotted”) Default (“Solid”)	Line font of the curve.
Line_Color[3]	Real	Range (0.0 <= Line_Color[0..2] <= 100.0) Optional (Line color is not required for the translation.)	Color of the curve, expressed as percentages of intensity of red, green, and blue (RGB), respectively.
Width	Real	Range (Width > 0.0) Optional (Line width is not required for the translation.)	Line width for the curve.
Start_Style	Enum	EList (“Round”, “Square”, “None”) Default (“None”)	End condition for the start of the curve.
End_Style	Enum	EList (“Round”, “Square”, “None”) Default (“None”)	End condition for the end of the curve.
Vertices[3..?][3]	Real	Range (-360.0 < Vertices[0..?][2] < 360.0) Informal (A polycurve must contain at least one circular arc segment.) Informal (Adjacent vertices may not be coincident.)	Set of vertices that make up the Polycurve. The Vertices attribute is defined as follows: Vertices[][0] X location of vertex Vertices[][1] Y location of vertex Vertices[][2] Included angle (in degrees) for the segment that connects to the previous vertex
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Annotation, Graphic, Pad (*Ref. by ID*), Trace)			

Usage Notes:

- The polycurve is defined in the coordinate system of the part, figure, footprint, or pad where it is used.

- The interpretation of the included angle (Vertices[][2]) is as follows:
  - If the included angle for a given vertex is 0.0, a line segment connects the vertex with the previous vertex.
  - If the included angle is negative, a clockwise arc connects the vertex with the previous vertex.
  - If the included angle is positive, a counter-clockwise arc connects the vertex with the previous vertex.
  - The included angle of the first vertex is not used.
- Figure 6.16 illustrates the polycurve vertex representation.



Vertices Attribute Value

```
Vertices (
  0.2, 0.6, 0.0,
  1.2, 0.6, 0.0,
  1.7, 1.0, 90.0,
  1.7, 2.4, 0.0
)
```

Figure 6.16 Polycurve Vertex Representation

6.34 Polycurve_Area

The Polycurve_Area entity is used to represent an XY planar area, bounded by a piecewise curve, consisting of either linear and/or circular arc segments. It is defined by a series of vertices, where each vertex represents an XY location, and an included angle.

Attribute Name	Type	Constraints	Descripton
Entity_ID	Ent_ID	- - -	IDF entity ID.
Line_Font	Enum	EList (“Solid”, “Center”, “Phantom”, “Dashed_Long”, “Dashed_Short”, “Dotted”) Default (“Solid”)	Line font of the boundary curve.
Line_Color[3]	Real	Range (0.0 <= Line_Color[0..2] <= 100.0) Optional (Line color is not required for the translation.)	Color of the boundary curve, expressed as percentages of intensity of red, green, and blue (RGB), respectively.
Fill_Color[3]	Real	Range (0.0 <= Fill_Color[0..2] <= 100.0) Optional (Fill color is not required for the translation.)	Fill color of the bounded area, expressed as percentages of intensity of red, green, and blue (RGB), respectively.
Vertices[3..?][3]	Real	Range (-360.0 < Vertices[0..?][2] < 360.0) Informal (A polycurve must contain at least one circular arc segment.) Informal (Adjacent vertices may not be coincident.) Informal (The first vertex and the last vertex must be coincident, to explicitly close the polycurve area.) Informal (The boundary of the polycurve area may not self-intersect, or become coincident at any location other than the start and end vertices.)	Set of vertices that make up the Polycurve. The Vertices attribute is defined as follows: Vertices[][0] X location of vertex Vertices[][1] Y location of vertex Vertices[][2] Included angle (in degrees) for the segment that connects to the previous vertex
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Annotation, Cavity, Cutout, Extrusion, Filled_Area, Graphic, Hole (*Ref. by ID*), Keepin, Keepout, Pad (*Ref. by ID*))			

Usage Notes:

- The polycurve area is defined in the coordinate system of the part, figure, footprint, pad, or hole where it is used.

- The interpretation of the included angle (Vertices[][2]) is as follows:
  - If the included angle for a given vertex is 0.0, a line segment connects the vertex with the previous vertex.
  - If the included angle is negative, a clockwise arc connects the vertex with the previous vertex.
  - If the included angle is positive, a counter-clockwise arc connects the vertex with the previous vertex.
  - The included angle of the first vertex is not used.
- See Figure 6.16 for an illustration of the polycurve area vertex representation.

6.35 Polygon

The Polygon entity is used to represent an XY planar area, bounded by a piecewise linear curve.

Attribute Name	Type	Constraints	Descripton
Entity_ID	Ent_ID	- - -	IDF entity ID.
Line_Font	Enum	EList (“Solid”, “Center”, “Phantom”, “Dashed_Long”, “Dashed_Short”, “Dotted”) Default (“Solid”)	Line font of the boundary curve.
Line_Color[3]	Real	Range (0.0 <= Line_Color[0..2] <= 100.0) Optional (Line color is not required for the translation.)	Color of the boundary curve, expressed as percentages of intensity of red, green, and blue (RGB), respectively.
Fill_Color[3]	Real	Range (0.0 <= Fill_Color[0..2] <= 100.0) Optional (Fill color is not required for the translation.)	Fill color of the bounded area, expressed as percentages of intensity of red, green, and blue (RGB), respectively.
XY_Pts[3..?][2]	Real	Informal (Adjacent points may not be coincident.) Informal (The first point and the last point must be coincident, to explicitly close the polygon.) Informal (The boundary of the polygon may not self-intersect, or become coincident at any location other than the start and end points.)	Set of XY points that make up the polygon.
Properties[0..?]	NValue	NList (*User-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Annotation, Cavity, Cutout, Extrusion, Filled_Area, Graphic, Hole (*Ref. by ID*), Keepin, Keepout, Pad (*Ref. by ID*))			

Usage Notes:

- The polygon is defined in the coordinate system of the part, figure, footprint, pad, or hole where it is used.

6.36 Polyline

The Polyline entity is used to represent an XY planar piecewise linear curve (open or closed).

Attribute Name	Type	Constraints	Descripton
Entity_ID	Ent_ID	- - -	IDF entity ID.
Line_Font	Enum	EList (“Solid”, “Center”, “Phantom”, “Dashed_Long”, “Dashed_Short”, “Dotted”) Default (“Solid”)	Line font of the curve.
Line_Color[3]	Real	Range (0.0 <= Line_Color[0..2] <= 100.0) Optional (Line color is not required for the translation.)	Color of the curve, expressed as percentages of intensity of red, green, and blue (RGB), respectively.
Width	Real	Range (Width > 0.0) Optional (Line width is not required for the translation.)	Line width for the curve.
Start_Style	Enum	EList (“Round”, “Square”, “None”) Default (“None”)	End condition for the start of the curve.
End_Style	Enum	EList (“Round”, “Square”, “None”) Default (“None”)	End condition for the end of the curve.
XY_Pts[2..?][2]	Real	Informal (Adjacent points may not be coincident.)	Set of XY points that make up the polyline.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Annotation, Graphic, Leader, Pad (*Ref. by ID*), Trace)			

Usage Notes:

- The polyline is defined in the coordinate system of the part, figure, footprint, or pad, where it is used.

6.37 Sublayout

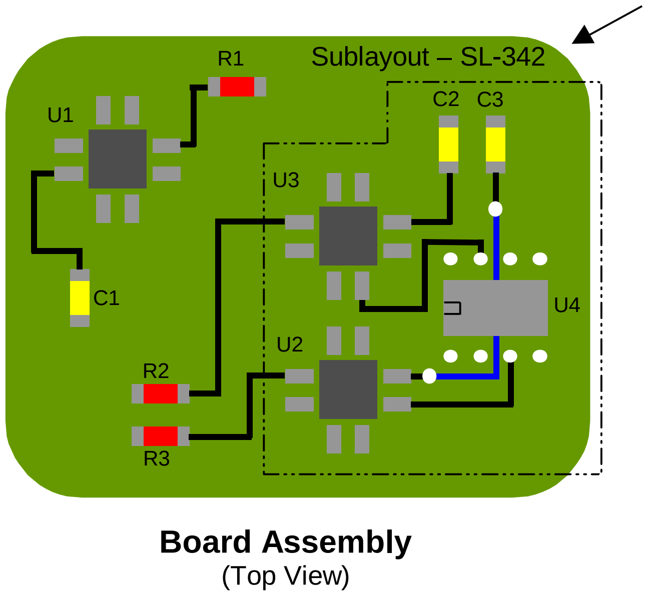
The Sublayout entity is used to represent a board assembly design re-use "block". It consists of one or more component instances, and optionally a set of board features (conductors, cavities, etc.) represented by a footprint.

Attribute Name	Type	Constraints	Descripton
Entity_ID	Ent_ID	- - -	IDF entity ID.
Sublayout_ID	String	Default ("Unassigned")	Unique sublayout identifier.
Lock	Ext_Enum	EList ("MCAD", "ECAD", "None") Default ("None")	Feature lock status.
Comp_Insts[1..?]	Ref_ID	- - -	Component instances that are associated with the sublayout.
Footprint	Ref_ID	Optional (There is either no footprint associated with the sublayout, or the footprint is not required for translation.) Informal (The footprint must be contained in the board part.)	Footprint that is associated with the sublayout.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Board_Assembly (Ref by Name))			

Usage Notes:

- A sublayout is used to maintain the relative position of component instances and board features. If any one of the component instances or board features is moved, all of the entities referenced by the sublayout move with it.
- A sublayout is typically used to ensure that a set of component instances maintains their relative positions with one another in the board assembly. In this case, the sublayout does not include a footprint.
- The sublayout is essentially a grouping entity, in that it does not have any location information associated with it. Consequently, it does not directly affect the location of the component instances or the board features; their locations are defined with respect to the board assembly or board part.
- Component instances in the sublayout are referenced by their entity ID rather than their reference designator, since mechanical part instances may not be assigned reference designators, and the reference designators for some electrical part instances may be unassigned.
- The board features in the sublayout footprint do not include the board features that are associated with the sublayout component instances.

- Figure 6.17 illustrates the representation of a sublayout.



Sublayout SL-342 includes component instances U2, U3, U4, C2 & C3, plus the routing connections between these five components.

The sublayout references the component instances and the associated footprint.

The footprint is contained in the board part.

The footprint consists of the traces and vias that connect the component instances. They are only contained in the footprint, which is contained in the board part.

Figure 6.17 Sublayout Representation

6.38 Text

The Text entity is used to represent a text string.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Color[3]	Real	Range (0.0 <= Color[0..2] <= 100.0) Optional (Text color is not required for the translation.)	Color of the text string, expressed as percentages of intensity of red, green, and blue (RGB), respectively.
Text_String	String	Informal (Text string must contain one or more characters.)	Text string to be displayed.
XY_Loc[2]	Real	- - -	XY location for the text string. Location specifies where the origin of the text string is to be located.
Origin	Enum	EList (“Top_Left”, “Top_Center”, “Top_Right”, “Middle_Left”, “Middle_Center”, “Middle_Right”, “Bottom_Left”, “Bottom_Center”, “Bottom_Right”) Default (“Bottom_Left”)	Origin of the text string.
Height	Real	Range (Height > 0.0)	Character height.
Width	Real	Range (Width > 0.0) Optional (Width of the text string is not required for the translation, it must be evaluated by the receiving application, based on the height and font.)	Width of the overall text string (often referred to as the text box width).
Rotation	Real	Range (0.0 <= Rotation < 360.0)	Rotation (in degrees) of the text string, about the Z-axis, at the text’s origin. Rotation is counter-clockwise with respect to the text location.
Slant	Real	Range (0.0 < Slant < 180.0) Default (90.0)	Slant angle of the characters within the text string.
Mirror	Boolean	Default (“False”)	Mirror flag. A value of true means that the text string is mirrored about the Y-axis, at the text’s origin.
Font	String	Optional (The specific font is not required for the translation, one must be assigned by the receiving application.) Informal (The font may be any text font that is recognized within the international printng industry.)	Character font for the text string.

Attribute Name	Type	Constraints	Description
Properties[0..?]	NValue	NList (*User-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Annotation, Graphic)			

Usage Notes:

- Text is defined in the coordinate system of the part in which its parent entity is located.
- Specific rules for text placement are as follows:
  - – Text is located at the XY location, with respect to the part's origin, and in the part's units.
  - – Text is mirrored about the Y-axis, at the text origin (if the Mirror attribute is True).
  - – Text is rotated about the Z-axis, at the text origin.
- Figure 6.18 illustrates the representation of text.

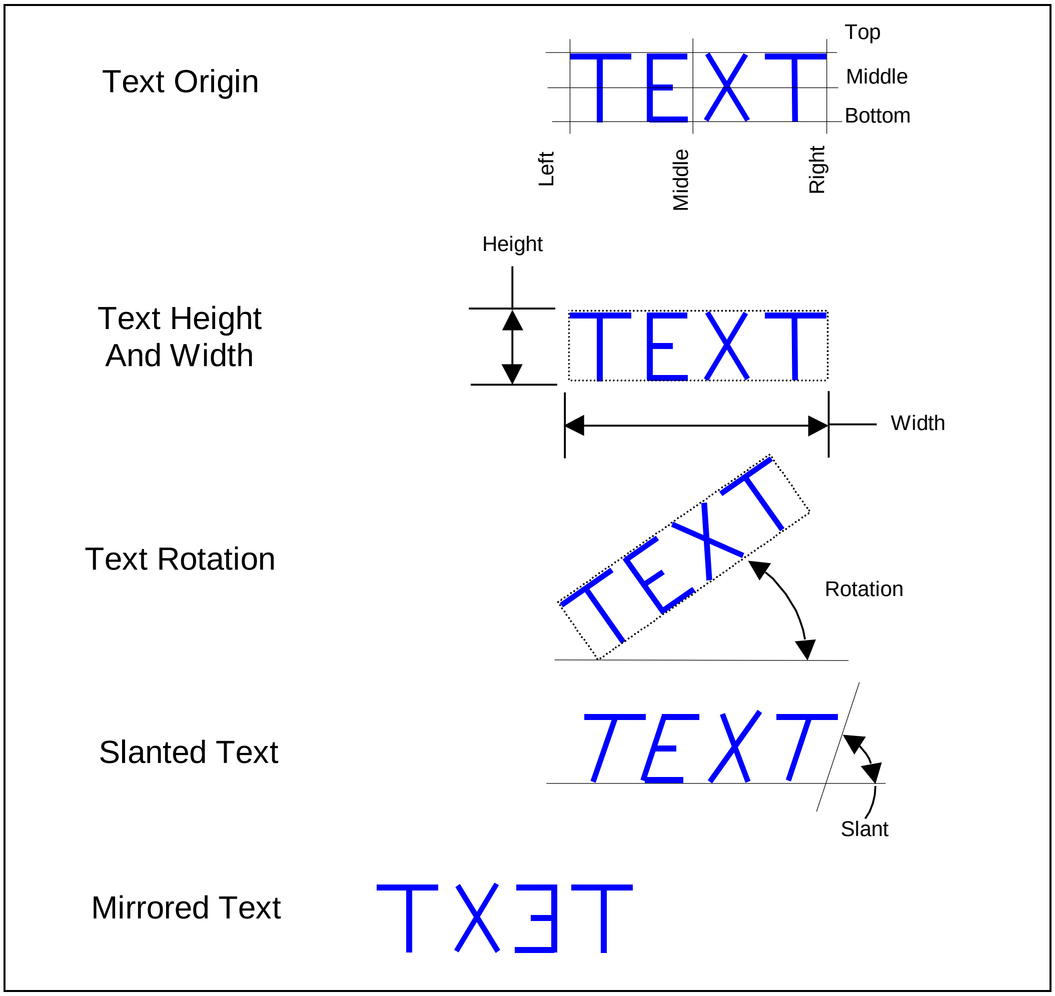


Figure 6.18 Text Representation

6.39 Thermal_Model

The Thermal_Model entity is used to represent a precise thermal model for electrical and mechanical parts. It consists of a thermal resistance-capacitance network that defines the thermal characteristics of a component. In addition, it defines how the network is connected to the board and the environment.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Model_Name	Ent_Name	- - -	Thermal model name.
TH_CVs[2..?]	Entity	RList (Thermal_CV)	List of thermal capacitance values.
TH_RVs[1..?]	Entity	RList (Thermal_RV)	List of thermal resistance values.
Properties[0..?]	NValue	NList (*User-defined*) Optional ()	Associated property values.
Entity Constraints: Where_Used (Electrical_Part (*Ref. by Name*), Mechanical_Part (*Ref. by Name*))			

Usage Notes:

- The thermal model contains two lists of data, a list of thermal capacitance values, and a list of thermal resistance values. The thermal capacitance values are defined for a set of two or more vertices. Each thermal resistance value is defined for a specific pair of vertices.
- Figure 6.19 illustrates the use of a thermal model for a simple plastic packaged electrical part. The thermal model consists of a simple resistance-capacitance network, consisting of 5 vertices and 4 resistance values.

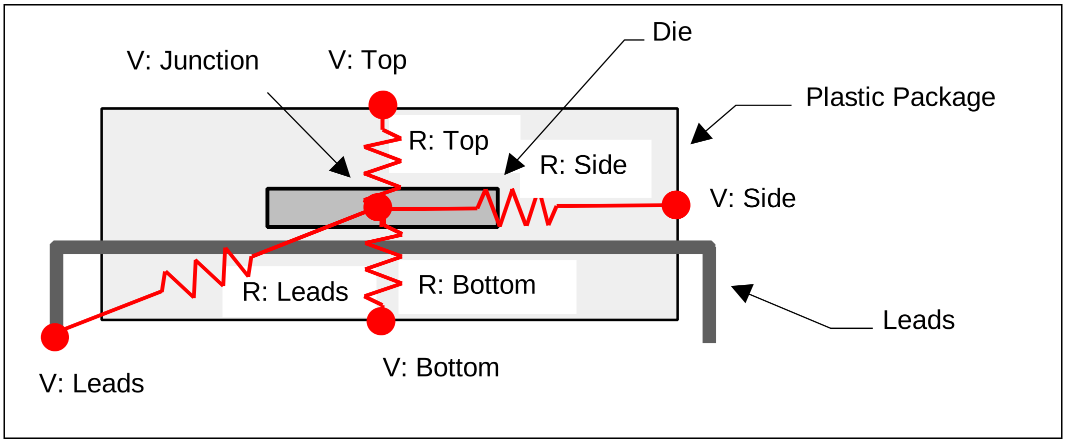


Figure 6.19 Thermal Model Representation

- The following table describes the vertices for the thermal model illustrated above.

Vertex Name	Description	Type	Capacitance	Area
Junction	Chip Die	Junction	2.53	0.0
Top	Top of the package	Ex_Environment	0.0	1,600.0
Bottom	Bottom of the package	Ex_Environment	0.0	1,600.0
Side	Side of the package	Ex_Environment	0.0	560.0
Leads	Leads for the package	Ex_Board	0.0	1,615.6

- The following table describes the resistance values for the thermal model illustrated above.

Resistance Name	Description	Resistance
Top	Conduction from the junction to the top surface of the case	1.56
Bottom	Conduction from the junction to the bottom surface of the case.	1.32
Side	Conduction from the junction to the side surfaces of the case.	2.93
Leads	Conduction from the junction to the board.	4.58

- The following sample file is for the thermal model illustrated above.

```
Thermal_Model (                  /* RC Thermal Model for Plastic Package */
  Entity_ID (#101),
  Model_Name (“RC_Type_A”),
  TH_CVs (                                    /* Vertices for RC Network */
    Thermal_CV (
      Entity_ID (#201),
      Vtx_Name ("Junction"),
      Type (“Junction”),
      Capacitance (2.53),             /* J/C assumes all Cap at Junction */
      Heat_Load (1.0),                           /* All heat at Junction */
      Area (0.0)
    ); /* End Thermal_CV */
    Thermal_CV (
      Entity_ID (#202),
      Vtx_Name ("Top"),
      Type (“Ex_Environment”),
      Capacitance (0.0),
      Heat_Load (0.0),
      Area (1600.0)                                /* Area = 40*40 MM**2 */
    ); /* End Thermal_CV */
    Thermal_CV (
      Entity_ID (#203),
      Vtx_Name ("Bottom"),
      Type (“Ex_Environment”),
      Capacitance (0.0),
      Heat_Load (0.0),
      Area (1600.0)                                /* Area = 40*40 MM**2 */
    ); /* End Thermal_CV */
    Thermal_CV (
      Entity_ID (#204),
      Vtx_Name ("Side"),
      Type (“Ex_Environment”),
      Capacitance (0.0),
      Heat_Load (0.0),
      Area (560.0)                    /* Area = 4 * sides * 3.5*40 MM**2 */
    ); /* End Thermal_CV */
    Thermal_CV (
      Entity_ID (#205),
      Vtx_Name ("Leads"),
      Type (“Ex_Board”),
      Capacitance (0.0),
      Heat_Load (0.0),
      Area (1615.6)         /* Area = 208 pins * 0.5*0.15 + 40*40  MM**2 */
    ); /* End Thermal_CV */
  ),
  TH_RVs (                           /* Resistance values for RC Network */
    Thermal_RV (
      Entity_ID (#301),
      Res_Name ("J2Top"),
      Vtx_1 (#201),
      Vtx_2 (#202),
      Resistance (1.56)                               /* Degrees C/Watts */
    ); /* End of Thermal_RV */
    Thermal_RV (
      Entity_ID (#302),
      Res_Name ("J2Bottom"),
      Vtx_1 (#201),
      Vtx_2 (#203),
      Resistance (1.32)                               /* Degrees C/Watts */
    ); /* End of Thermal_RV */
    Thermal_RV (
       Entity_ID (#303),
       Res_Name ("J2Side"),
       Vtx_1 (#201),
       Vtx_2 (#204),
       Resistance (2.93)                              /* Degrees C/Watts */
    ); /* End of Thermal_RV */
    Thermal_RV (
      Entity_ID (#304),
      Res_Name ("J2Leads"),
      Vtx_1 (#201),
      Vtx_2 (#205),
      Resistance (4.58)                               /* Degrees C/Watts */
    ); /* End of Thermal_RV */
  )
); /* End Thermal_Model */
```

6.40 Thermal_CV

The Thermal_CV entity is used to represent a list of thermal capacitance values for two or more vertices.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Vtx_Name	String	Optional (Vertex name is either not known, or not required for the translation.)	Vertex name.
Type	Ext_Enum	EList ("Ex_Board", "Ex_Component", "Ex_Environment", "Internal", "Junction", "Unspecified")	Vertex type.
Capacitance	Real	Range (Capacitance >= 0.0)	Thermal capacitance value (in joules/°C).
Heat_Load	Real	Range (0.0 <= Heat_Load <= 1.0)	Heat load fraction. The heat load fraction represents the portion of the heat that dissipates at the vertex.
Area	Real	Range (Area >= 0.0) Optional (There is no area associated with the interface.)	Area (in the parent part definition units^2). The area defines the convecting (or conducting or radiating) area associated with the external interface.
Properties[0..?]	NValue	NList (*User-defined*) Optional ()	Associated property values.
Entity Constraints: Informal (The sum of Heat_Load attribute values, one for each Thermal_CV entity contained within the thermal model, must equal 1.0.) Where_Used (Thermal_Model, Thermal_RV (*Ref. by ID*))			

Usage Notes:

- See the usage notes for the Thermal Model entity.
- The vertex type defines whether the vertex is internal to the thermal model or a connection point

- (interface) to the external environment. There are three types of external vertices that define connections to the board, component (such as a heat sink), or the environment. External vertices usually have an area associated with them. The vertex types are defined as follows:

Type	Description
Ex_Board	Vertex is externally connected to the board. The associated area defines the thermal interface to the board.
Ex_Component	Vertex is externally connected to another component (such as a heat sink). The associated area defines the thermal interface to the component.
Ex_Environment	Vertex is externally connected to the environment. The associated area defines the thermal interface to the environment.
Internal	Vertex is internal to the thermal model.
Junction	Vertex is on the component junction.
Unspecified	Vertex type is unkown or not required.

6.41 Thermal_RV

The Thermal_RV entity is used to represent a list of thermal resistance values, each of which is defined for a specific pair of vertices, defined by a Thermal_CV entity.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Res_Name	String	Optional (The resistance name is either not known, or it is not required for the translation.)	Resistance name.
Vtx_1	Ref_ID	RList (Thermal_CV)	Vertex one.
Vtx_2	Ref_ID	RList (Thermal_CV)	Vertex two.
Resistance	Real	Range (Resistance > 0.0)	Resistance value (in °C/Watts).
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints:			
Informal (Vtx_1 and VTx_2 must reference separate Thermal_CV entities, which are contained in the same Thermal_Model entity that the Thermal_RV entity is contained within.)			
Where_Used (Thermal_Model)			

Usage Notes:

- See the usage notes for the Thermal Model entity.

6.42 Trace

The Trace entity is used to represent a contiguous set of linear and/or circular arc conductor segments. In addition, a trace may optionally have an associated net name.

Attribute Name	Type	Constraints	Description
Entity_ID	Ent_ID	- - -	IDF entity ID.
Feature_ID	String	Default (“Unassigned”)	Unique feature identifier.
Lock	Ext_Enum	EList (“MCAD”, “ECAD”, “None”) Default (“None”)	Feature lock status.
Layer	Name	RList (Physical_Layer) Informal (Layer attribute must specify a physical layer name or a predefined layer alias.)	Board or panel layer on which the trace is located.
Curve	Entity	RList (Polycurve, Polyline)	Curve that represents the trace.
Net_Name	String	Optional (Net name is not required for the translation.)	Net name that is associated with the trace.
Properties[0..?]	NValue	NList (U*ser-defined*) Optional ()	Associated property values.
Entity Constraints: Informal (A trace can only be be on a single layer.) Where_Used (Board_Part, Figure, Footprint, Panel_Part)			

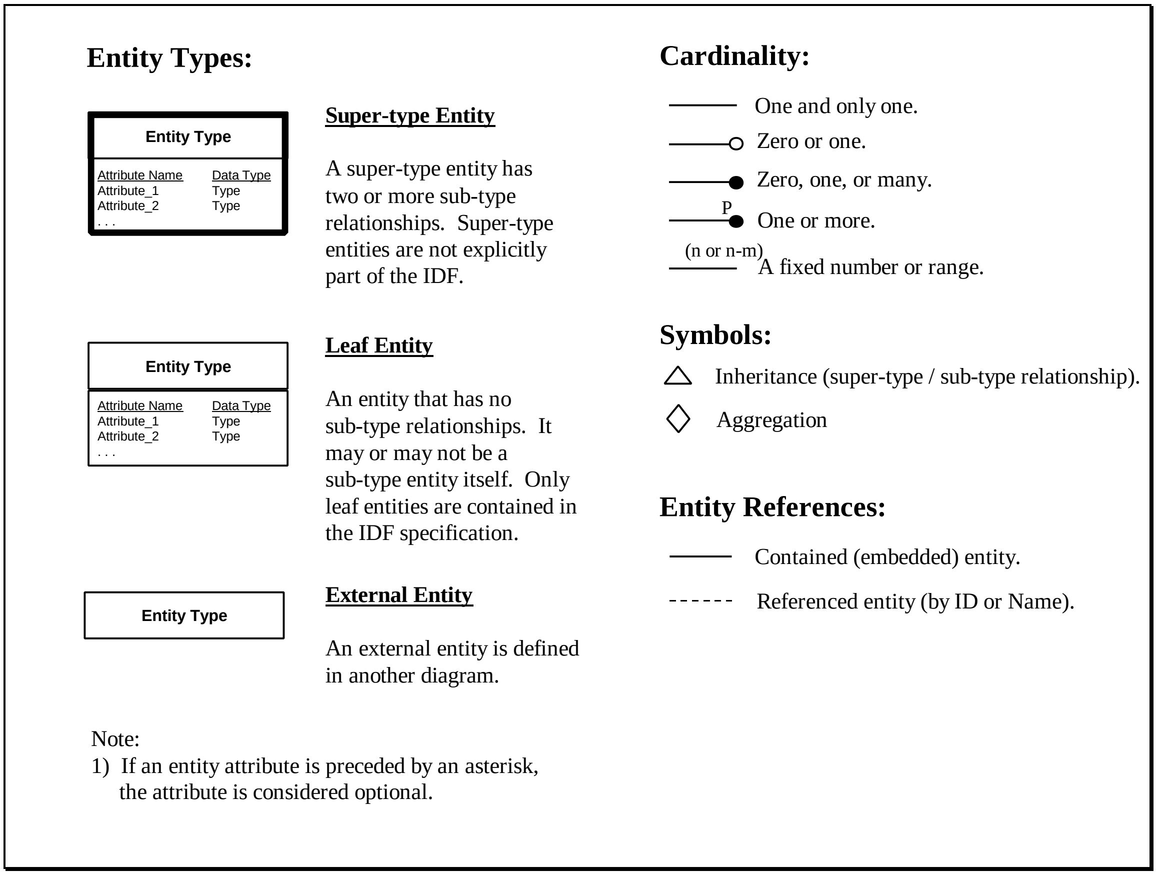
Usage Notes:

- The curve that represents the trace is defined in the coordinate system of the entity that contains the trace.
- It is not possible to represent a tapered curve using the Trace entity. A tapered trace should either be represented with a Filled_Area entity or the tapered aspect should simply be ignored.
- It is not possible to represent complex corner conditions such as chamfers using the Trace entity. A trace with complex corner conditions must be represented with a Filled_Area entity.

Appendix A Model Diagrams

Appendix A represents a collection of IDF entity diagrams using a variant of the Unified Modeling Language (UML) notation. The purpose of this appendix is to provide a broad view of the overall entity content and relationships among the entities.

The following diagram illustrates the notation used for the entity diagrams:



Super-type Entity

A super-type entity has two or more sub-type relationships. Super-type entities are not explicitly part of the IDF.

Leaf Entity

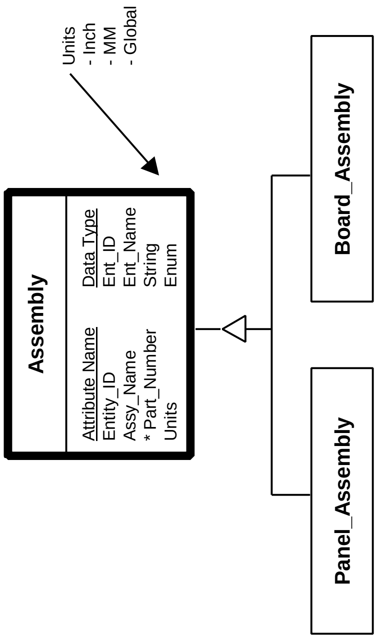
An entity that has no sub-type relationships. It may or may not be a sub-type entity itself. Only leaf entities are contained in the IDF specification.

External Entity

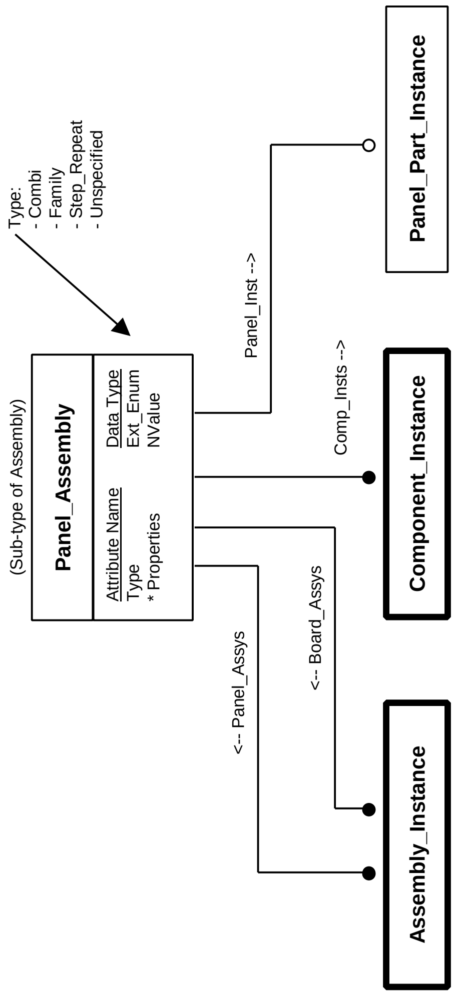
An external entity is defined in another diagram.

Note:
1) If an entity attribute is preceded by an asterisk, the attribute is considered optional.

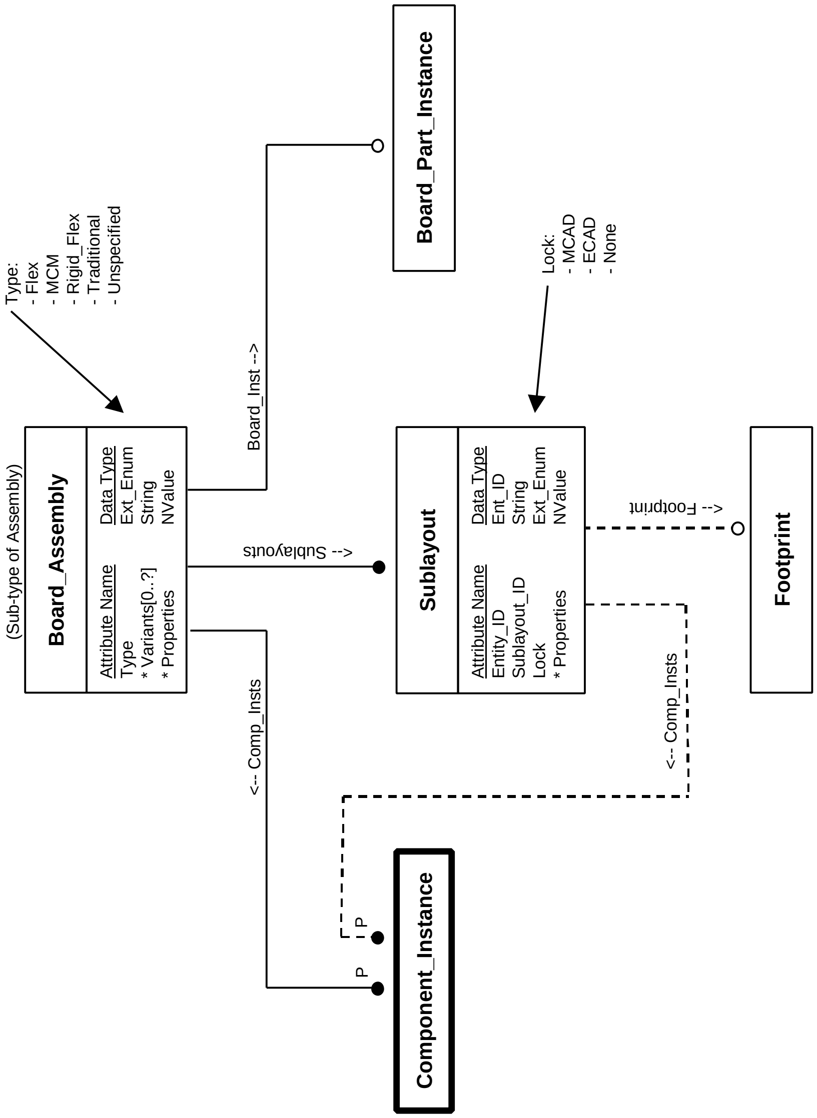
A.1 Assembly


Units
- Inch
- MM
- Global
Assembly
Attribute Name
Entity_ID
Assy_Name
* Part_Number
Units
Data Type
Ent_ID
Ent_Name
String
Enum
Board_Assembly
Panel_Assembly

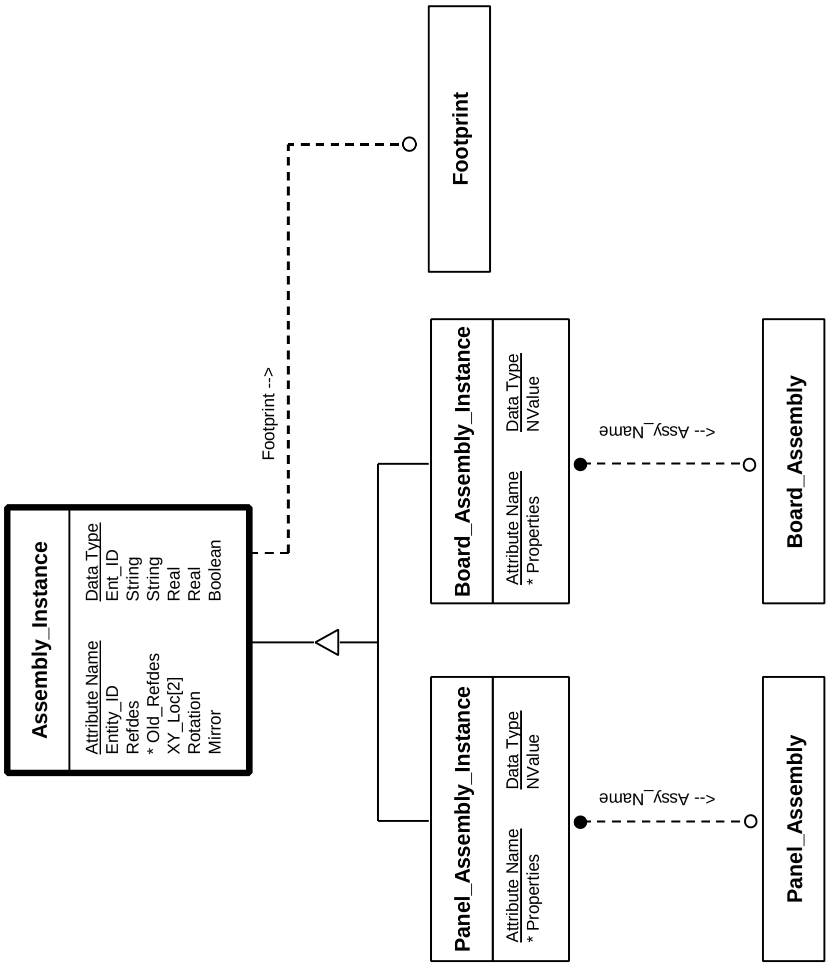
A.2 Panel Assembly



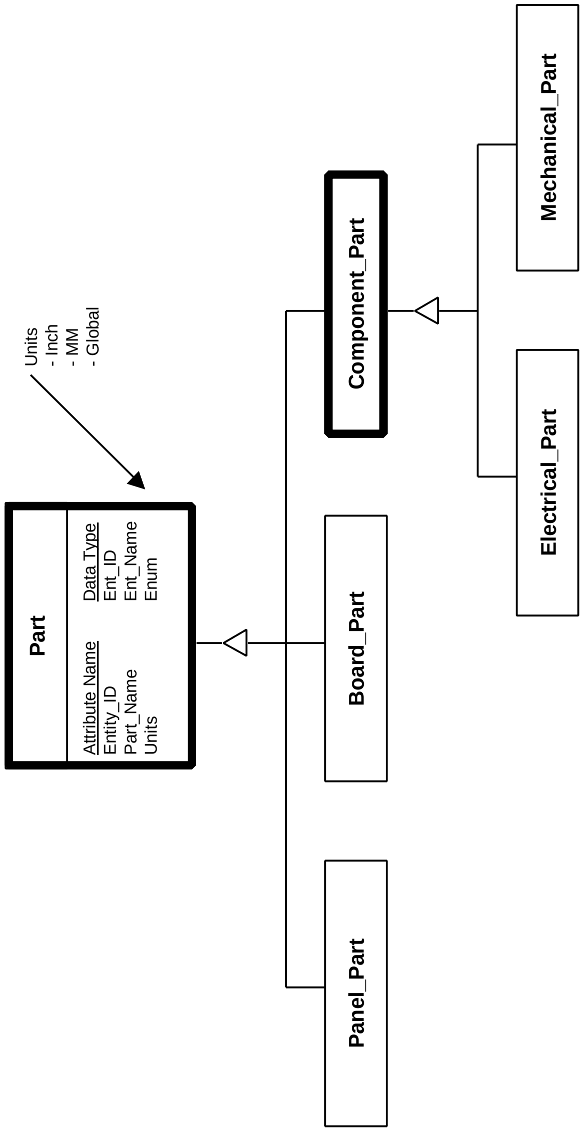
A.3 Board Assembly



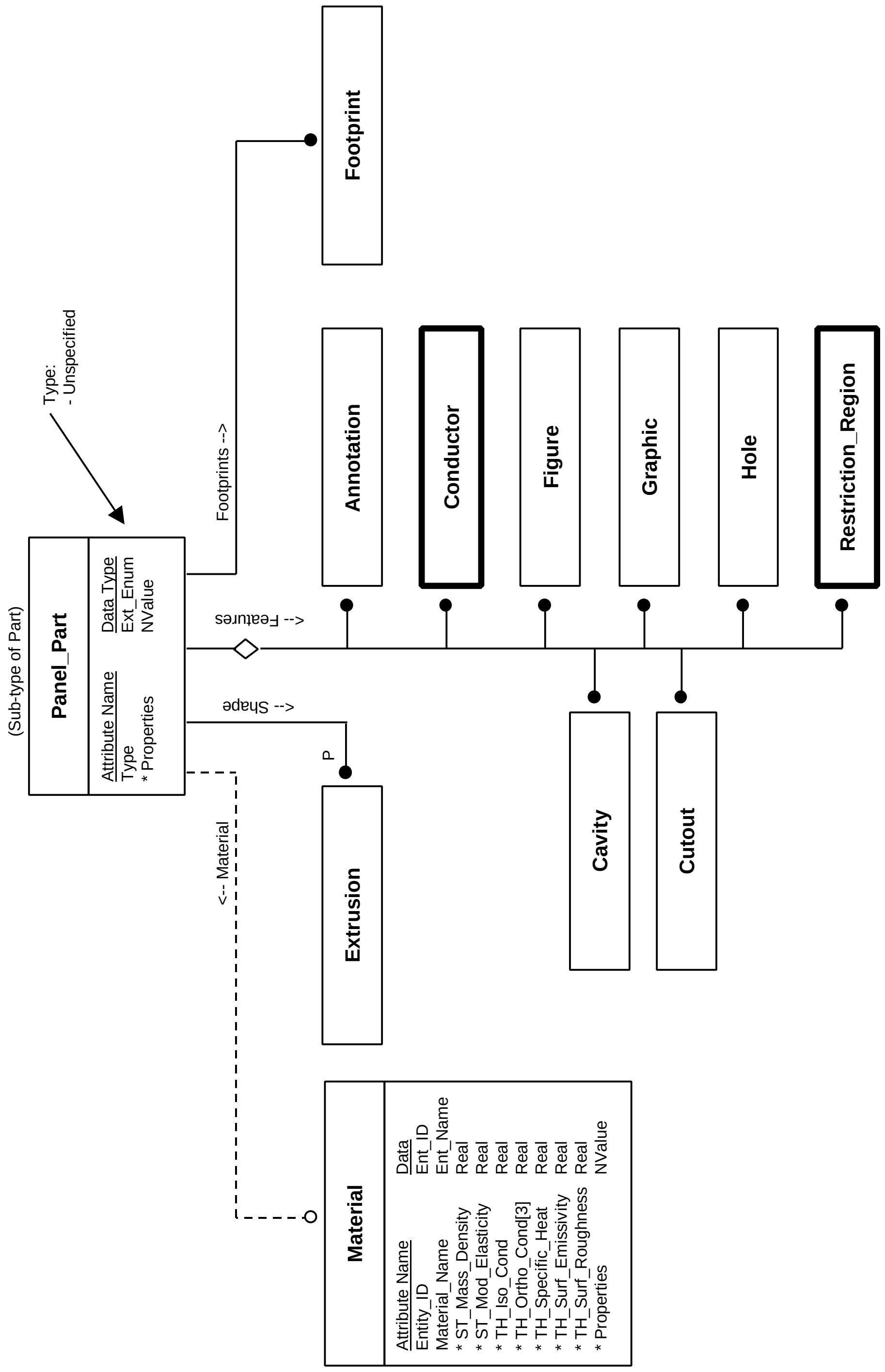
A.4 Assembly Instance



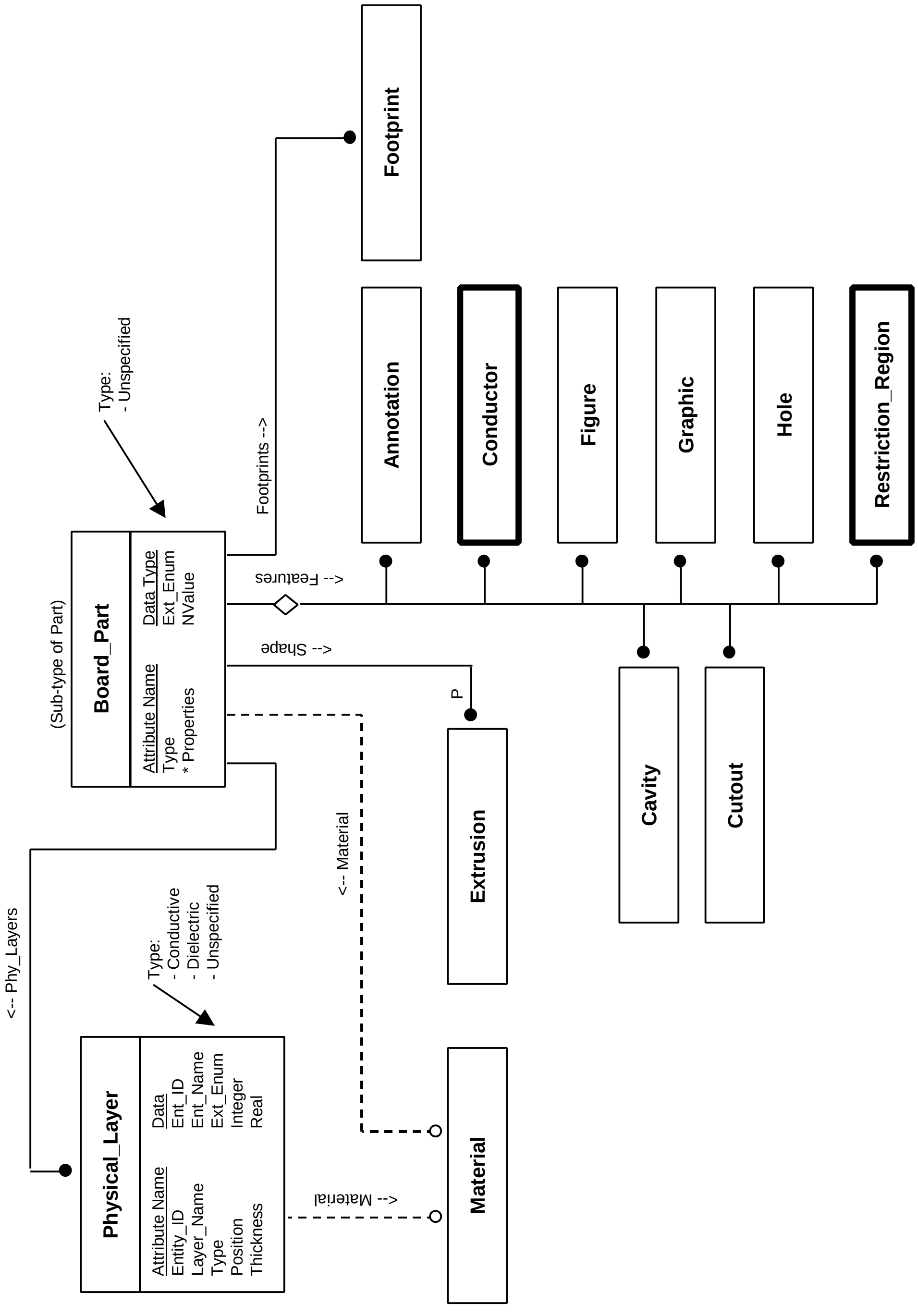
A.5 Part


Units
- Inch
- MM
- Global
Part
Attribute Name
Entity_ID
Part_Name
Units
Data Type
Ent_ID
Ent_Name
Enum
Component_Part
Board_Part
Panel_Part
Mechanical_Part
Electrical_Part

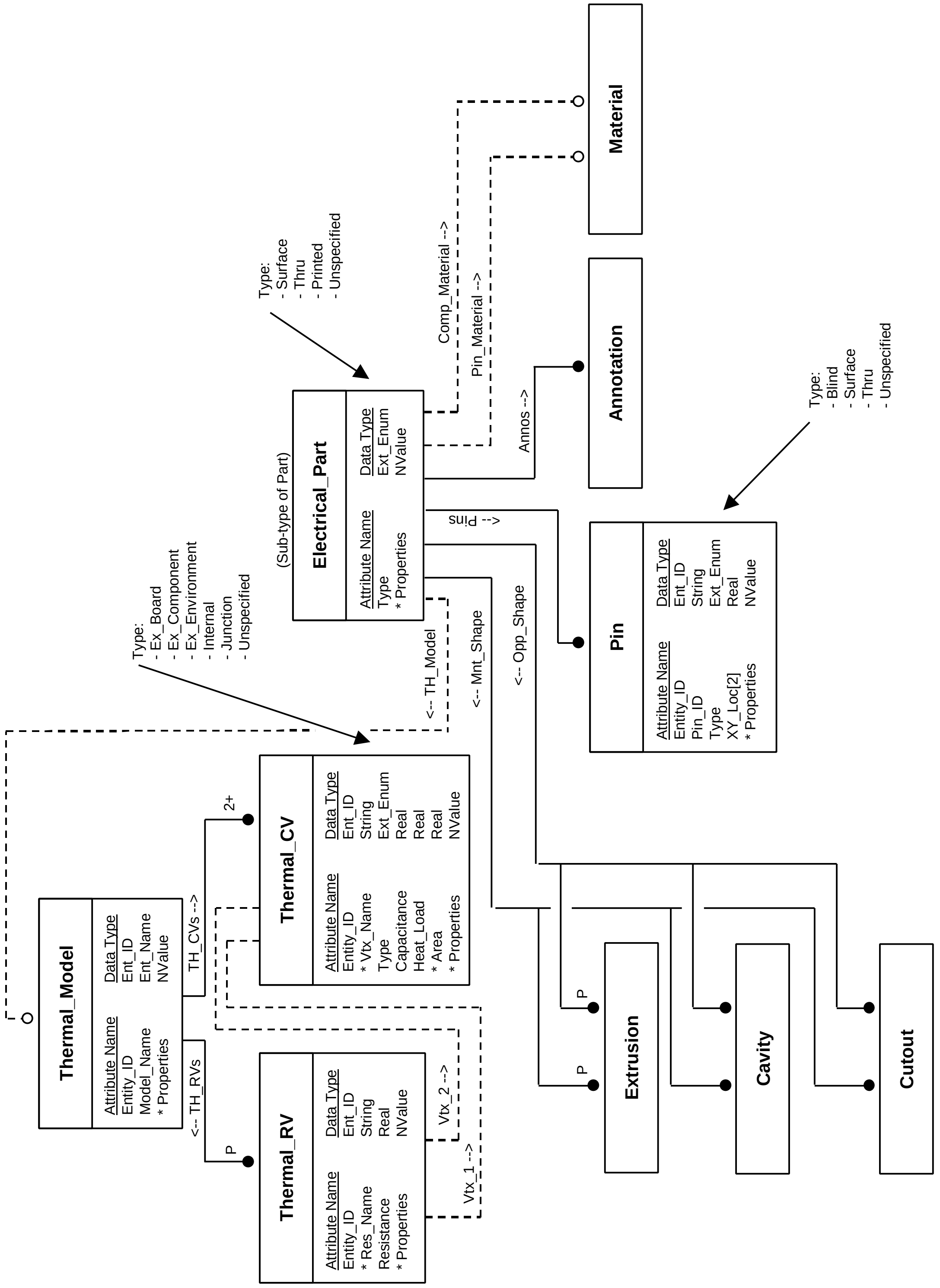
A.6 Panel Part



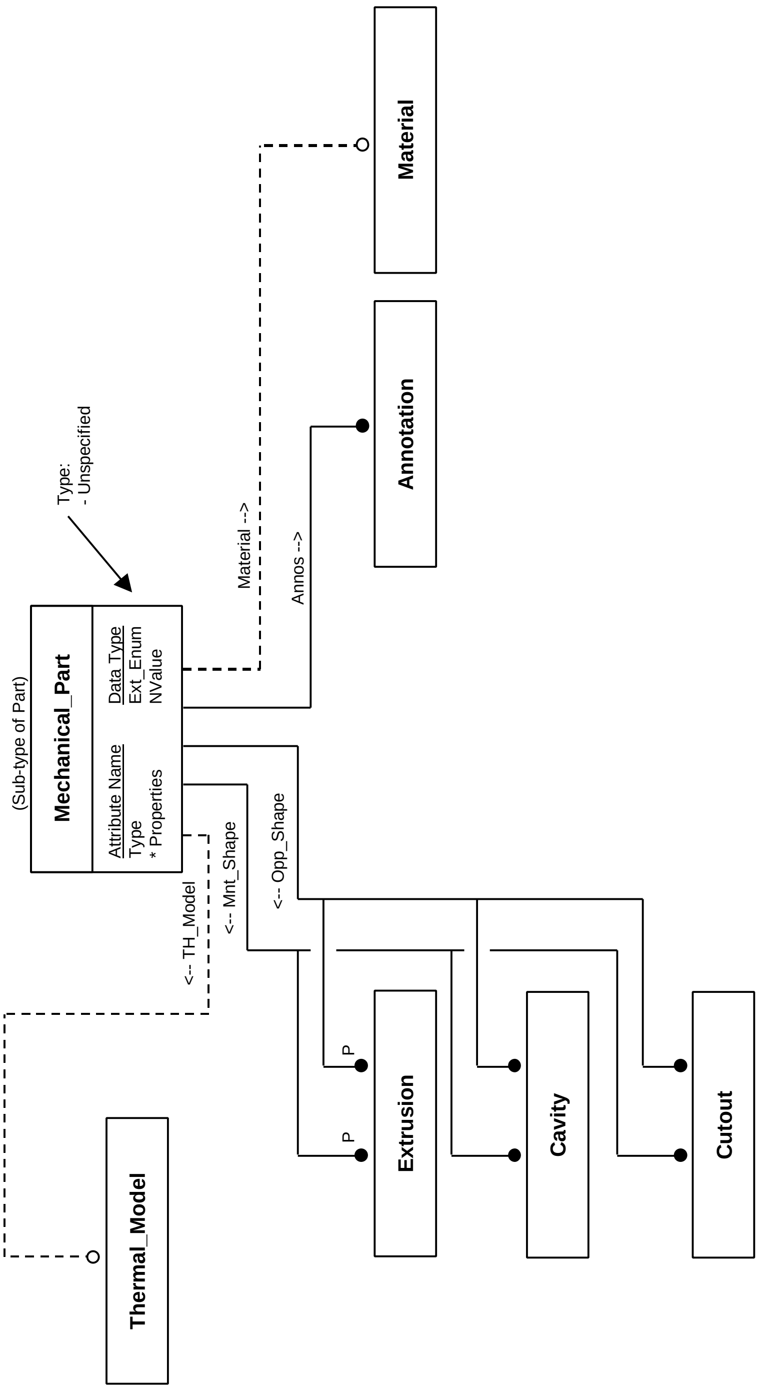
A.7 Board Part



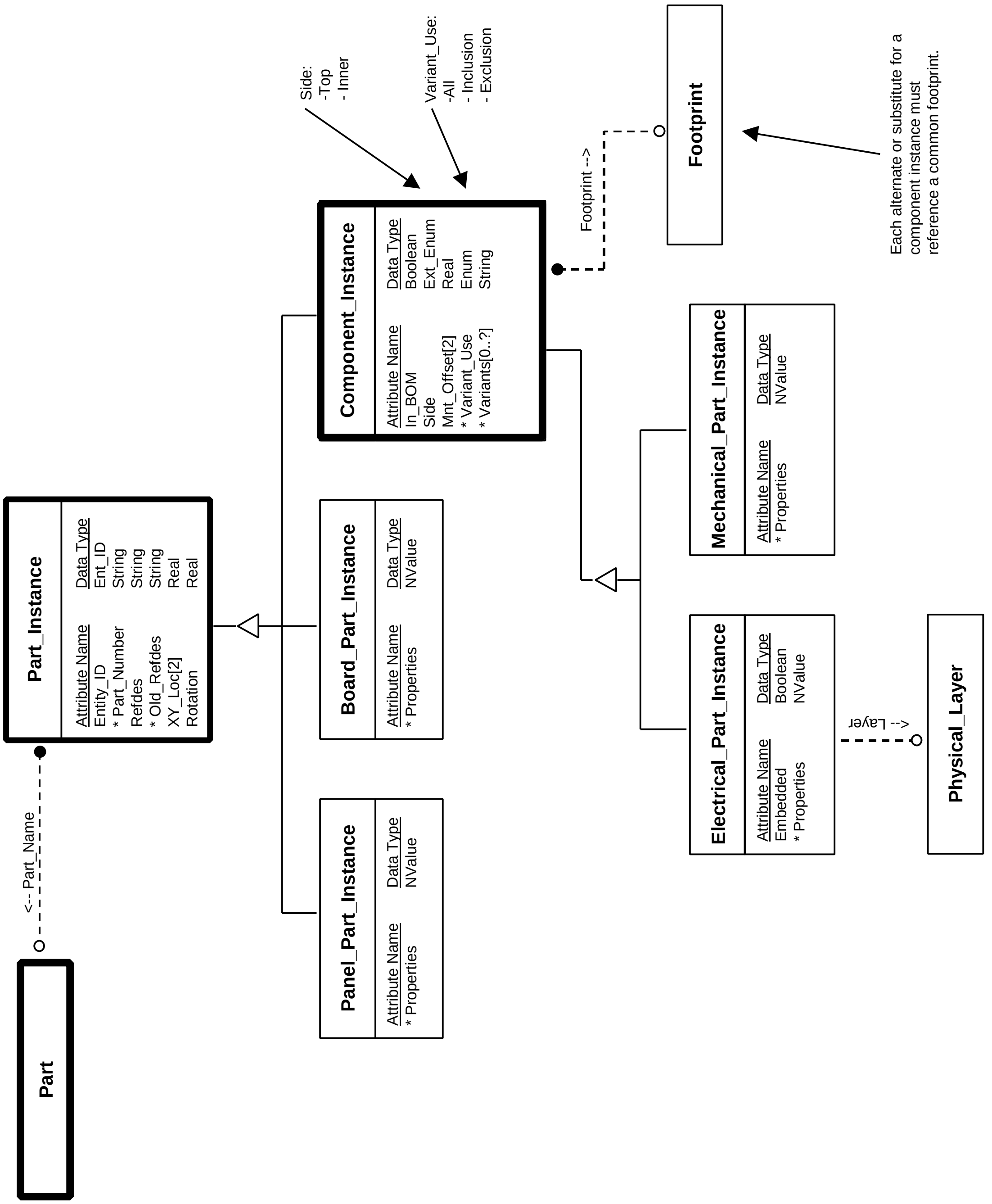
A.8 Electrical Part



A.9 Mechanical Part



A.10 Part Instance



Part_Instance

Attribute Name	Data Type
Entity_ID	Ent_ID
* Part_Number	String
Refdes	String
* Old_Refdes	String
XY_Loc[2]	Real
Rotation	Real

Component_Instance

Attribute Name	Data Type
In_BOM	Boolean
Side	Ext_Enum
Mnt_Offset[2]	Real
* Variant_Use	Enum
* Variants[0..?]	String

Board_Part_Instance

Attribute Name	Data Type
* Properties	NValue

Panel_Part_Instance

Attribute Name	Data Type
* Properties	NValue

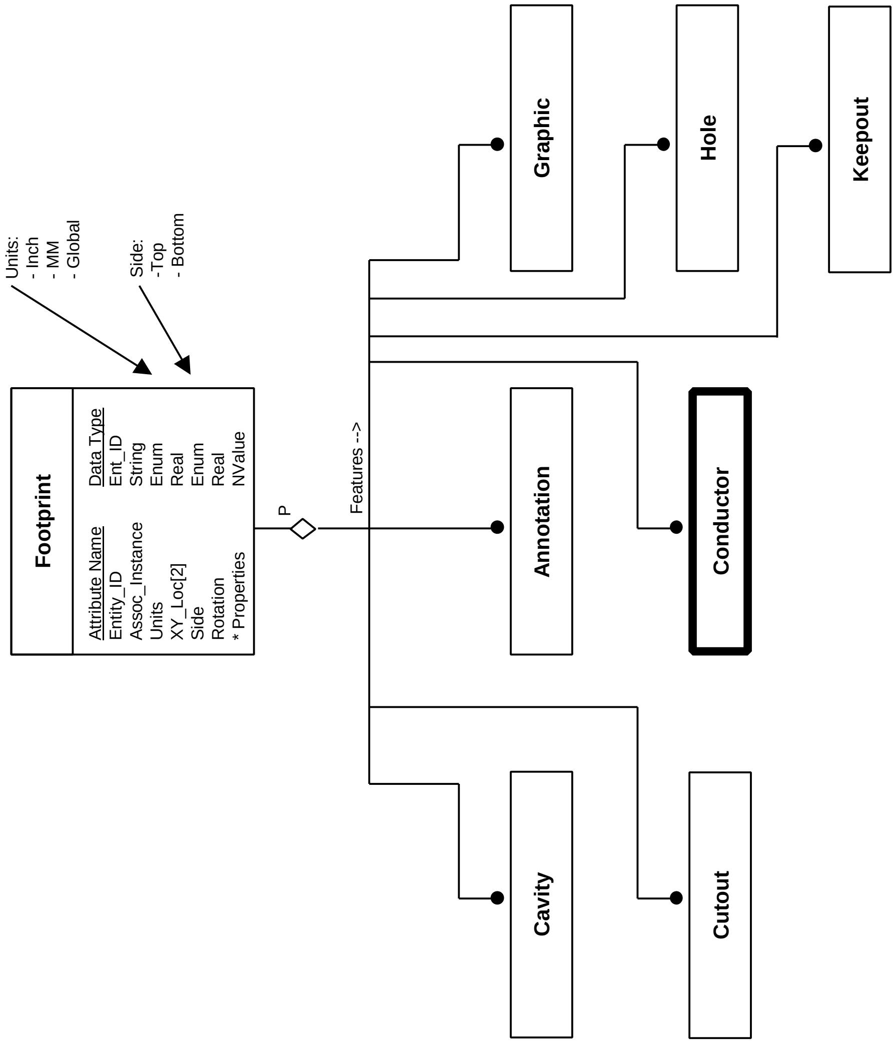
Mechanical_Part_Instance

Attribute Name	Data Type
* Properties	NValue

Electrical_Part_Instance

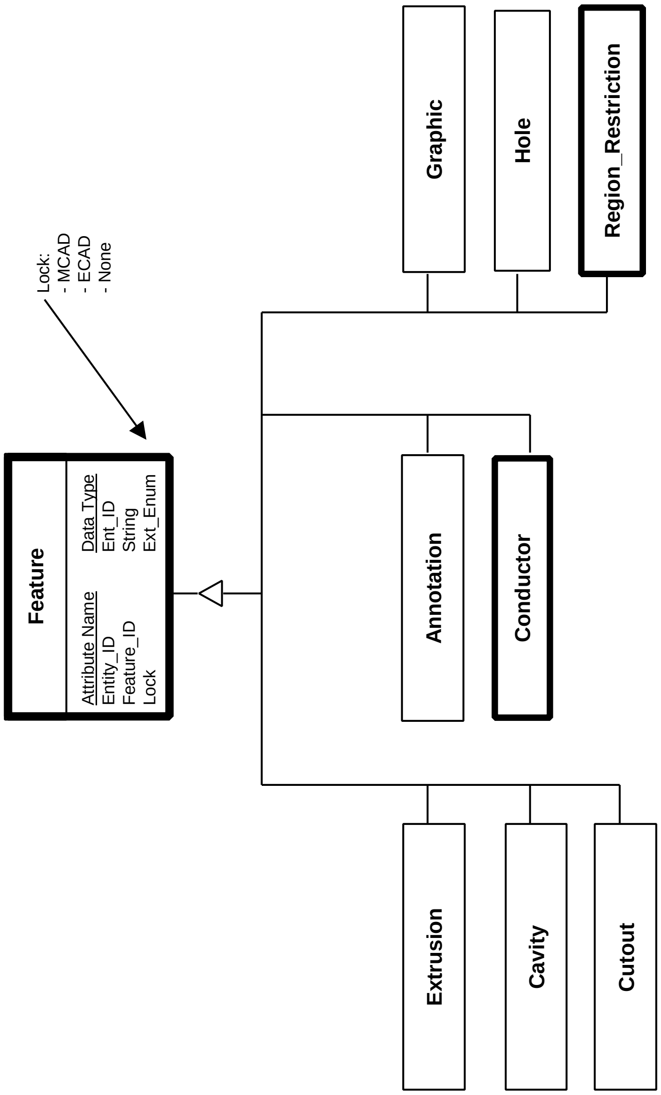
Attribute Name	Data Type
Embedded	Boolean
* Properties	NValue

A.11 Footprint

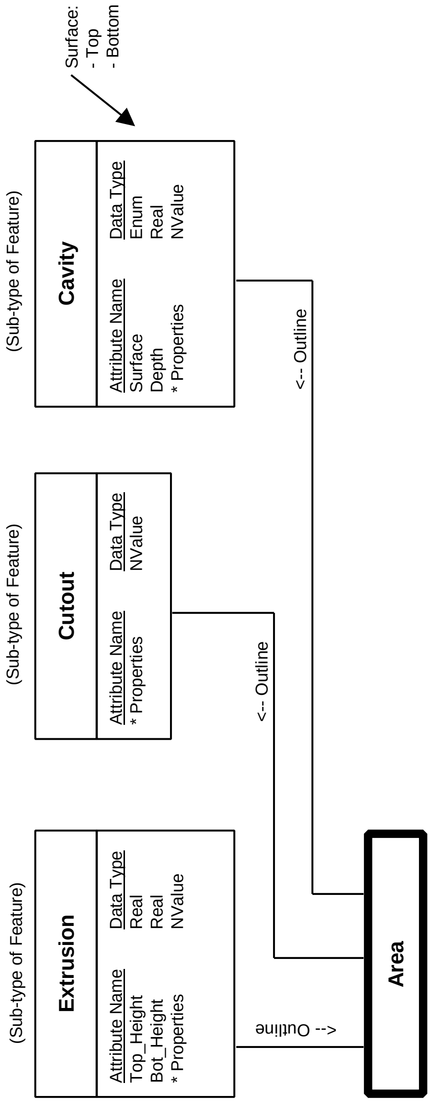


Attribute Name	Data Type
Entity_ID	Ent_ID
Assoc_Instance	String
Units	Enum
XY_Loc[2]	Real
Side	Enum
Rotation	Real
* Properties	NValue

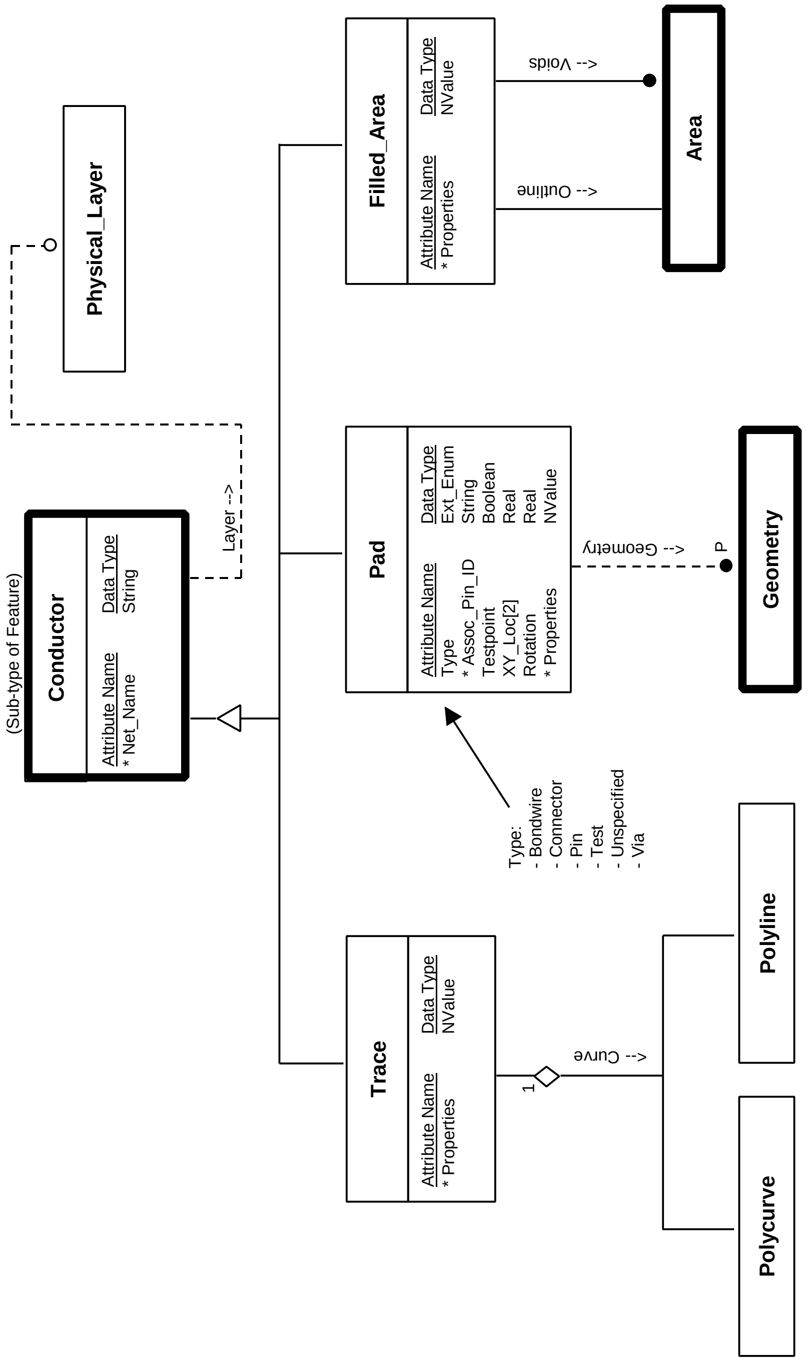
A.12 Feature


Feature
Attribute Name
Entity_ID
Feature_ID
Lock
Data Type
Ent_ID
String
Ext_Enum
Lock:
- MCAD
- ECAD
- None
Graphic
Hole
Region_Restriction
Annotation
Conductor
Extrusion
Cavity
Cutout

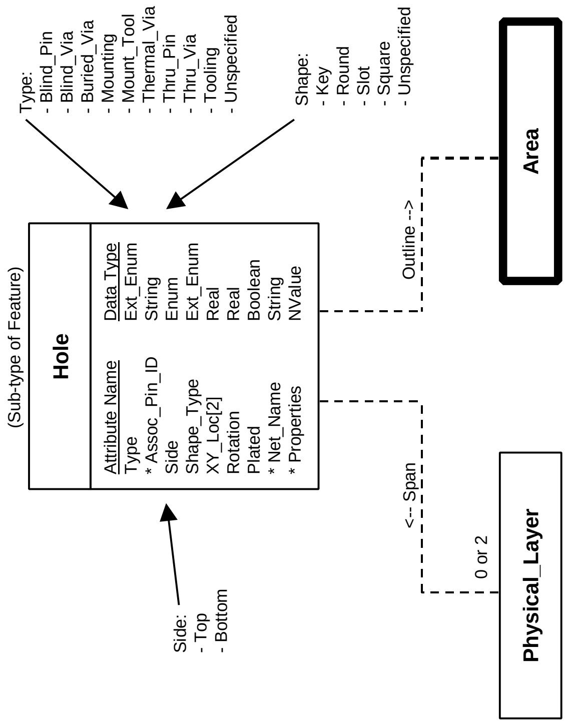
A.13 Shape


Surface:
- Top
- Bottom
(Sub-type of Feature)
Cavity
Attribute Name
Surface
Depth
* Properties
Data Type
Enum
Real
NValue
<-- Outline
(Sub-type of Feature)
Cutout
Attribute Name
* Properties
Data Type
NValue
<-- Outline
(Sub-type of Feature)
Extrusion
Attribute Name
Top_Height
Bot_Height
* Properties
Data Type
Real
Real
NValue
<-- Outline
Area

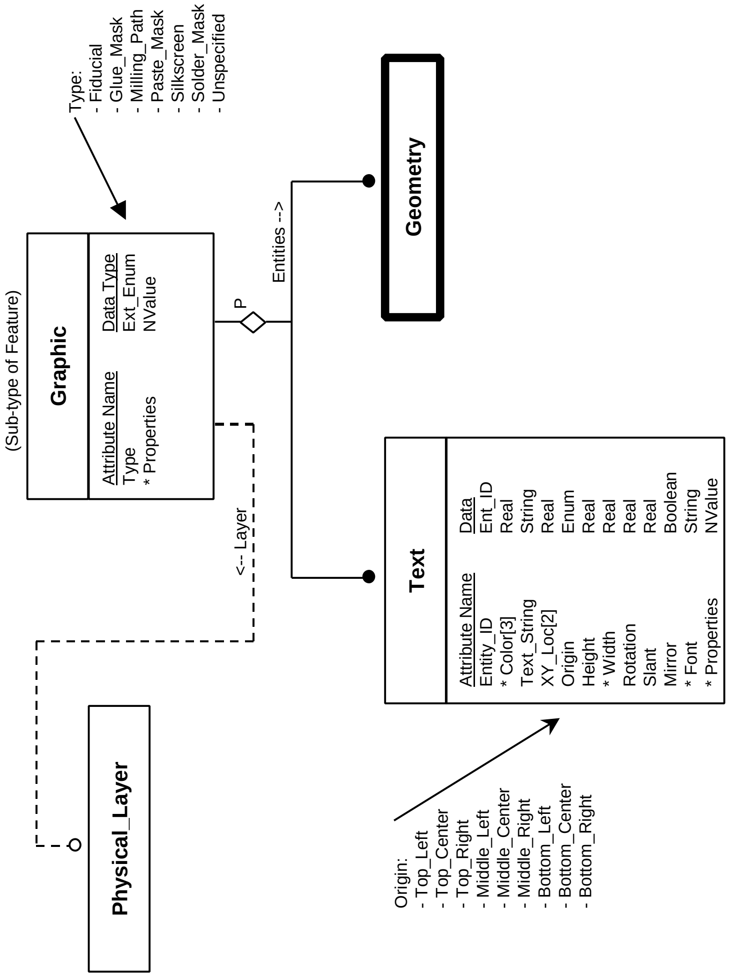
A.14 Conductor



A.15 Hole


(Sub-type of Feature)
Hole
Attribute Name
Type
* Assoc_Pin_ID
Side
Shape_Type
XY_Loc[2]
Rotation
Plated
* Net_Name
* Properties
Data Type
Ext_Enum
String
Enum
Ext_Enum
Real
Real
Boolean
String
NValue
Type:
- Blind_Pin
- Blind_Via
- Buried_Via
- Mounting
- Mount_Tool
- Thermal_Via
- Thru_Pin
- Thru_Via
- Tooling
- Unspecified
Shape:
- Key
- Round
- Slot
- Square
- Unspecified
Side:
- Top
- Bottom
Outline -->
Area
<-- Span
0 or 2
Physical_Layer

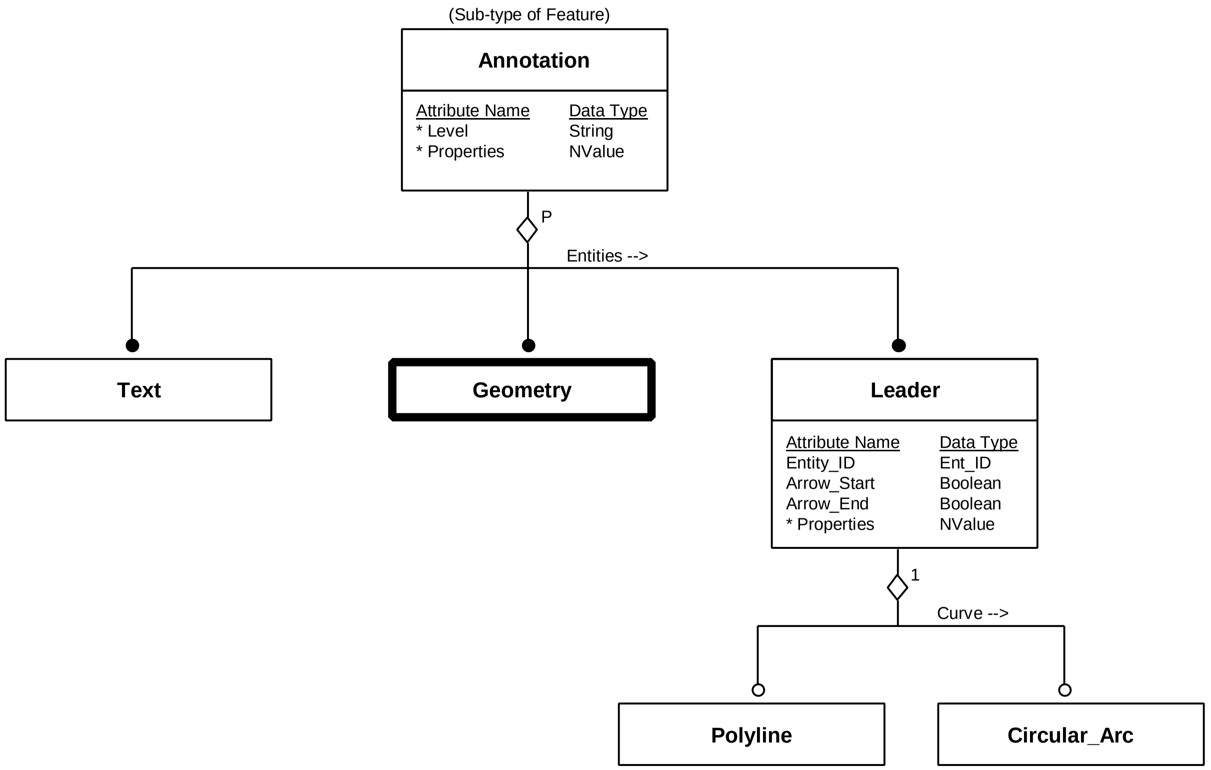
A.16 Graphic



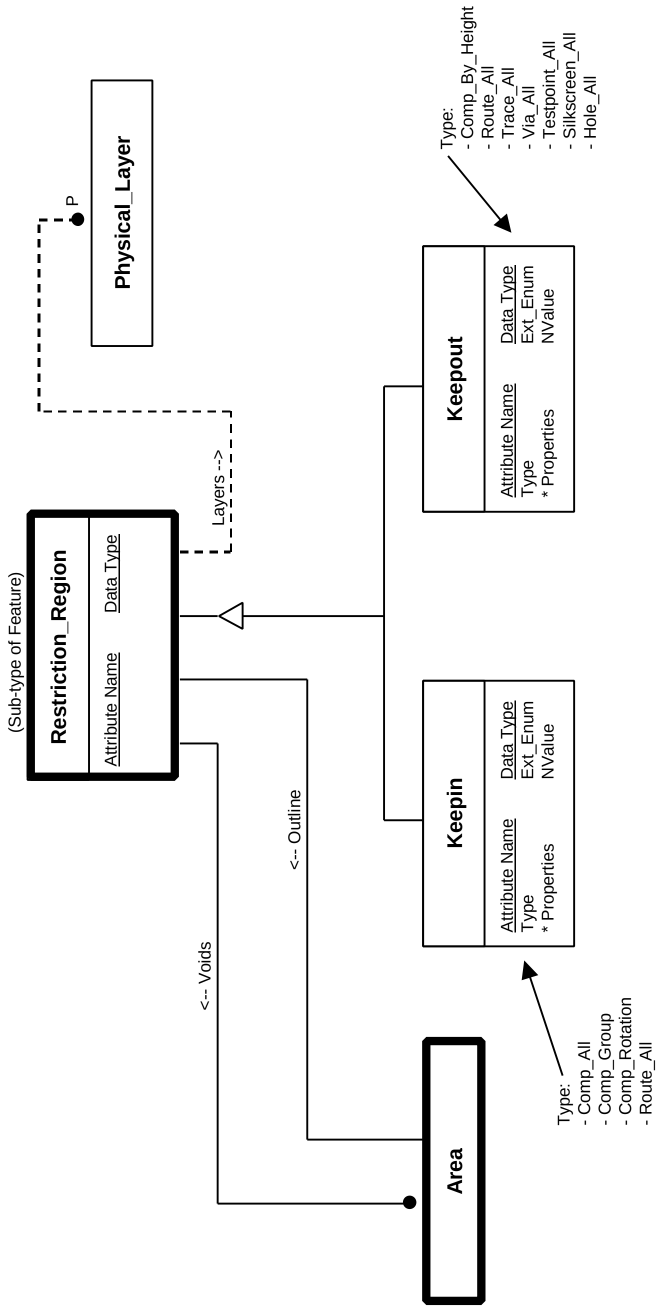
Attribute Name	Data Type
Type	Ext_Enum
* Properties	NValue

Attribute Name	Data
Entity_ID	Ent_ID
* Color[3]	Real
Text_String	String
XY_Loc[2]	Real
Origin	Enum
Height	Real
* Width	Real
Rotation	Real
Slant	Real
Mirror	Boolean
* Font	String
* Properties	NValue

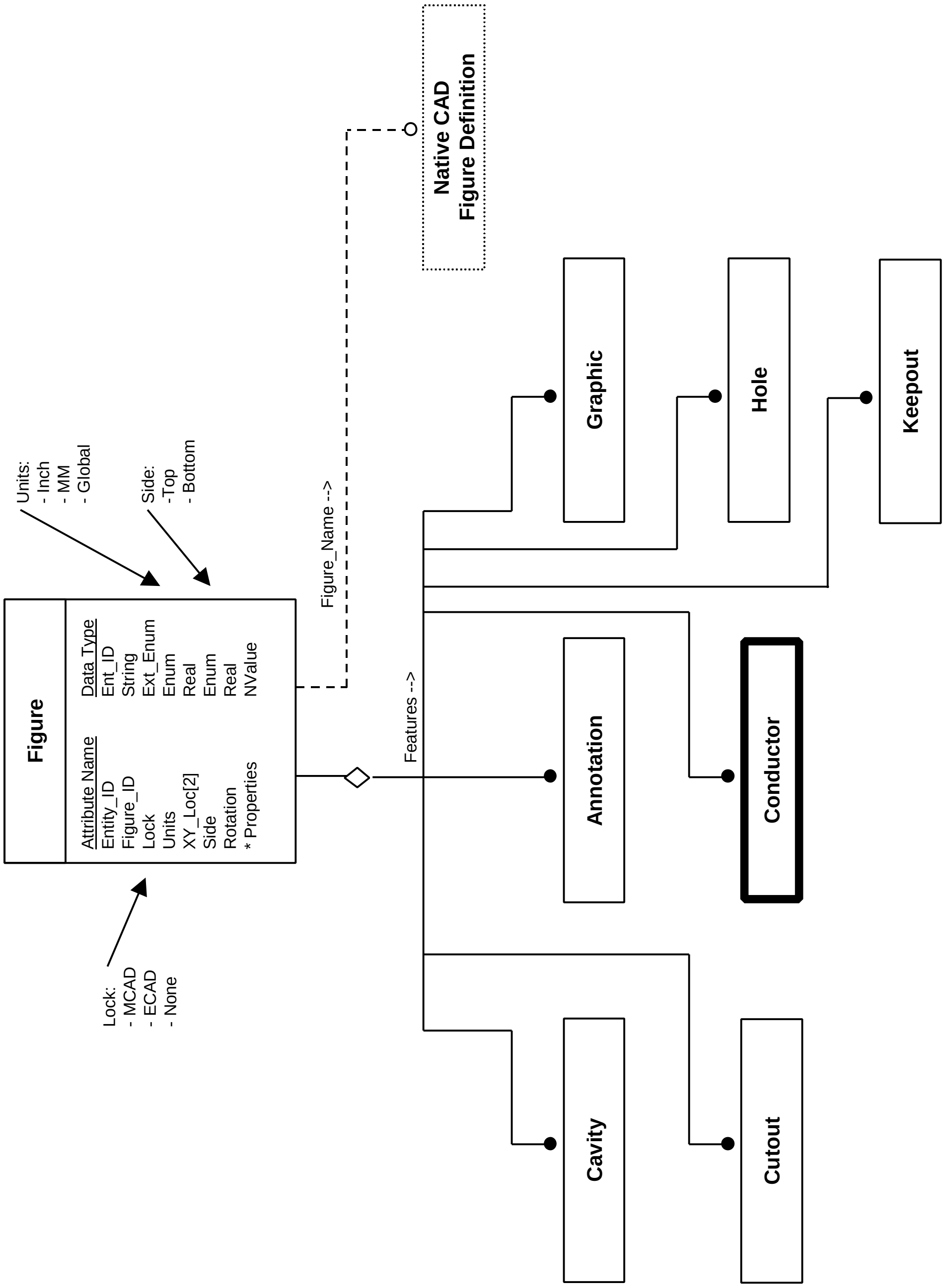
A.17 Annotation



A.18 Keepin and Keepout

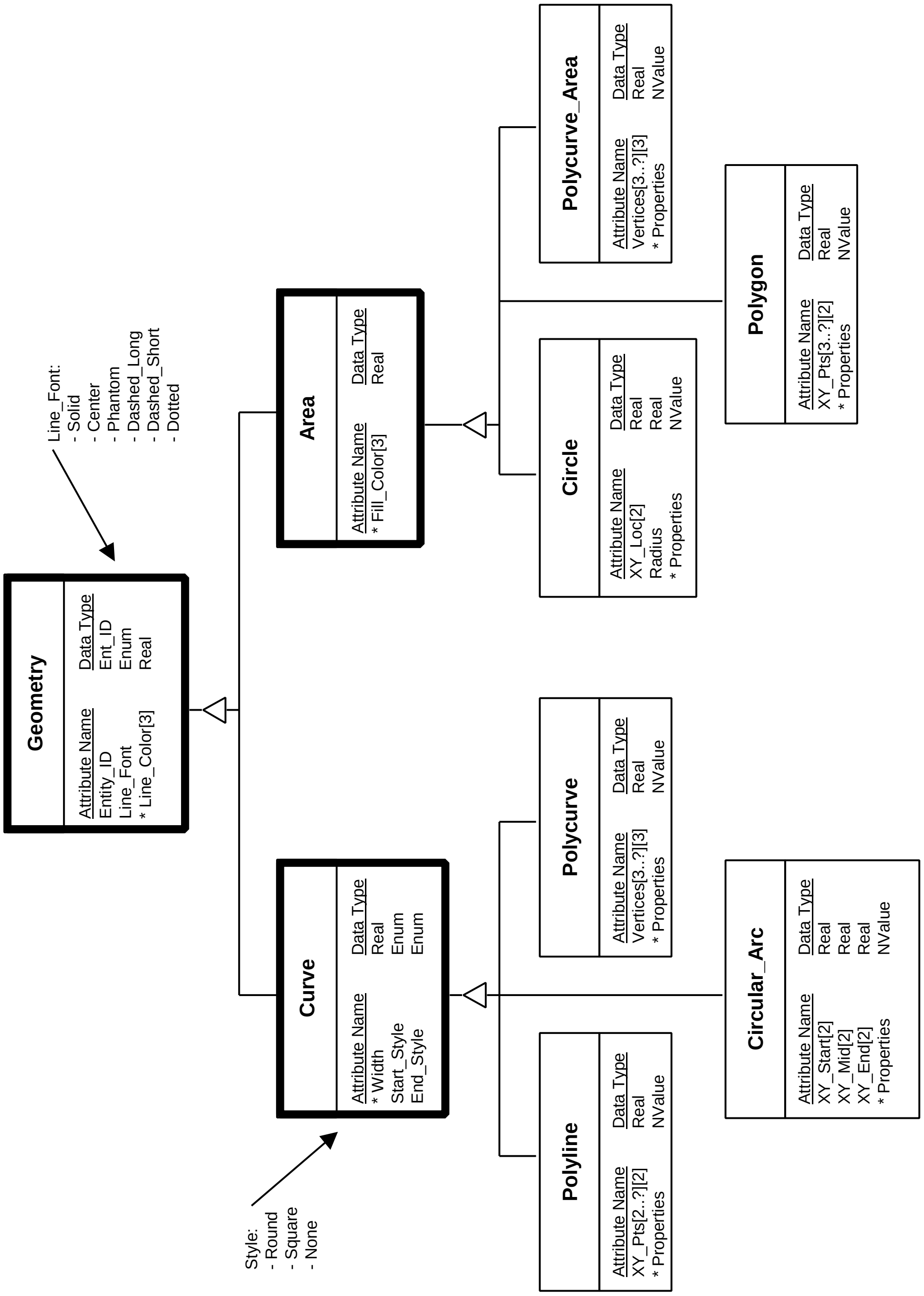


A.19 Figure



Attribute Name	Data Type
Entity_ID	Ent_ID
Figure_ID	String
Lock	Ext_Enum
Units	Enum
XY_Loc[2]	Real
Side	Enum
Rotation	Real
* Properties	NValue

A.20 Geometry



Appendix B Sample File

This section represents a sample PCA design, encoded in a set of IDF 4.0 files. It is intended to provide a base for how entities are represented and related in the physical file. There are five files that represent the PCA design, one file for each content category:

- Component Parts
- Board Part
- Board Assembly
- Panel Part
- Panel Assembly

Please note that the files have been hand-generated. Consequently, there may be minor syntax and structure errors.

B.1 Component Parts

This section represents a sample file for the following component part definitions.

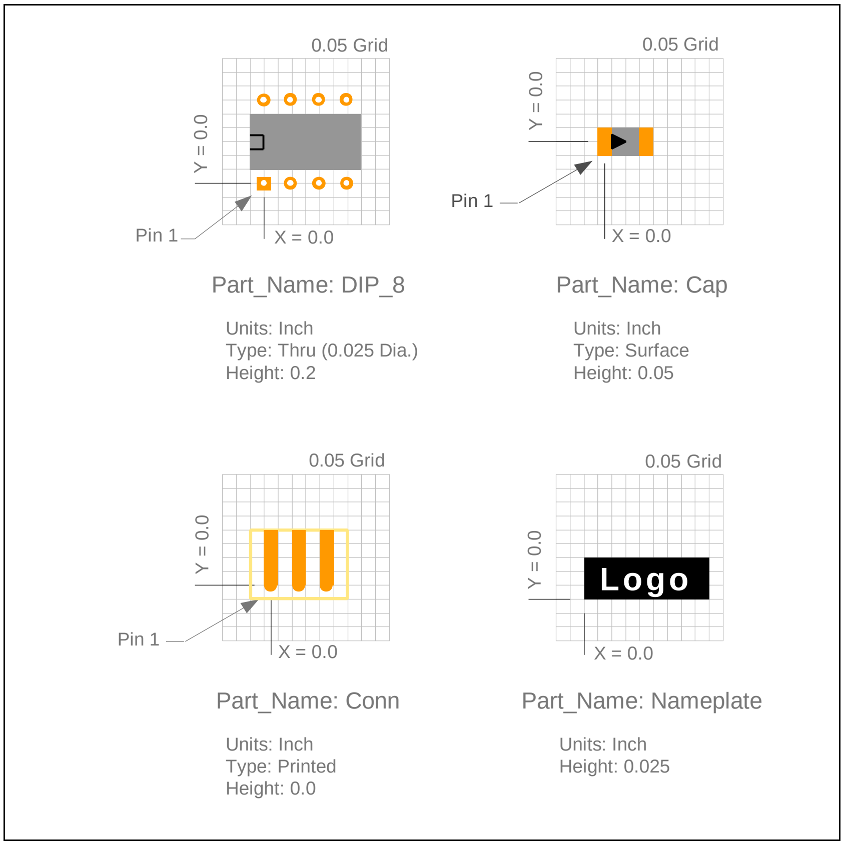


Figure B.1 Component Part Definitions

```
/*************************************************************************/
/* Component Part Library                                                */
/*************************************************************************/

IDF_Header (

  Version (“4.0”),
  Creation_Date_Time (“1998/06/05.10:00:00”),
  Owner_Name (“Tom Makoski”),
  Owner_Phone (“513-528-5059”),
  Owner_EMail (“Tom.Makoski@Intermedius.com”),
  Source_App_Type (“Hand_Generated”),
  Source_App_Vendor (“N/A”),
  Source_App_Name (“N/A”),
  Source_App_Version (“N/A”),
  IDF_Tx_Name (“N/A”),
  IDF_Tx_Version (“N/A”),
  Entity_Count (
    Elec_Part_Defs (3),
    Elec_Part_Insts (0),
    Mech_Part_Defs (1),
    Mech_Part_Insts (0),
    Board_Part_Defs (0),
    Board_Part_Insts (0),
    Board_Assy_Defs (0),
    Board_Assy_Insts (0),
    Panel_Part_Defs (0),
    Panel_Part_Insts (0),
    Panel_Assy_Defs (0),
    Panel_Assy_Insts (0)
  ),
  Comp_Part (
    “Annotation”,
    “Electrical_Part”,
    “Extrusion”,
    “Mechanical_Part”,
    “Pin”,
    “Polygon”,
    “Polyline”,
    “Text
  ),
  Default_Units (“Inch”),
  Min_Res (0.0005),
  Notes (
    “This IDF file represents a component part library.”
  )


); /* End IDF_Header Section */

/*************************************************************************/
/* Parts Section                                                         */
/*************************************************************************/

Parts (
                             /********************************************/
                             /* DIP_8 Part Definition                    */
                             /********************************************/
  Electrical_Part (
    Entity_ID (#1001),
    Part_Name (“DIP_8”),
    Units (“Global”),
    Type (“Thru”),
    Mnt_Shape (                                   /* Mounting Side Shape */
      Extrusion (
        Entity_ID (#1002),
        Top_Height (0.2),
        Bot_Height (0.0),
        Outline (
          Polygon (
            Entity_ID (#1003),
            Line_Color (59.0, 59.0, 59.0), /* Medium Grey */
            Fill_Color (59.0, 59.0, 59.0), /* Medium Grey */
            XY_Pts (
              -0.050, 0.050,
               0.350, 0.050,
               0.350, 0.250,
              -0.050, 0.250,
              -0.050, 0.050
            )
          ); /* End Polygon */
        )
      ); /* End Extrusion */
    ),
    Pins (                                                       /* Pins */
      Pin
        Entity_ID (#1004),
        Pin_ID (“1”),
        Type (“Thru”),
        XY_Loc (0.0, 0.0)
      ); /* End Pin */
      Pin
        Entity_ID (#1005),
        Pin_ID (“2”),
        Type (“Thru”),
        XY_Loc (0.1, 0.0)
      ); /* End Pin */
      Pin
        Entity_ID (#1006),
        Pin_ID (“3”),
        Type (“Thru”),
        XY_Loc (0.2, 0.0)
      ); /* End Pin */
      Pin
        Entity_ID (#1007),
        Pin_ID (“4”),
        Type (“Thru”),
        XY_Loc (0.3, 0.0)
      ); /* End Pin */
      Pin
        Entity_ID (#1008),
        Pin_ID (“5”),
        Type (“Thru”),
        XY_Loc (0.3, 0.3)
      ); /* End Pin */
      Pin
        Entity_ID (#1009),
        Pin_ID (“6”),
        Type (“Thru”),
        XY_Loc (0.2, 0.3)
      ); /* End Pin */
      Pin
        Entity_ID (#1010),
        Pin_ID (“7”),
        Type (“Thru”),
        XY_Loc (0.1, 0.3)
      ); /* End Pin */
      Pin
        Entity_ID (#1011),
        Pin_ID (“8”),
        Type (“Thru”),
        XY_Loc (0.0, 0.3)
      ); /* End Pin */
    ),
    Annos (                                                /* Annotations */
      Annotation (
        Entity_ID (#1012),
        Level (“Comp_Body_Graphics”),
        Entities (
          Polyline (
            Entity_ID (#1013),
            Line_Color (0.0, 0.0, 0.0), /* Black */
            Width (0.008),
            XY_Pts (
              -0.05, 0.125,
               0.00, 0.125,
               0.00, 0.175,
              -0.05, 0.175
            )
          ); /* End Polyline */
        )
      ); /* End Annotation */
    ),
    Properties (
      EL_Power_Nominal (0.1),
      EL_Tolerance (0.05)
    )
  ); /* End Electrical Part */
                             /********************************************/
                             /* Cap Part Definition                      */
                             /********************************************/
  Electrical_Part (
    Entity_ID (#1014),
    Part_Name (“Cap”),
    Units (“Global”),
    Type (“Surface”),
    Mnt_Shape (                                    /* Mounting Side Shape */
      Extrusion (
        Entity_ID (#1015),
        Top_Height (0.05),
        Bot_Height (0.0),
        Outline (
          Polygon (
            Entity_ID (#1016),
            Line_Color (59.0, 59.0, 59.0), /* Medium Grey */
            Fill_Color (59.0, 59.0, 59.0), /* Medium Grey */
            XY_Pts (
              0.025, -0.050,
              0.125, -0.050,
              0.125,  0.050,
              0.025,  0.050,
              0.025, -0.050
            )
          ); /* End Polygon */
        )
      ); /* End Extrusion */
    ),
    Pins (                                                       /* Pins */
      Pin
        Entity_ID (#1017),
        Pin_ID (“1”),
        Type (“Surface”),
        XY_Loc (0.0, 0.0)
      ); /* End Pin */
      Pin
        Entity_ID (#1018),
        Pin_ID (“2”),
        Type (“Surface”),
        XY_Loc (0.15, 0.0)
      ); /* End Pin */
    ),
    Annos (                                               /* Annotations */
      Annotation (
        Entity_ID (#1019),
        Level (“Comp_Body_Graphics”),
        Entities (
          Polygon (
            Entity_ID (#1020),
            Line_Color (0.0, 0.0, 0.0), /* Black */
            Fill_Color (0.0, 0.0, 0.0), /* Black */
            XY_Pts (
              0.025, -0.025,
              0.075,  0.000,
              0.025,  0.025,
              0.025, -0.025
            )
          ); /* End Polygon */
        )
      ); /* End Annotation */
    ),
    Properties (
      EL_Capacitance (0.1),
      EL_Tolerance (0.05)
    )
  ); /* End Electrical Part */
                             /********************************************/
                             /* Conn Part Definition                     */
                             /********************************************/
  Electrical_Part (
    Entity_ID (#1021),
    Part_Name (“Conn”),
    Units (“Global”),
    Type (“Printed”),
    Mnt_Shape (                                   /* Mounting Side Shape */
      Extrusion (
        Entity_ID (#1022),
        Top_Height (0.0),
        Bot_Height (0.0),
        Outline (
          Polygon (
            Entity_ID (#1023),
            Line_Color (100.0, 100.0, 0.0), /* Dark Yellow */
            Fill_Color (100.0, 100.0, 0.0), /* Dark Yellow */
            XY_Pts (
              -0.075, -0.050,
               0.275, -0.050,
               0.275,  0.200,
              -0.075,  0.200,
              -0.075, -0.050
            )
          ); /* End Polygon */
        )
      ); /* End Extrusion */
    ),
    Opp_Shape (                                   /* Opposite Side Shape */
      Extrusion (
        Entity_ID (#1024),
        Top_Height (0.0),
        Bot_Height (0.0),
        Outline (
          Polygon (
            Entity_ID (#1025),
            Line_Color (100.0, 100.0, 0.0), /* Dark Yellow */
            Fill_Color (100.0, 100.0, 0.0), /* Dark Yellow */
            XY_Pts (
              -0.075, -0.050,
               0.275, -0.050,
               0.275,  0.200,
              -0.075,  0.200,
              -0.075, -0.050
            )
          ); /* End Polygon */
        )
      ); /* End Extrusion */
    ),
    Pins (                                                       /* Pins */
      Pin
        Entity_ID (#1026),
        Pin_ID (“1”),
        Type (“Surface”),
        XY_Loc (0.0, 0.0)
      ); /* End Pin */
      Pin
        Entity_ID (#1027),
        Pin_ID (“2”),
        Type (“Surface”),
        XY_Loc (0.1, 0.0)
      ); /* End Pin */
      Pin
        Entity_ID (#1028),
        Pin_ID (“3”),
        Type (“Surface”),
        XY_Loc (0.2, 0.0)
      ); /* End Pin */
    ),
  ); /* End Electrical Part */
                             /********************************************/
                             /* Nameplate Part Definition                */
                             /********************************************/
  Mechanical_Part (
    Entity_ID (#1029),
    Part_Name (“Nameplate”),
    Units (“Global”),
    Type (“Unspecified”),
    Mnt_Shape (                                   /* Mounting Side Shape */
      Extrusion (
        Entity_ID (#1030),
        Top_Height (0.025),
        Bot_Height (0.0),
        Outline (
          Polygon (
            Entity_ID (#1031),
            Line_Color (0.0, 0.0, 0.0), /* Black */
            Fill_Color (0.0, 0.0, 0.0), /* Black */
            XY_Pts (
              0.00, 0.00,
              0.45, 0.00,
              0.45, 0.15,
              0.00, 0.15,
              0.00, 0.00
            )
          ); /* End Polygon */
        )
      ); /* End Extrusion */
    ),
    Annos (                                                   /* Annotations */
      Annotation (                         /* Visual aid for Nameplate text. */
        Entity_ID (#1032),
        Level (“Comp_Body_Graphics”),
        Entities (
          Text (
            Entity_ID (#1033),
            Color (100.0, 100.0, 100.0), /* White */
            Text_String (“Logo”),
            XY_Loc (0.05, 0.025),
            Height (0.1),
            Width (0.35),
            Rotation (0.0),
            Font (“Copperplate Gothic Light”)
          ); /* End Text */
        )
      ); /* End Annotation */
    )
  ); /* End Mechanical Part */


); /* End Parts Section */
```

B.2 Board Part

This section represents a sample file for the following board part definition.

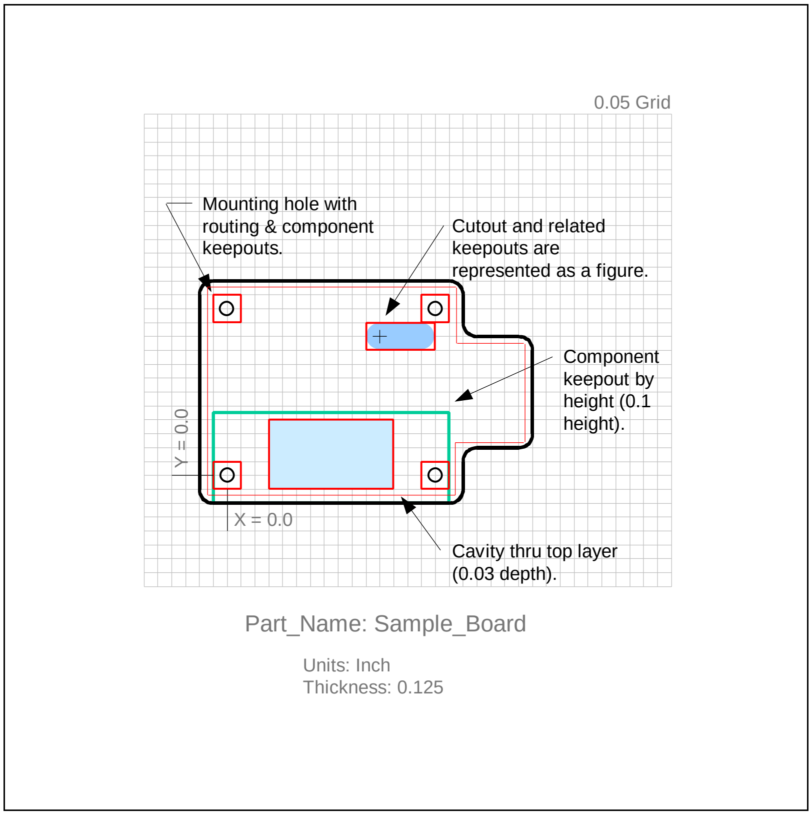

0.05 Grid
Mounting hole with routing & component keepouts.
Cutout and related keepouts are represented as a figure.
Component keepout by height (0.1 height).
Y = 0.0
X = 0.0
Cavity thru top layer (0.03 depth).
Part_Name: Sample_Board
Units: Inch
Thickness: 0.125

Figure B.2 Board Part Definition

```
/**************************************************************************/
/* Board Part                                                             */
/**************************************************************************/

IDF_Header (

  Version (“4.0”),
  Creation_Date_Time (“1998/06/05.10:00:00”),
  Owner_Name (“Tom Makoski”),
  Owner_Phone (“513-528-5059”),
  Owner_EMail (“Tom.Makoski@Intermedius.com”),
  Source_App_Type (“Hand_Generated”),
  Source_App_Vendor (“N/A”),
  Source_App_Name (“N/A”),
  Source_App_Version (“N/A”),
  IDF_Tx_Name (“N/A”),
  IDF_Tx_Version (“N/A”),
  Entity_Count (
    Elec_Part_Defs (0),
    Elec_Part_Insts (0),
    Mech_Part_Defs (0),
    Mech_Part_Insts (0),
    Board_Part_Defs (1),
    Board_Part_Insts (0),
    Board_Assy_Defs (0),
    Board_Assy_Insts (0),
    Panel_Part_Defs (0),
    Panel_Part_Insts (0),
    Panel_Assy_Defs (0),
    Panel_Assy_Insts (0)
  ),
  Board_Part (
    “Annotation”,
    “Board_Part”,
    “Cavity”,
    “Circle”,
    “Cutout”,
    “Extrusion”,
    “Figure”,
    “Filled_Area”,
    “Footprint”,
    “Graphic”,
    “Hole”,
    “Keepin”,
    “Keepout”,
    “Leader”,
    “Pad”,
    “Physical_Layer”,
    “Polycurve_Area”,
    “Polygon”,
    “Polyline”,
    “Text”,
    “Trace”
  ),
  Default_Units (“Inch”),
  Min_Res (0.0005),
  Notes (
    “This file represents a board part. The board has a cutout “,
    “(which is represented as a figure), a cavity (which is part “,
    “of a sublayout), and four mounting holes.  In addition, it ”,
    “contains the footprints for the component part instances and ”,
    “sublayout that are contributed via the board assembly.”
  )


); /* End IDF_Header Section */

/*************************************************************************/
/* Parts Section                                                         */
/*************************************************************************/

Parts (
                             /********************************************/
                             /* Sample Board Part Definition             */
                             /********************************************/
  Board_Part (
    Entity_ID (#2001),
    Part_Name (“Sample_Board”),
    Units (“Global”),
    Type (“Unspecified”),
                             /********************************************/
                             /* Board Shape                              */
                             /********************************************/
    Shape (
      Extrusion (
        Entity_ID (#2002),
        Top_Height (0.125),
        Bot_Height (0.0),
        Outline (
          Polycurve_Area (
            Entity_ID (#2003),
            Line_Color (0.0, 0.0, 0.0), /* Black */
            Fill_Color (100.0, 100.0, 100.0), /* White */
            Vertices (
              -0.05, -0.10,   0.0,
               0.80, -0.10,   0.0,
               0.85, -0.05,  90.0,
               0.85,  0.05,   0.0,
               0.90,  0.10, -90.0,
               1.05,  0.10,   0.0,
               1.10,  0.15,  90.0,
               1.10,  0.45,   0.0,
               1.05,  0.50,  90.0,
               0.90,  0.50,   0.0,
              0.85,  0.55, -90.0,
              0.85,  0.65,   0.0,
              0.80,  0.70,  90.0,
             -0.05,  0.70,   0.0,
             -0.10,  0.65,  90.0,
             -0.10, -0.05,   0.0,
             -0.05, -0.10,  90.0
            )
          ); /* End Polycurve_Area */
        )
      ); /* End Extrusion */
    ),
                             /********************************************/
                             /* Physical Layer Stackup                   */
                             /********************************************/
    Phy_Layers (
      Physical_Layer (                                /* Physical Layers */
        Entity_ID (#2501),
        Layer_Name (“Conductor_Top”),
        Type (“Conductive”),
        Position (1),
        Thickness (0.00)
      ); /* End Physical Layer */
      Physical_Layer (
        Entity_ID (#2502),
        Layer_Name (“Silkscreen_Top”),
        Type (“Silkscreen”),
        Position (2),
        Thickness (0.00)
      ); /* End Physical Layer */
      Physical_Layer (
        Entity_ID (#2503),
        Layer_Name (“Conductor_Inner_1”),
        Type (“Conductive”),
        Position (3),
        Thickness (0.00)
      ); /* End Physical Layer */
      Physical_Layer (
        Entity_ID (#2504),
        Layer_Name (“Conductor_inner_2”),
        Type (“Conductive”),
        Position (4),
        Thickness (0.00)
      ); /* End Physical Layer */
      Physical_Layer (
        Entity_ID (#2505),
        Layer_Name (“Silkscreen_Bottom”),
        Type (“Silkscreen”),
        Position (5),
        Thickness (0.00)
      ); /* End Physical Layer */
      Physical_Layer (
        Entity_ID (#2506),
        Layer_Name (“Conductor_Bottom”),
        Type (“Conductive”),
        Position (6),
        Thickness (0.00)
      ); /* End Physical Layer */
    ),
                             /********************************************/
                             /* Features                                 */
                             /********************************************/
    Features (
                             /********************************************/
                             /* Keepins                                  */
                             /********************************************/

      Keepin (                                       /* Component Keepin */
        Entity (#2004),
        Type (“Comp_All”),
        Layer (“Conductor_Top”, “Conductor_Bottom”),
        Outline (
          Polygon (
            Entity_ID (#2005),
            Line_Color (100.0, 0.0, 0.0), /* Medium Red */
            Fill_Color (100.0, 0.0, 0.0), /* Medium Red */
            XY_Pts (
              -0.075, -0.075,
               0.825, -0.075,
               0.825,  0.125,
               1.075,  0.125,
               1.075,  0.475,
               0.825,  0.475,
               0.825,  0.675,
              -0.075,  0.675,
              -0.075, -0.075
            )
          ); /* End Polygon */
        ),
        Properties (KI_Comp_Height (0.4))
      ); /* End Keepin */
      Keepin (                                           /* Route Keepin */
        Entity (#2006),
        Type (“Route_All”),
        Layer (“All”),
        Outline (
          Polygon (
            Entity_ID (#2007),
            Line_Color (100.0, 0.0, 0.0), /* Medium Red */
            Fill_Color (100.0, 0.0, 0.0), /* Medium Red */
            XY_Pts (
              -0.075, -0.075,
               0.825, -0.075,
               0.825,  0.125,
               1.075,  0.125,
               1.075,  0.475,
               0.825,  0.475,
               0.825,  0.675,
              -0.075,  0.675,
              -0.075, -0.075
            )
          ); /* End Polygon */
        )
      ); /* End Keepin */
                             /********************************************/
                             /* Component Keepouts                       */
                             /********************************************/
      Keepout (
        Entity (#2008),
        Type (“Comp_By_Height”),
        Layer (“Conductor_Top”),
        Outline (
          Polygon (
            Entity_ID (#2009),
            Line_Color (0.0, 100.0, 0.0), /* Medium Green */
            Fill_Color (0.0, 100.0, 0.0), /* Medium Green */
            XY_Pts (
              -0.05, -0.10,
               0.80, -0.10,
               0.80,  0.225,
              -0.05,  0.225,
              -0.05, -0.10
            )
          ); /* End Polygon */
        ),
        Properties (KO_Comp_Height (0.1))
      ); /* End Keepout */
                             /********************************************/
                             /* Figures                                  */
                             /********************************************/

      Figure (                /* Figure with cutout and related keepouts */
        Entity_ID (#2010),
        Figure_Name (“Slot_15X10”),
        Units (“Global”)
        XY_Loc (0.55, 0.50),
        Side (“Top”),
        Rotation (0.0),
        Features (
          Cutout (
            Entity (#2011),
            Outline (
              Polycurve_Area (
                Entity_ID (#2012),
                Line_Color (80.0, 93.0, 100.0), /* Light Blue */
                Fill_Color (80.0, 93.0, 100.0), /* Light Blue */
                Vertices (
                  0.00, -0.05,   0.0,
                  0.15, -0.05,   0.0,
                  0.15,  0.05, 180.0,
                  0.00,  0.05,   0.0,
                  0.00, -0.05, 180.0
                )
              ); /* End Polycurve_Area */
            )
          ); /* End Cutout */
          Keepout (
            Entity (#2013),
            Type (“Comp_By_Height”),
            Layer (“Conductor_Top”, “Conductor_Bottom”),
            Outline (
              Polygon (
                Entity_ID (#2014),
                Line_Color (100.0, 0.0, 0.0), /* Medium Red */
                Fill_Color (100.0, 0.0, 0.0), /* Medium Red */
                XY_Pts (
                  -0.05, -0.05,
                   0.20, -0.05,
                   0.20,  0.05,
                  -0.05,  0.05,
                  -0.05, -0.05
                )
              ); /* End Polygon */
            ),
            Properties (KO_Comp_Height (0.0))
          ); /* End Keepout */
          Keepout (
            Entity (#2015),
            Type (“Route_All”),
            Layer (“All”),
            Outline (
              Polygon (
                Entity_ID (#2016),
                Line_Color (100.0, 0.0, 0.0), /* Medium Red */
                Fill_Color (100.0, 0.0, 0.0), /* Medium Red */
                XY_Pts (
                  -0.05, -0.05,
                   0.20, -0.05,
                   0.20,  0.05,
                  -0.05,  0.05,
                  -0.05, -0.05
                )
              ); /* End Polygon */
            )
          ); /* End Keepout */
        )
      ); /* End Figure */
                             /********************************************/
                             /* Mounting Holes and Related Keepouts      */
```

```
                           /********************************************/

    Hole (                                /* Mounting and Tooling Hole */
      Entity_ID (#2017),
      Type (“Mount_Tool”),
      Shape_Type (“Round”),
      Outline (#2521),
      XY_Loc (0.0, 0.0),
      Rotation (0.0)
    ); /* End Hole */
    Keepout (                                     /* Component Keepout */
      Entity (#2018),
      Type (“Comp_By_Height”),
      Layer (“Conductor_Top”, “Conductor_Bottom”),
      Outline (
        Polygon (
          Entity_ID (#2019),
          Line_Color (100.0, 0.0, 0.0), /* Medium Red */
          Fill_Color (100.0, 0.0, 0.0), /* Medium Red */
          XY_Pts (
            -0.05, -0.05,
             0.05, -0.05,
             0.05,  0.05,
            -0.05,  0.05,
            -0.05, -0.05
          )
        ); /* End Polygon */
      ),
      Properties (KO_Comp_Height (0.0))
    ); /* End Keepout */
    Keepout (                                         /* Route Keepout */
      Entity (#2020),
      Type (“Route_All”),
      Layer (“All”),
      Outline (
        Polygon (
          Entity_ID (#2021),
          Line_Color (100.0, 0.0, 0.0), /* Medium Red */
          Fill_Color (100.0, 0.0, 0.0), /* Medium Red */
          XY_Pts (
            -0.05, -0.05,
             0.05, -0.05,
             0.05,  0.05,
            -0.05,  0.05,
            -0.05, -0.05
          )
        ); /* End Polygon */
      )
    ); /* End Keepout */
    Hole (                                            /* Mounting Hole */
      Entity_ID (#2022),
      Type (“Mounting”),
      Shape_Type (“Round”),
      Outline (#2521),
      XY_Loc (0.75, 0.0),
      Rotation (0.0)
    ); /* End Hole */
    Keepout (                                     /* Component Keepout */
      Entity (#2023),
      Type (“Comp_By_Height”),
      Layer (“Conductor_Top”, “Conductor_Bottom”),
      Outline (
        Polygon (
          Entity_ID (#2024),
          Line_Color (100.0, 0.0, 0.0), /* Medium Red */
          Fill_Color (100.0, 0.0, 0.0), /* Medium Red */
          XY_Pts (
            0.70, -0.05,
            0.75, -0.05,
            0.75,  0.05,
            0.70,  0.05,
            0.70, -0.05
          )
        ); /* End Polygon */
      ),
      Properties (KO_Comp_Height (0.0))
    ); /* End Keepout */
    Keepout (                                       /* Routing Keepout */
      Entity (#2025),
      Type (“Route_All”),
      Layer (“All”),
      Outline (
        Polygon (
          Entity_ID (#2026),
          Line_Color (100.0, 0.0, 0.0), /* Medium Red */
          Fill_Color (100.0, 0.0, 0.0), /* Medium Red */
          XY_Pts (
            0.70, -0.05,
            0.75, -0.05,
            0.75,  0.05,
            0.70,  0.05,
            0.70, -0.05
          )
        ); /* End Polygon */
      )
    ); /* End Keepout */
    Hole (                                /* Mounting and Tooling Hole */
      Entity_ID (#2027),
      Type (“Mount_Tool”),
      Shape_Type (“Round”),
      Outline (#2521),
      XY_Loc (0.75, 0.60),
      Rotation (0.0)
    ); /* End Hole */
    Keepout (                                     /* Component Keepout */
      Entity (#2028),
      Type (“Comp_By_Height”),
      Layer (“Conductor_Top”, “Conductor_Bottom”),
      Outline (
        Polygon (
          Entity_ID (#2029),
          Line_Color (100.0, 0.0, 0.0), /* Medium Red */
          Fill_Color (100.0, 0.0, 0.0), /* Medium Red */
          XY_Pts (
            0.70, 0.55,
            0.75, 0.55,
            0.75, 0.60,
            0.70, 0.60,
            0.70, 0.55
          )
        ); /* End Polygon */
      ),
      Properties (KO_Comp_Height (0.0))
    ); /* End Keepout */
    Keepout (                                         /* Route Keepout */
      Entity (#2030),
      Type (“Route_All”),
      Layer (“All”),
      Outline (
        Polygon (
          Entity_ID (#2031),
          Line_Color (100.0, 0.0, 0.0), /* Medium Red */
          Fill_Color (100.0, 0.0, 0.0), /* Medium Red */
          XY_Pts (
            0.70, 0.55,
            0.75, 0.55,
            0.75, 0.60,
            0.70, 0.60,
            0.70, 0.55
          )
        ); /* End Polygon */
      )
    ); /* End Keepout */
    Hole (                                            /* Mounting Hole */
      Entity_ID (#2032),
      Type (“Mounting”),
      Shape_Type (“Round”),
      Outline (#2521),
      XY_Loc (0.0, 0.60),
      Rotation (0.0)
    ); /* End Hole */
    Keepout (                                     /* Component Keepout */
      Entity (#2033),
      Type (“Comp_By_Height”),
      Layer (“Conductor_Top”, “Conductor_Bottom”),
      Outline (
        Polygon (
          Entity_ID (#2034),
          Line_Color (100.0, 0.0, 0.0), /* Medium Red */
          Fill_Color (100.0, 0.0, 0.0), /* Medium Red */
          XY_Pts (
            -0.05, 0.55,
             0.05, 0.55,
             0.05, 0.60,
            -0.05, 0.60,
            -0.05, 0.55
          )
        ); /* End Polygon */
      ),
      Properties (KO_Comp_Height (0.0))
    ); /* End Keepout */
    Keepout (                                         /* Route Keepout */
      Entity (#2035),
      Type (“Route_All”),
      Layer (“All”),
      Outline (
        Polygon (
          Entity_ID (#2036),
          Line_Color (100.0, 0.0, 0.0), /* Medium Red */
          Fill_Color (100.0, 0.0, 0.0), /* Medium Red */
          XY_Pts (
            -0.05, 0.55,
             0.05, 0.55,
             0.05, 0.60,
            -0.05, 0.60,
            -0.05, 0.55
          )
        ); /* End Polygon */
      )
```

```
); /* End Keepout */
                       /********************************************/
                       /* Traces, Filled Areas, and Vias           */
                       /********************************************/

Trace (                                                  /* Trace */
  Entity-ID (#2037),
  Layer (“Conductor_Top”),
  Curve (
    Polyline (
      Entity_ID (#2038),
      Line_Color (0.0, 0.0, 0.0), /* Black */
      Width (0.0125),
      XY_Pts (
        0.10, 0.60,
        0.10, 0.625,
        0.45, 0.625,
        0.45, 0.35,
        0.85, 0.35
      )
    ); /* End Polyline */
  ),
  Net_Name (“Vcc”)
); /* End Trace */
Trace (                                                  /* Trace */
  Entity-ID (#2039),
  Layer (“Conductor_Top”),
  Curve (
    Polyline (
      Entity_ID (#2040),
      Line_Color (0.0, 0.0, 0.0), /* Black */
      Width (0.0125),
      XY_Pts (
        0.00, 0.20,
        0.00, 0.30,
        0.10, 0.30
      )
    ); /* End Polyline */
  ),
  Net_Name (“N51”)
); /* End Trace */
Trace (                                                  /* Trace */
  Entity-ID (#2041),
  Layer (“Conductor_Top”),
  Curve (
    Polyline (
      Entity_ID (#2042),
      Line_Color (0.0, 0.0, 0.0), /* Black */
      Width (0.0125),
      XY_Pts (
        0.40, 0.30,
        0.50, 0.30
      )
    ); /* End Polyline */
  ),
  Net_Name (“N65”)
); /* End Trace */
Trace (                                                  /* Trace */
  Entity-ID (#2043),
  Layer (“Conductor_Top”),
  Curve (
    Polyline (
      Entity_ID (#2044),
      Line_Color (0.0, 0.0, 0.0), /* Black */
      Width (0.0125),
      XY_Pts (
        0.90, 0.20,
        0.80, 0.20
      )
    ); /* End Polyline */
  ),
  Net_Name ("Ground")
); /* End Trace */
Trace (                                                   /* Trace */
  Entity-ID (#2045),
  Layer ("Conductor_Inner_1"),
  Curve (
    Polyline (
      Entity_ID (#2046),
      Line_Color (0.0, 0.0, 0.0), /* Black */
      Width (0.0125),
      XY_Pts (
        0.00, 0.20,
        0.30, 0.20,
        0.30, 0.15
      )
    ); /* End Polyline */
  ),
  Net_Name ("N51")
); /* End Trace */
Trace (                                                   /* Trace */
  Entity-ID (#2047),
  Layer ("Conductor_Inner_1"),
  Curve (
    Polyline (
      Entity_ID (#2048),
      Line_Color (0.0, 0.0, 0.0), /* Black */
      Width (0.0125),
      XY_Pts (
        0.45, 0.15,
        0.45, 0.25,
        0.50, 0.25,
        0.50, 0.30
      )
    ); /* End Polyline */
  ),
  Net_Name ("N65")
); /* End Trace */
Trace (                                                   /* Trace */
  Entity-ID (#2049),
  Layer ("Conductor_Bottom"),
  Curve (
    Polyline (
      Entity_ID (#2050),
      Line_Color (0.0, 0.0, 0.0), /* Black */
      Width (0.0125),
      XY_Pts (
        0.40, 0.40,
        0.40, 0.35,
        0.55, 0.35,
        0.55, 0.30,
        0.60, 0.30
      )
    ); /* End Polyline */
  ),
  Net_Name ("N32")
```

```
); /* End Trace */
Trace (                                                   /* Trace */
  Entity-ID (#2051),
  Layer (“Conductor_Bottom”),
  Curve (
    Polyline (
      Entity_ID (#2052),
      Line_Color (0.0, 0.0, 0.0), /* Black */
      Width (0.0125),
      XY_Pts (
        0.40, 0.30,
        0.50, 0.30
      )
    ); /* End Polyline */
  ),
  Net_Name (“N48”)
); /* End Trace */
Filled_Area (                                       /* Filled Area */
  Entity-ID (#2053),
  Layer (“Conductor_Inner_2”),
  Outline (
    Polygon (
      Entity_ID (#2054),
      Line_Color (92.0, 92.0, 92.0), /* Light Grey */
      Fill_Color (92.0, 92.0, 92.0), /* Light Grey */
      XY_Pts (
        0.300, 0.150,
        1.050, 0.150,
        1.050, 0.400,
        0.475, 0.400,
        0.475, 0.675,
        0.300, 0.675,
        0.300, 0.150
      )
    ); /* End Polygon */
  ),
  Voids (
    Polygon (
      Entity_ID (#2055),
      Line_Color (92.0, 92.0, 92.0), /* Light Grey */
      Fill_Color (100.0, 100.0, 100.0), /* White */
      XY_Pts (
        0.35, 0.25,
        0.60, 0.25,
        0.60, 0.35,
        0.45, 0.35,
        0.45, 0.55,
        0.35, 0.55,
        0.35, 0.25
      )
    ); /* End Polygon */
    Circle (
      Entity_ID (#2056),
      Line_Color (92.0, 92.0, 92.0), /* Light Grey */
      Fill_Color (100.0, 100.0, 100.0), /* White */
      XY_Loc (0.40, 0.60),
      Radius (0.0125)
    ); /* End Circle */
    Circle (
      Entity_ID (#2057),
      Line_Color (92.0, 92.0, 92.0), /* Light Grey */
      Fill_Color (100.0, 100.0, 100.0), /* White */
      XY_Loc (0.80, 0.20),
      Radius (0.0125)
    ); /* End Circle */
  ),
  Net_Name (“Vcc”)
); /* End Filled_Area */
Hole (                                                 /* Via Hole */
  Entity_ID (#2058),
  Type (“Thru_Via”),
  Shape_Type (“Round”),
  Outline (#2523),
  XY_Loc (0.00, 0.20),
  Rotation (0.0),
  Plated (“True”),
  Net_Name (“N51”)
); /* End Hole */
Pad (                                                   /* Via Pad */
  Entity_ID (#2059),
  Type (“Via”),
  XY_Loc (0.00, 0.20),
  Rotation (0.0),
  Layer (“All”),
  Geometry (#2522),
  Net_Name (“N51”)
); /* End Pad */
Hole (                                                 /* Via Hole */
  Entity_ID (#2060),
  Type (“Thru_Via”),
  Shape_Type (“Round”),
  Outline (#2523),
  XY_Loc (0.50, 0.30),
  Rotation (0.0),
  Plated (“True”),
  Net_Name (“N65”)
); /* End Hole */
Pad (                                                   /* Via Pad */
  Entity_ID (#2061),
  Type (“Via”),
  XY_Loc (0.50, 0.30),
  Rotation (0.0),
  Layer (“All”),
  Geometry (#2522),
  Net_Name (“N65”)
); /* End Pad */
Hole (                                                 /* Via Hole */
  Entity_ID (#2062),
  Type (“Blind_Via”),
  Side (“Top”),
  Shape_Type (“Round”),
  Outline (#2523),
  XY_Loc (0.80, 0.20),
  Rotation (0.0),
  Span (“Conductor_Top”, “Conductor_Inner_2”),
  Plated (“True”),
  Net_Name (“Ground”)
); /* End Hole */
Pad (                                                   /* Via Pad */
  Entity_ID (#2063),
  Type (“Via”),
  XY_Loc (0.80, 0.20),
  Rotation (0.0),
  Layer (“Conductor_Top”, “Conductor_Inner_1”),
  Geometry (#2522),
  Net_Name (“Ground”)
); /* End Pad */
                       /********************************************/
                       /* Annotation Describing Board Features     */
                       /********************************************/

Annotation (                                         /* Annotation */
  Entity_ID (#2064),
  Level (“Notes”),
  Entities (
    Leader (
      Entity-ID (#2065),
      Arrow_Start (“True”),
      Arrow_End (“False”),
      Curve (
        Polyline (
          Entity_ID (#2066),
          Line_Color (0.0, 0.0, 0.0), /* Black */
          Width (0.005),
          XY_Pts (
            0.625, -0.075,
            0.775, -0.275
          )
        ); /* End Polyline */
      )
    ); /* End Leader */
    Text (
      Entity_ID (#2067),
      Color (0.0, 0.0, 0.0), /* Black */
      Text_String (“Cavity thru top layer”),
      XY_Loc (0.80, -0.30),
      Height (0.0625),
      Rotation (0.0),
      Font (“Ariel”)
    ); /* End Text */
    Text (
      Entity_ID (#2068),
      Color (0.0, 0.0, 0.0), /* Black */
      Text_String (“(0.03 depth).”),
      XY_Loc (0.80, -0.40),
      Height (0.0625),
      Rotation (0.0),
      Font (“Ariel”)
    ); /* End Text */
  )
); /* End Annotation */
Annotation (                                         /* Annotation */
  Entity_ID (#2069),
  Level (“Notes”),
  Entities (
    Leader (
      Entity-ID (#2070),
      Arrow_Start (“True”),
      Arrow_End (“False”),
      Curve (
        Polyline (
          Entity_ID (#2071),
          Line_Color (0.0, 0.0, 0.0), /* Black */
          Width (0.005),
          XY_Pts (
            0.825, 0.275,
            1.175, 0.425
          )
        ); /* End Polyline */
```

```
         )
        ); /* End Leader */
        Text (
         Entity_ID (#2072),
         Color (0.0, 0.0, 0.0), /* Black */
         Text_String (“Component”),
         XY_Loc (1.20, 0.40),
         Height (0.0625),
         Rotation (0.0),
         Font (“Ariel”)
        ); /* End Text */
        Text (
         Entity_ID (#2073),
         Color (0.0, 0.0, 0.0), /* Black */
         Text_String (“keepout by”),
         XY_Loc (1.20, 0.325),
         Height (0.0625),
         Rotation (0.0),
         Font (“Ariel”)
        ); /* End Text */
        Text (
         Entity_ID (#2074),
         Color (0.0, 0.0, 0.0), /* Black */
         Text_String (“height (0.1”),
         XY_Loc (1.20, 0.25),
         Height (0.0625),
         Rotation (0.0),
         Font (“Ariel”)
        ); /* End Text */
        Text (
         Entity_ID (#2075),
         Color (0.0, 0.0, 0.0), /* Black */
         Text_String (“height).”),
         XY_Loc (1.20, 0.175),
         Height (0.0625),
         Rotation (0.0),
         Font (“Ariel”)
        ); /* End Text */
       )
      ); /* End Annotation */
      Annotation (                    /* Annotation */
       Entity_ID (#2076),
       Level (“Notes”),
       Entities (
        Leader (
         Entity-ID (#2077),
         Arrow_Start (“True”),
         Arrow_End (“False”),
         Curve (
          Polyline (
           Entity_ID (#2078),
           Line_Color (0.0, 0.0, 0.0), /* Black */
           Width (0.005),
           XY_Pts (
            0.525, 0.625,
            0.725, 1.000
           )
          ); /* End Polyline */
         )
        ); /* End Leader */
        Text (
         Entity_ID (#2079),
         Color (0.0, 0.0, 0.0), /* Black */
```

```
            Text_String (“Cutout and related”),
            XY_Loc (0.75, 0.875),
            Height (0.0625),
            Rotation (0.0),
            Font (“Ariel”)
          ); /* End Text */
          Text (
            Entity_ID (#2080),
            Color (0.0, 0.0, 0.0), /* Black */
            Text_String (“keepouts are”),
            XY_Loc (0.75, 0.80),
            Height (0.0625),
            Rotation (0.0),
            Font (“Ariel”)
          ); /* End Text */
          Text (
            Entity_ID (#2081),
            Color (0.0, 0.0, 0.0), /* Black */
            Text_String (“represented as a figure.),
            XY_Loc (0.75, 0.725),
            Height (0.0625),
            Rotation (0.0),
            Font (“Ariel”)
          ); /* End Text */
        )
      ); /* End Annotation */
      Annotation (                                         /* Annotation */
        Entity_ID (#2082),
        Level (“Notes”),
        Entities (
          Leader (
            Entity-ID (#2083),
            Arrow_Start (“True”),
            Arrow_End (“False”),
            Curve (
              Polyline (
                Entity_ID (#2084),
                Line_Color (0.0, 0.0, 0.0), /* Black */
                Width (0.005),
                XY_Pts (
                  -0.075, 0.675,
                  -0.225, 0.975,
                  -0.125, 0.975
                )
              ); /* End Polyline */
            )
          ); /* End Leader */
          Text (
            Entity_ID (#2085),
            Color (0.0, 0.0, 0.0), /* Black */
            Text_String (“Mounting hole with”),
            XY_Loc (-0.10, 0.95),
            Height (0.0625),
            Rotation (0.0),
            Font (“Ariel”)
          ); /* End Text */
          Text (
            Entity_ID (#2086),
            Color (0.0, 0.0, 0.0), /* Black */
            Text_String (“routing & component”),
            XY_Loc (-0.10, 0.875),
            Height (0.0625),
            Rotation (0.0),
            Font (“Ariel”)
          ); /* End Text */
          Text (
            Entity_ID (#2087),
            Color (0.0, 0.0, 0.0), /* Black */
            Text_String (“keepouts.”),
            XY_Loc (-0.10, 0.80),
            Height (0.0625),
            Rotation (0.0),
            Font (“Ariel”)
          ); /* End Text */
        )
      ); /* End Annotation */
    ),
                             /********************************************/
                             /* Component and Sublayout Footprints       */
                             /********************************************/
    Footprints (
                             /********************************************/
                             /* Footprint for Component Instance – U1    */
                             /********************************************/
      Footprint (
        Entity_ID (#2088),
        Assoc_Instance (“U1”)
        Units (“Global”),
        XY_Loc (0.10, 0.60),
        Side (“Top”),
        Rotation (270.0),
        Features (
          Graphic (                                /* Graphic for Refdes */
            Entity_ID (#2089),
            Type (“Silkscreen”),
            Layer (“Silkscreen_Top”),
            Entities (
              Text (
                Entity_ID (#2090),
                Color (0.0, 50.0, 0.0), /* Dark Green */
                Text_String (“U1”),
                XY_Loc (-0.15, -0.15),
                Height (0.1),
                Rotation (90.0),
                Font (“Ariel”)
              ); /* End Text */
            )
          ); /* End Graphic */
          Pad (                             /* Pads and Hole for Pin – 1 */
            Entity_ID (#2091),
            Type (“Pin”),
            Assoc_Pin_ID (“1”),
            XY_Loc (0.00, 0.00),
            Rotation (0.0),
            Layer (“Conductor_Top”),
            Geometry (#2524),
            Net_Name (“Vcc”)
          ); /* End Pad */
          Pad (
            Entity_ID (#2092),
            Type (“Pin”),
            Assoc_Pin_ID (“1”),
            XY_Loc (0.00, 0.00),
            Rotation (0.0),
            Layer (
              “Conductor_Inner_1”,
              “Conductor_Inner_2”,
              “Conductor_Bottom”
            ),
            Geometry (#2525),
            Net_Name (“Vcc”)
          ); /* End Pad */
          Hole (
            Entity_ID (#2093),
            Type (“Thru_Pin”),
            Assoc_Pin_ID (“1”),
            Shape_Type (“Round”),
            Outline (#2526),
            XY_Loc (0.00, 0.00),
            Rotation (0.0),
            Plated (“True”),
            Net_Name (“Vcc”)
          ); /* End Hole */
          Pad (                              /* Pad and Hole for Pin – 2 */
            Entity_ID (#2094),
            Type (“Pin”),
            Assoc_Pin_ID (“2”),
            XY_Loc (0.10, 0.00),
            Rotation (0.0),
            Layer (“All”),
            Geometry (#2525)
          ); /* End Pad */
          Hole (
            Entity_ID (#2095),
            Type (“Thru_Pin”),
            Assoc_Pin_ID (“2”),
            Shape_Type (“Round”),
            Outline (#2526),
            XY_Loc (0.10, 0.00),
            Rotation (0.0),
            Plated (“True”)
          ); /* End Hole */
          Pad (                              /* Pad and Hole for Pin – 3 */
            Entity_ID (#2096),
            Type (“Pin”),
            Assoc_Pin_ID (“3”),
            XY_Loc (0.20, 0.00),
            Rotation (0.0),
            Layer (“All”),
            Geometry (#2525)
          ); /* End Pad */
          Hole (
            Entity_ID (#2097),
            Type (“Thru_Pin”),
            Assoc_Pin_ID (“3”),
            Shape_Type (“Round”),
            Outline (#2526),
            XY_Loc (0.20, 0.00),
            Rotation (0.0),
            Plated (“True”)
          ); /* End Hole */
          Pad (                              /* Pad and Hole for Pin – 4 */
            Entity_ID (#2098),
            Type (“Pin”),
            Assoc_Pin_ID (“4”),
            XY_Loc (0.30, 0.00),
            Rotation (0.0),
            Layer (“All”),
            Geometry (#2525),
            Net_Name (“N51”)
          ); /* End Pad */
          Hole (
            Entity_ID (#2099),
            Type (“Thru_Pin”),
            Assoc_Pin_ID (“4”),
            Shape_Type (“Round”),
            Outline (#2526),
            XY_Loc (0.30, 0.00),
            Rotation (0.0),
            Plated (“True”),
            Net_Name (“N51”)
          ); /* End Hole */
          Pad (                              /* Pad and Hole for Pin – 5 */
            Entity_ID (#2100),
            Type (“Pin”),
            Assoc_Pin_ID (“5”),
            XY_Loc (0.30, 0.30),
            Rotation (0.0),
            Layer (“All”),
            Geometry (#2525),
            Net_Name (“N65”)
          ); /* End Pad */
          Hole (
            Entity_ID (#2101),
            Type (“Thru_Pin”),
            Assoc_Pin_ID (“5”),
            Shape_Type (“Round”),
            Outline (#2526),
            XY_Loc (0.30, 0.30),
            Rotation (0.0),
            Plated (“True”),
            Net_Name (“N65”)
          ); /* End Hole */
          Pad (                              /* Pad and Hole for Pin – 6 */
            Entity_ID (#2102),
            Type (“Pin”),
            Assoc_Pin_ID (“6”),
            XY_Loc (0.20, 0.30),
            Rotation (0.0),
            Layer (“All”),
            Geometry (#2525),
            Net_Name (“N32”)
          ); /* End Pad */
          Hole (
            Entity_ID (#2103),
            Type (“Thru_Pin”),
            Assoc_Pin_ID (“6”),
            Shape_Type (“Round”),
            Outline (#2526),
            XY_Loc (0.20, 0.30),
            Rotation (0.0),
            Plated (“True”),
            Net_Name (“N32”)
          ); /* End Hole */
          Pad (                              /* Pad and Hole for Pin – 7 */
            Entity_ID (#2104),
            Type (“Pin”),
            Assoc_Pin_ID (“7”),
            XY_Loc (0.10, 0.30),
            Rotation (0.0),
            Layer (“All”),
            Geometry (#2525)
          ); /* End Pad */
          Hole (
            Entity_ID (#2105),
            Type (“Thru_Pin”),
            Assoc_Pin_ID (“7”),
            Shape_Type (“Round”),
            Outline (#2526),
            XY_Loc (0.10, 0.30),
            Rotation (0.0),
            Plated (“True”)
          ); /* End Hole */
          Pad (                              /* Pad and Hole for Pin – 8 */
            Entity_ID (#2106),
            Type (“Pin”),
            Assoc_Pin_ID (“8”),
            XY_Loc (0.00, 0.30),
            Rotation (0.0),
            Layer (
              “Conductor_Top”,
              “Conductor_Inner_1”,
              “Conductor_Bottom”
            ),
            Geometry (#2525)
          ); /* End Pad */
          Hole (
            Entity_ID (#2107),
            Type (“Thru_Pin”),
            Assoc_Pin_ID (“8”),
            Shape_Type (“Round”),
            Outline (#2526),
            XY_Loc (0.00, 0.30),
            Rotation (0.0),
            Plated (“True”)
          ); /* End Hole */
        )
      ); /* End Footprint */
                             /********************************************/
                             /* Footprint for Component Instance – C1    */
                             /********************************************/
      Footprint (
        Entity_ID (#2108),
        Assoc_Instance (“C1”)
        Units (“Global”),
        XY_Loc (0.30, 0.00),
        Side (“Top”),
        Rotation (90.0),
        Features (
          Graphic (                                /* Graphic for Refdes */
            Entity_ID (#2109),
            Type (“Silkscreen”),
            Layer (“Silkscreen_Top”),
            Entities (
              Text (
                Entity_ID (#2110),
                Color (0.0, 50.0, 0.0), /* Dark Green */
                Text_String (“C1”),
                XY_Loc (0.05, 0.15),
                Height (0.1),
                Rotation (0.0),
                Font (“Ariel”)
              ); /* End Text */
            )
          ); /* End Graphic */
```

```
    Pad (                                       /* Pad for Pin – 1 */
      Entity_ID (#2111),
      Type (“Pin”),
      Assoc_Pin_ID (“1”),
      XY_Loc (0.0, 0.0),
      Rotation (0.0),
      Layer (“Conductor_Top”),
      Geometry (#2527),
      Net_Name (“N51”)
    ); /* End Pad */
    Pad (                                       /* Pad for Pin – 2 */
      Entity_ID (#2112),
      Type (“Pin”),
      Assoc_Pin_ID (“2”),
      XY_Loc (0.15, 0.00),
      Rotation (0.0),
      Layer (“Conductor_Top”),
      Geometry (#2527)
    ); /* End Pad */
  )
); /* End Footprint */
                       /********************************************/
                       /* Footprint for Component Instance – C2    */
                       /********************************************/
Footprint (
  Entity_ID (#2113),
  Assoc_Instance (“C2”)
  Units (“Global”),
  XY_Loc (0.45, 0.00),
  Side (“Top”),
  Rotation (90.0),
  Features (
    Graphic (                                /* Graphic for Refdes */
      Entity_ID (#2114),
      Type (“Silkscreen”),
      Layer (“Silkscreen_Top”),
      Entities (
        Text (
          Entity_ID (#2115),
          Color (0.0, 50.0, 0.0), /* Dark Green */
          Text_String (“C2”),
          XY_Loc (0.05, -0.05),
          Height (0.1),
          Rotation (0.0),
          Font (“Ariel”)
        ); /* End Text */
      )
    ); /* End Graphic */
    Pad (                                       /* Pad for Pin – 1 */
      Entity_ID (#2116),
      Type (“Pin”),
      Assoc_Pin_ID (“1”),
      XY_Loc (0.00, 0.00),
      Rotation (0.0),
      Layer (“Conductor_Top”),
      Geometry (#2527),
      Net_Name (“N65”)
    ); /* End Pad */
    Pad (                                       /* Pad for Pin – 2 */
      Entity_ID (#2117),
      Type (“Pin”),
      Assoc_Pin_ID (“2”),
      XY_Loc (0.15, 0.00),
      Rotation (0.0),
      Layer (“Conductor_Top”),
      Geometry (#2527)
    ); /* End Pad */
  )
); /* End Footprint */
                       /********************************************/
                       /* Footprint for Component Instance – J1    */
                       /********************************************/
Footprint (
  Entity_ID (#2118),
  Assoc_Instance (“J1”)
  Units (“Global”),
  XY_Loc (0.90, 0.40),
  Side (“Top”),
  Rotation (270.0),
  Features (
    Graphic (                                /* Graphic for Refdes */
      Entity_ID (#2119),
      Type (“Silkscreen”),
      Layer (“Silkscreen_Top”),
      Entities (
        Text (
          Entity_ID (#2120),
          Color (0.0, 50.0, 0.0), /* Dark Green */
          Text_String (“J1”),
          XY_Loc (0.075, -0.15),
          Height (0.1),
          Rotation (90.0),
          Font (“Ariel”)
        ); /* End Text */
      )
    ); /* End Graphic */
    Pad (                                       /* Pad for Pin – 1 */
      Entity_ID (#2121),
      Type (“Pin”),
      Assoc_Pin_ID (“1”),
      XY_Loc (0.00, 0.00),
      Rotation (0.0),
      Layer (“Conductor_Top”, “Conductor_Bottom”),
      Geometry (#2528),
      Net_Name (“Vcc”)
    ); /* End Pad */
    Pad (                                       /* Pad for Pin – 2 */
      Entity_ID (#2122),
      Type (“Pin”),
      Assoc_Pin_ID (“2”),
      XY_Loc (0.10, 0.00),
      Rotation (0.0),
      Layer (“Conductor_Top”, “Conductor_Bottom”),
      Geometry (#2528)
    ); /* End Pad */
    Pad (                                       /* Pad for Pin – 3 */
      Entity_ID (#2123),
      Type (“Pin”),
      Assoc_Pin_ID (“3”),
      XY_Loc (0.20, 0.00),
      Rotation (0.0),
      Layer (“Conductor_Top”, “Conductor_Bottom”),
      Geometry (#2528),
      Net_Name (“Ground”)
    ); /* End Pad */
  )
```

```
); /* End Footprint */
                       /********************************************/
                       /* Footprint for Component Instance – C3    */
                       /********************************************/
Footprint (
  Entity_ID (#2124),
  Assoc_Instance (“C3”)
  XY_Loc (0.75, 0.30),
  Units (“Global”),
  Side (“Bottom”),
  Rotation (0.0),
  Features (
    Graphic (                                /* Graphic for Refdes */
      Entity_ID (#2125),
      Type (“Silkscreen”),
      Layer (“Silkscreen_Bottom”),
      Entities (
        Text (
          Entity_ID (#2126),
          Color (0.0, 50.0, 0.0), /* Dark Green */
          Text_String (“C3”),
          XY_Loc (0.025, -0.125),
          Height (0.1),
          Rotation (0.0),
          Font (“Ariel”)
        ); /* End Text */
      )
    ); /* End Graphic */
    Pad (                                       /* Pad for Pin – 1 */
      Entity_ID (#2127),
      Type (“Pin”),
      Assoc_Pin_ID (“1”),
      XY_Loc (0.00, 0.00),
      Rotation (0.0),
      Layer (“Conductor_Bottom”),
      Geometry (#2527)
    ); /* End Pad */
    Pad (                                       /* Pad for Pin – 2 */
      Entity_ID (#2128),
      Type (“Pin”),
      Assoc_Pin_ID (“2”),
      XY_Loc (0.15, 0.00),
      Rotation (0.0),
      Layer (“Conductor_Bottom”),
      Geometry (#2527),
      Net_Name (“N32”)
    ); /* End Pad */
  )
); /* End Footprint */
                       /********************************************/
                       /* Footprint for the Sublayout              */
                       /********************************************/
Footprint (
  Entity_ID (#2129),
  Assoc_Instance (“Unassigned”),
  Units (“Global”),
  XY_Loc (0.00, 0.00),
  Side (“Top”),
  Rotation (0.0),
  Features (
    Cavity (
      Entity (#2130),
      Surface (“Top”),
            Depth (0.03),
            Outline (
              Polygon (
                Entity_ID (#2131),
                Line_Color (80.0, 93.0, 100.0), /* Light Blue */
                Fill_Color (80.0, 93.0, 100.0), /* Light Blue */
                XY_Pts (
                  0.15, -0.05,
                  0.60, -0.05,
                  0.60,  0.20,
                  0.15,  0.20,
                  0.15, -0.05
                )
              ); /* End Polygon */
            )
          ); /* End Cavity */
          Keepout (
            Entity (#2132),
            Type (“Comp_By_Height”),
            Layer (“Conductor_Top”),
            Outline (
              Polygon (
                Entity_ID (#2133),
                Line_Color (100.0, 0.0, 0.0), /* Medium Red */
                Fill_Color (100.0, 0.0, 0.0), /* Medium Red */
                XY_Pts (
                  0.15, -0.05,
                  0.60, -0.05,
                  0.60,  0.20,
                  0.15,  0.20,
                  0.15, -0.05
                )
              ); /* End Polygon */
            ),
            Properties (KO_Comp_Height (0.0))
          ); /* End Keepout */
          Keepout (
            Entity (#2134),
            Type (“Route_All”),
            Layer (“Conductor_Top”),
            Outline (
              Polygon (
                Entity_ID (#2135),
                Line_Color (100.0, 0.0, 0.0), /* Medium Red */
                Fill_Color (100.0, 0.0, 0.0), /* Medium Red */
                XY_Pts (
                  0.15, -0.05,
                  0.60, -0.05,
                  0.60,  0.20,
                  0.15,  0.20,
                  0.15, -0.05
                )
              ); /* End Polygon */
            )
          ); /* End Keepout */
        )
      ); /* End Footprint */
    )
  ); /* End Board Part */


); /* End Parts Section */
```

```
/*************************************************************************/
/* Referenced Entities Section                                           */
/*************************************************************************/

Ref_Entities (

  Physical_Layer (                                    /* Physical Layers */
    Entity_ID (#2501),
    Layer_Name (“Conductor_Top”),
    Type (“Conductive”),
    Position (1),
    Thickness (0.00)
  ); /* End Physical Layer */
  Physical_Layer (
    Entity_ID (#2502),
    Layer_Name (“Silkscreen_Top”),
    Type (“Silkscreen”),
    Position (2),
    Thickness (0.00)
  ); /* End Physical Layer */
  Physical_Layer (
    Entity_ID (#2503),
    Layer_Name (“Conductor_Inner_1”),
    Type (“Conductive”),
    Position (3),
    Thickness (0.00)
  ); /* End Physical Layer */
  Physical_Layer (
    Entity_ID (#2504),
    Layer_Name (“Conductor_inner_2”),
    Type (“Conductive”),
    Position (4),
    Thickness (0.00)
  ); /* End Physical Layer */
  Physical_Layer (
    Entity_ID (#2505),
    Layer_Name (“Silkscreen_Bottom”),
    Type (“Silkscreen”),
    Position (5),
    Thickness (0.00)
  ); /* End Physical Layer */
  Physical_Layer (
    Entity_ID (#2506),
    Layer_Name (“Conductor_Bottom”),
    Type (“Conductive”),
    Position (6),
    Thickness (0.00)
  ); /* End Physical Layer */
  Circle (                                        /* Tooling Hole Circle */
    Entity_ID (#2521),
    Line_Color (0.0, 0.0, 0.0), /* Black */
    Fill_Color (100.0, 100.0, 100.0), /* White */
    XY_Loc (0.0, 0.0),
    Radius (0.025)
  ); /* End Circle */
  Circle (                                       /* Circular Pad for Via */
    Entity_ID (#2522),
    Line_Color (100.0, 100.0, 0.0), /* Dark Yellow */
    Fill_Color (100.0, 100.0, 0.0), /* Dark Yellow */
    XY_Loc (0.0, 0.0),
    Radius (0.025)
  ); /* End Circle */
  Circle (                                            /* Via Hole Circle */
    Entity_ID (#2523),
    Line_Color (100.0, 100.0, 100.0), /* White */
    Fill_Color (100.0, 100.0, 100.0), /* White */
    XY_Loc (0.0, 0.0),
    Radius (0.0125)
  ); /* End Circle */
  Polygon (                                /* Square Pad Shape for DIP_8 */
    Entity_ID (#2524),
    Line_Color (100.0, 100.0, 0.0), /* Dark Yellow */
    Fill_Color (100.0, 100.0, 0.0), /* Dark Yellow */
    XY_Pts (
      -0.025, -0.025,
       0.025, -0.025,
       0.025,  0.025,
      -0.025,  0.025,
      -0.025, -0.025
    )
  ); /* End Polygon */
  Circle (                               /* Circular Pad Shape for DIP_8 */
    Entity_ID (#2525),
    Line_Color (100.0, 100.0, 0.0), /* Dark Yellow */
    Fill_Color (100.0, 100.0, 0.0), /* Dark Yellow */
    XY_Loc (0.0, 0.0),
    Radius (0.025)
  ); /* End Circle */
  Circle (                                  /* Pin Hole Circle for DIP_8 */
    Entity_ID (#2526),
    Line_Color (100.0, 100.0, 100.0), /* White */
    Fill_Color (100.0, 100.0, 100.0), /* White */
    XY_Loc (0.0, 0.0),
    Radius (0.0125)
  ); /* End Circle */
  Polygon (                             /* Rectangular Pad Shape for Cap */
    Entity_ID (#2527),
    Line_Color (100.0, 100.0, 0.0), /* Dark Yellow */
    Fill_Color (100.0, 100.0, 0.0), /* Dark Yellow */
    XY_Pts (
      -0.025, -0.05,
       0.025, -0.05,
       0.025,  0.05,
      -0.025,  0.05,
      -0.025, -0.05,
    )
  ); /* End Polygon */
  Polycurve_Area (                                 /* Pad Shape for Conn */
    Entity_ID (#2528),
    Line_Color (100.0, 100.0, 0.0), /* Dark Yellow */
    Fill_Color (100.0, 100.0, 0.0), /* Dark Yellow */
    Vertices (
      -0.025, 0.00,   0.0,
       0.025, 0.00, 180.0,
       0.025, 0.20,   0.0,
      -0.025, 0.20,   0.0,
      -0.025, 0.00,   0.0
    )
  ); /* End Polycurve Area */


); /* End Referenced Entities Section */
```

B.3 Board Assembly

This section represents a sample file for the following board assembly.

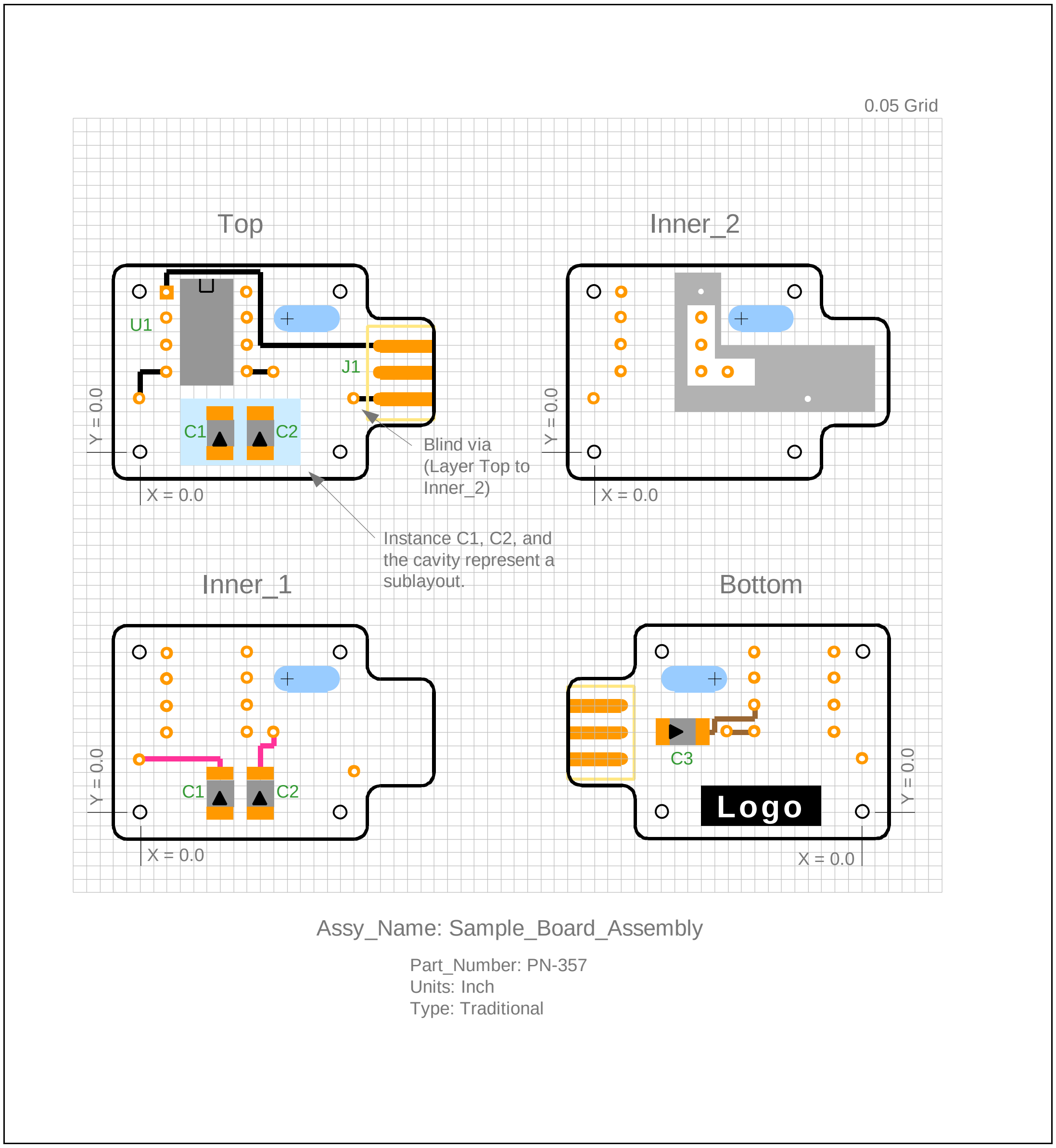

0.05 Grid
Top
Inner_2
U1
J1
C1
C2
Y = 0.0
X = 0.0
Blind via (Layer Top to Inner_2)
Instance C1, C2, and the cavity represent a sublayout.
Inner_1
Bottom
C3
Logo
Assy_Name: Sample_Board_Assembly
Part_Number: PN-357
Units: Inch
Type: Traditional

Figure B.3 Board Assembly

```
/**************************************************************************/
/* Board Assembly                                                         */
/**************************************************************************/

IDF_Header (

  Version (“4.0”),
  Creation_Date_Time (“1998/06/05.10:00:00”),
  Owner_Name (“Tom Makoski”),
  Owner_Phone (“513-528-5059”),
  Owner_EMail (“Tom.Makoski@Intermedius.com”),
  Source_App_Type (“Hand_Generated”),
  Source_App_Vendor (“N/A”),
  Source_App_Name (“N/A”),
  Source_App_Version (“N/A”),
  IDF_Tx_Name (“N/A”),
  IDF_Tx_Version (“N/A”),
  Entity_Count (
    Elec_Part_Defs (0),
    Elec_Part_Insts (5),
    Mech_Part_Defs (0),
    Mech_Part_Insts (1),
    Board_Part_Defs (0),
    Board_Part_Insts (1),
    Board_Assy_Defs (1),
    Board_Assy_Insts (0),
    Panel_Part_Defs (0),
    Panel_Part_Insts (0),
    Panel_Assy_Defs (0),
    Panel_Assy_Insts (0)
  Board_Assy (
    “Board_Assembly”,
    “Board_Part_Instance”,
    “Electrical_Part_Instance”,
    “Mechanical_Part_Instance”,
    “Sublayout”,
  ),
  Default_Units (“Inch”),
  Min_Res (0.0005),
  Notes (
    “This file represents a board assembly.  ”,
    “It only contains part instances and a sublayout.  ”,
    “The corresponding part definitions can be found in the ”,
    “Component Parts and Board Part sample files.  It is not ”,
    “unusual for an IDF file to contain component instances, without ”,
    “the corresponding part definitions.  However, it is very ”,
    “unusual to have a board part instance without the ”,
    “corresponding board part definition.  This is only done “,
    “to make the sample files more readable.”
  )


); /* End IDF_Header Section */

/**************************************************************************/
/* Assemblies Section                                                     */
/**************************************************************************/

Assemblies (
                             /********************************************/
                             /* Sample Board Assembly Definition         */
                             /********************************************/
  Board_Assembly (
    Entity_ID (#3001),
    Assy_Name (“Sample_Board_Assembly”),
    Part_Number (“PN-357”),
    Units (“Global”),
    Type (“Traditional”),
    Board_Inst (
      Board_Part_Instance (                       /* Board Part Instance */
        Entity_ID (#3002),
        Part_Name (“Sample_Board”),
        Refdes (“Unassigned”),
        XY_Loc (0.0, 0.0),
        Rotation (0.0)
      ); /* End Board_Part_Instance */
    ),
    Comp_Insts (
      Electrical_Part_Instance (              /* U1 - Component Instance */
        Entity_ID (#3003),
        Part_Name (“DIP_8”),
        Part_Number (“PN-2245-D”),
        Refdes (“U1”),
        XY_Loc (0.1, 0.6),
        Side (“Top”),
        Rotation (270.0),
        Footprint (#2088)
      ); /* End Electrical_Part_Instance */
      Electrical_Part_Instance (                /* C1 Component Instance */
        Entity_ID (#3004),
        Part_Name (“Cap”),
        Part_Number (“PN-5257-C”),
        Refdes (“C1”),
        XY_Loc (0.3, 0.0),
        Side (“Top”),
        Rotation (90.0),
        Footprint (#2108)
      ); /* End Electrical_Part_Instance */
      Electrical_Part_Instance (                /* C2 Component Instance */
        Entity_ID (#3005),
        Part_Name (“Cap”),
        Part_Number (“PN-5257-C”),
        Refdes (“C2”),
        XY_Loc (0.45, 0.0),
        Side (“Top”),
        Rotation (90.0),
        Footprint (#2113)
      ); /* End Electrical_Part_Instance */
      Electrical_Part_Instance (                /* C3 Component Instance */
        Entity_ID (#3006),
        Part_Name (“Cap”),
        Part_Number (“PN-5257-C”),
        Refdes (“C3”),
        XY_Loc (0.75, 0.3),
        Side (“Bottom”),
        Rotation (0.0),
        Footprint (#2124)
      ); /* End Electrical_Part_Instance */
      Electrical_Part_Instance (                /* J1 Component Instance */
        Entity_ID (#3007),
        Part_Name (“Conn”),
        In_BOM (“False”),
        Refdes (“J1”),
        XY_Loc (0.9, 0.4),
        Side (“Top”),
        Rotation (0.0),
        Footprint (#2118)
      ); /* End Electrical_Part_Instance */
      Mechanical_Part_Instance (                /* M1 Component Instance */
        Entity_ID (#3008),
        Part_Name (“Nameplate”),
        Part_Number (“PN-5614-M”),
        Refdes (“M1”),
        XY_Loc (0.6, -0.5),
        Side (“Bottom”),
        Rotation (0.0)
                                                  /* No footprint for M1 */
      ); /* End Mechanical_Part_Instance */
    ),
    Sublayouts (                                           /* Sublayouts */
      Sublayout (
        Entity-ID (#3009),
        Comp_Insts (#3004, #3005),
        Footprint (#2129)
      ); /* End Sublayout */
    )
  ); /* End Board_Assembly */


); /* End Assemblies Section */
```

B.4 Panel Part

This section represents a sample file for the following panel part definition.

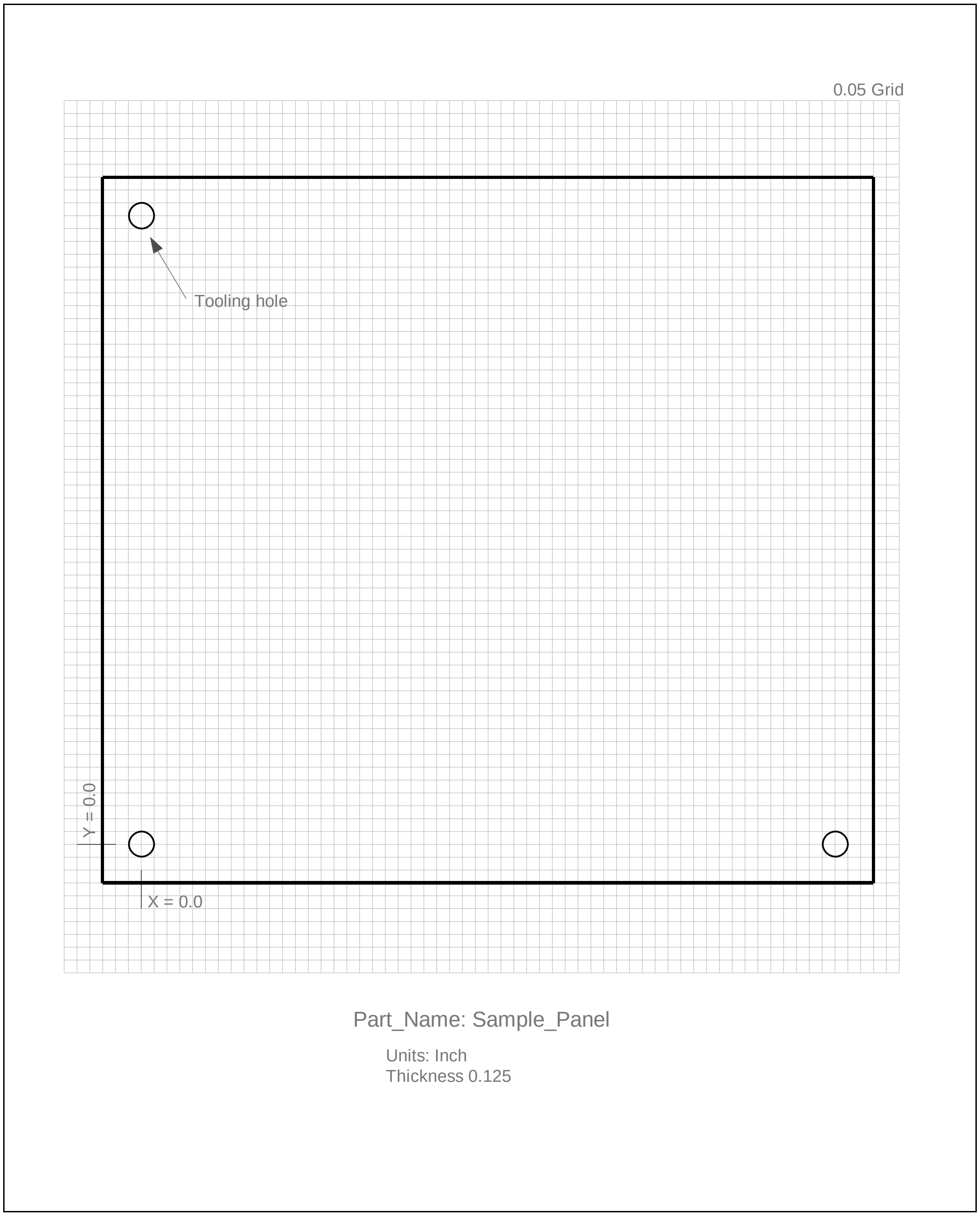


Figure B.4 Panel Part Definition

```
/*************************************************************************/
/* Panel Part                                                            */
/*************************************************************************/

IDF_Header (

  Version (“4.0”),
  Creation_Date_Time (“1998/06/05.10:00:00”),
  Owner_Name (“Tom Makoski”),
  Owner_Phone (“513-528-5059”),
  Owner_EMail (“Tom.Makoski@Intermedius.com”),
  Source_App_Type (“Hand_Generated”),
  Source_App_Vendor (“N/A”),
  Source_App_Name (“N/A”),
  Source_App_Version (“N/A”),
  IDF_Tx_Name (“N/A”),
  IDF_Tx_Version (“N/A”),
  Entity_Count (
    Elec_Part_Defs (0),
    Elec_Part_Insts (0),
    Mech_Part_Defs (0),
    Mech_Part_Insts (0),
    Board_Part_Defs (0),
    Board_Part_Insts (0),
    Board_Assy_Defs (0),
    Board_Assy_Insts (0),
    Panel_Part_Defs (1),
    Panel_Part_Insts (0),
    Panel_Assy_Defs (0),
    Panel_Assy_Insts (0)
  Panel_Part (
    “Circle”,
    “Extrusion”,
    “Footprint”,
    “Graphic”,
    “Hole”,
    “Panel_Part”,
    “Polycurve”,
    “Polygon”,
  ),
  Default_Units (“Inch”),
  Min_Res (0.0005),
  Notes (
    “This file represents a panel part definition.  The panel is very ”,
    “simple.  It only has three tooling holes.  In addition, it ”,
    “contains the footprints for the board assembly instances, ”,
    “that are contributed via the panel assembly.”
  )


); /* End IDF_Header Section */

/*************************************************************************/
/* Parts Section                                                         */
/*************************************************************************/

Parts (
                             /********************************************/
                             /* Sample Panel Part Definition             */
                             /********************************************/
  Panel_Part (
    Entity_ID (#4001),
    Part_Name (“Sample_Panel”),
    Units (“Global”),
    Type (“Unspecified”),
                             /********************************************/
                             /* Panel Shape                              */
                             /********************************************/
    Shape (
      Extrusion (
        Entity_ID (#4002),
        Top_Height (0.125),
        Bot_Height (0.0),
        Outline (
          Polygon (
            Entity_ID (#4003),
            Line_Color (0.0, 0.0, 0.0), /* Black */
            Fill_Color (100.0, 100.0, 100.0), /* White */
            XY_Pts (
              -0.15, -0.15,
               2.70, -0.15,
               2.70,  2.60,
              -0.15,  2.60,
              -0.15, -0.15
            )
          ); /* End Polygon */
        )
      ); /* End Extrusion */
    ),
                             /********************************************/
                             /* Features                                 */
                             /********************************************/
    Features (
      Hole (
        Entity_ID (#4004),
        Type (“Tooling”),
        Shape_Type (“Round”),
        Outline (#4501),
        XY_Loc (0.0, 0.0),
        Rotation (0.0)
      ); /* End Hole */
      Hole (
        Entity_ID (#4005),
        Type (“Tooling”),
        Shape_Type (“Round”),
        Outline (#4501),
        XY_Loc (2.7, 0.0),
        Rotation (0.0)
      ); /* End Hole */
      Hole (
        Entity_ID (#4006),
        Type (“Tooling”),
        Shape_Type (“Round”),
        Outline (#4501),
        XY_Loc (0.0, 2.45),
        Rotation (0.0)
      ); /* End Hole */
    ),
    Footprints (
                          /***********************************************/
                          /* Footprint for Board Assembly Instance - BD1 */
                          /***********************************************/
      Footprint (
        Entity_ID (#4007),
        Assoc_Instance (“BD1”),
        Units (“Global”),
        XY_Loc (0.10, 1.05),
        Side (“Top”),
        Rotation (270.0),
        Features (
          Graphic (
            Entity_ID (#4008),
            Type (“Milling_Path”),
            Layer (“Top”),
            Entities (
              Polycurve (
                Entity_ID (#4009),
                Line_Font (Phantom),
                Line_Color (0.0, 50.0, 0.0), /* Dark Green */
                Width (0.012),
                Vertices (
                  -0.05, -0.15,   0.0,
                   0.80, -0.15,   0.0,
                   0.90, -0.05,  90.0,
                   0.90,  0.00,   0.0,
                   0.95,  0.05, -90.0,
                   1.05,  0.05,   0.0,
                   1.15,  0.15,  90.0,
                   1.15,  0.45,   0.0,
                   1.05,  0.55,  90.0,
                   0.95,  0.55,   0.0,
                   0.90,  0.60, -90.0,
                   0.90,  0.65,   0.0,
                   0.80,  0.75,  90.0,
                   0.05,  0.75,   0.0,
                  -0.15,  0.65,  90.0,
                  -0.15, -0.05,   0.0,
                  -0.05, -0.15,  90.0
                )
              ); /* End Polycurve */
            )
          ); /* End Graphic */
        )
      ); /* End Footprint */
                          /***********************************************/
                          /* Footprint for Board Assembly Instance – BD2 */
                          /***********************************************/
      Footprint (
        Entity_ID (#4010),
        Assoc_Instance (“BD2”),
        Units (“Global”),
        XY_Loc (1.05, 1.05),
        Side (“Top”),
        Rotation (270.0),
        Features (
          Graphic (
            Entity_ID (#4011),
            Type (“Milling_Path”),
            Layer (“Top”),
            Entities (
              Polycurve (
                Entity_ID (#4012),
                Line_Font (Phantom),
                Line_Color (0.0, 50.0, 0.0), /* Dark Green */
                Width (0.012),
                Vertices (
                  -0.05, -0.15,   0.0,
                   0.80, -0.15,   0.0,
                   0.90, -0.05,  90.0,
                   0.90,  0.00,   0.0,
                   0.95,  0.05, -90.0,
                   1.05,  0.05,   0.0,
                   1.15,  0.15,  90.0,
                   1.15,  0.45,   0.0,
                   1.05,  0.55,  90.0,
                   0.95,  0.55,   0.0,
                   0.90,  0.60, -90.0,
                   0.90,  0.65,   0.0,
                   0.80,  0.75,  90.0,
                   0.05,  0.75,   0.0,
                  -0.15,  0.65,  90.0,
                  -0.15, -0.05,   0.0,
                  -0.05, -0.15,  90.0
                )
              ); /* End Polycurve */
            )
          ); /* End Graphic */
        )
      ); /* End Footprint */
                          /***********************************************/
                          /* Footprint for Board Assembly Instance – BD3 */
                          /***********************************************/
      Footprint (
        Entity_ID (#4013),
        Assoc_Instance (“BD3”),
        Units (“Global”),
        XY_Loc (2.00, 1.05),
        Side (“Top”),
        Rotation (270.0),
        Features (
          Graphic (
            Entity_ID (#4014),
            Type (“Milling_Path”),
            Layer (“Top”),
            Entities (
              Polycurve (
                Entity_ID (#4015),
                Line_Font (Phantom),
                Line_Color (0.0, 50.0, 0.0), /* Dark Green */
                Width (0.012),
                Vertices (
                  -0.05, -0.15,   0.0,
                   0.80, -0.15,   0.0,
                   0.90, -0.05,  90.0,
                   0.90,  0.00,   0.0,
                   0.95,  0.05, -90.0,
                   1.05,  0.05,   0.0,
                   1.15,  0.15,  90.0,
                   1.15,  0.45,   0.0,
                   1.05,  0.55,  90.0,
                   0.95,  0.55,   0.0,
                   0.90,  0.60, -90.0,
                   0.90,  0.65,   0.0,
                   0.80,  0.75,  90.0,
                   0.05,  0.75,   0.0,
                  -0.15,  0.65,  90.0,
                  -0.15, -0.05,   0.0,
                  -0.05, -0.15,  90.0
                )
              ); /* End Polycurve */
```

```
          )
        ); /* End Graphic */
      )
    ); /* End Footprint */
                        /***********************************************/
                        /* Footprint for Board Assembly Instance – BD4 */
                        /***********************************************/
    Footprint (
      Entity_ID (#4016),
      Assoc_Instance (“BD4”),
      Units (“Global”),
      XY_Loc (2.60, 1.40),
      Side (“Top”),
      Rotation (90.0),
      Features (
        Graphic (
          Entity_ID (#4017),
          Type (“Milling_Path”),
          Layer (“Top”),
          Entities (
            Polycurve (
              Entity_ID (#4018),
              Line_Font (Phantom),
              Line_Color (0.0, 50.0, 0.0), /* Dark Green */
              Width (0.012),
              Vertices (
                -0.05, -0.15,   0.0,
                 0.80, -0.15,   0.0,
                 0.90, -0.05,  90.0,
                 0.90,  0.00,   0.0,
                 0.95,  0.05, -90.0,
                 1.05,  0.05,   0.0,
                 1.15,  0.15,  90.0,
                 1.15,  0.45,   0.0,
                 1.05,  0.55,  90.0,
                 0.95,  0.55,   0.0,
                 0.90,  0.60, -90.0,
                 0.90,  0.65,   0.0,
                 0.80,  0.75,  90.0,
                 0.05,  0.75,   0.0,
                -0.15,  0.65,  90.0,
                -0.15, -0.05,   0.0,
                -0.05, -0.15,  90.0
              )
            ); /* End Polycurve */
          )
        ); /* End Graphic */
      )
    ); /* End Footprint */
                        /***********************************************/
                        /* Footprint for Board Assembly Instance – BD5 */
                        /***********************************************/
      Footprint (
      Entity_ID (#4019),
      Assoc_Instance (“BD5”),
      Units (“Global”),
      XY_Loc (1.65, 1.40),
      Side (“Top”),
      Rotation (90.0),
      Features (
        Graphic (
          Entity_ID (#4020),
          Type (“Milling_Path”),
          Layer (“Top”),
          Entities (
            Polycurve (
              Entity_ID (#4021),
              Line_Font (Phantom),
              Line_Color (0.0, 50.0, 0.0), /* Dark Green */
              Width (0.012),
              Vertices (
                -0.05, -0.15,   0.0,
                 0.80, -0.15,   0.0,
                 0.90, -0.05,  90.0,
                 0.90,  0.00,   0.0,
                 0.95,  0.05, -90.0,
                 1.05,  0.05,   0.0,
                 1.15,  0.15,  90.0,
                 1.15,  0.45,   0.0,
                 1.05,  0.55,  90.0,
                 0.95,  0.55,   0.0,
                 0.90,  0.60, -90.0,
                 0.90,  0.65,   0.0,
                 0.80,  0.75,  90.0,
                 0.05,  0.75,   0.0,
                -0.15,  0.65,  90.0,
                -0.15, -0.05,   0.0,
                -0.05, -0.15,  90.0
              )
            ); /* End Polycurve */
          )
        ); /* End Graphic */
      )
    ); /* End Footprint */
                        /************************************************/
                        /* Footprint for Board Assembly Instance – BD6 */
                        /************************************************/
    Footprint (
      Entity_ID (#4022),
      Assoc_Instance (“BD6”),
      Units (“Global”),
      XY_Loc (0.70, 1.40),
      Side (“Top”),
      Rotation (90.0),
      Features (
        Graphic (
          Entity_ID (#4023),
          Type (“Milling_Path”),
          Layer (“Top”),
          Entities (
            Polycurve (
              Entity_ID (#4024),
              Line_Font (Phantom),
              Line_Color (0.0, 50.0, 0.0), /* Dark Green */
              Width (0.012),
              Vertices (
                -0.05, -0.15,   0.0,
                 0.80, -0.15,   0.0,
                 0.90, -0.05,  90.0,
                 0.90,  0.00,   0.0,
                 0.95,  0.05, -90.0,
                 1.05,  0.05,   0.0,
                 1.15,  0.15,  90.0,
                 1.15,  0.45,   0.0,
                 1.05,  0.55,  90.0,
                 0.95,  0.55,   0.0,
                       0.90,  0.60, -90.0,
                       0.90,  0.65,   0.0,
                       0.80,  0.75,  90.0,
                       0.05,  0.75,   0.0,
                      -0.15,  0.65,  90.0,
                      -0.15, -0.05,   0.0,
                      -0.05, -0.15,  90.0
                    )
                  ); /* End Polycurve */
                )
              ); /* End Graphic */
            )
          ); /* End Footprint */
        )
      ); /* End Panel Part */


    ); /* End Parts Section */


    /**************************************************************************/
    /* Referenced Entities Section                                            */
    /**************************************************************************/

    Ref_Entities (

      Circle (                                        /* Tooling Hole Circle */
        Entity_ID (#4501),
        Line_Color (0.0, 0.0, 0.0), /* Black */
        Fill_Color (100.0, 100.0, 100.0), /* White */
        XY_Loc (0.0, 0.0),
        Radius (0.05)
      ); /* End Circle */

    ); /* End Referenced Entities Section */
```

B.5 Panel Assembly

This section represents a sample file for the following panel assembly.

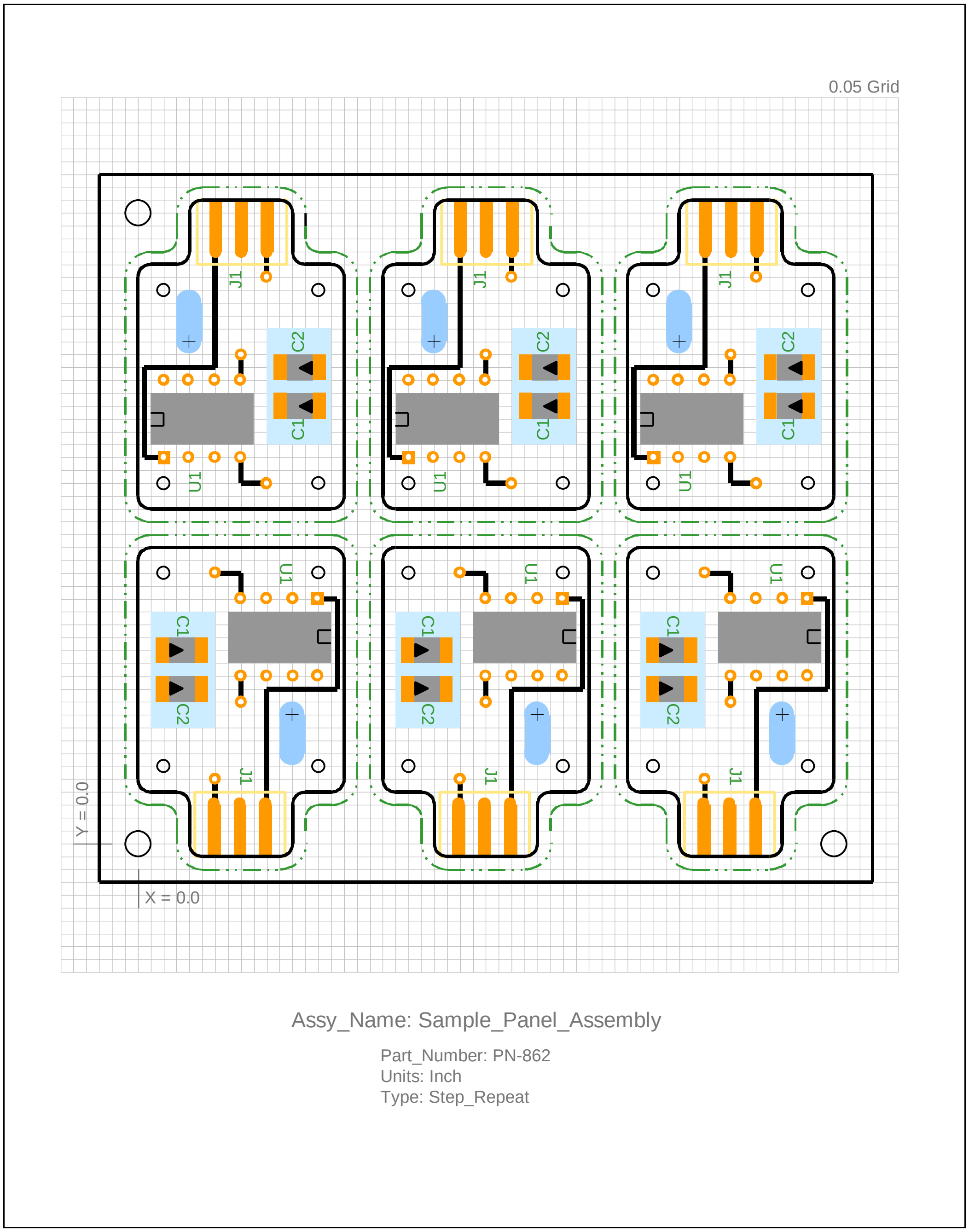


Figure B.5 Panel Assembly

```
/*************************************************************************/
/* Panel Assembly                                                        */
/*************************************************************************/

IDF_Header (

  Version (“4.0”),
  Creation_Date_Time (“1998/06/05.10:00:00”),
  Owner_Name (“Tom Makoski”),
  Owner_Phone (“513-528-5059”),
  Owner_EMail (“Tom.Makoski@Intermedius.com”),
  Source_App_Type (“Hand_Generated”),
  Source_App_Vendor (“N/A”),
  Source_App_Name (“N/A”),
  Source_App_Version (“N/A”),
  IDF_Tx_Name (“N/A”),
  IDF_Tx_Version (“N/A”),
  Entity_Count (
    Elec_Part_Defs (0),
    Elec_Part_Insts (0),
    Mech_Part_Defs (0),
    Mech_Part_Insts (0),
    Board_Part_Defs (0),
    Board_Part_Insts (0),
    Board_Assy_Defs (0),
    Board_Assy_Insts (6),
    Panel_Part_Defs (0),
    Panel_Part_Insts (1),
    Panel_Assy_Defs (1),
    Panel_Assy_Insts (0)
  Panel_Assy (
    “Board_Assembly_Instance”,
    “Panel_Assembly”,
    “Panel_Part_Instance”,
  ),
  Default_Units (“Inch”),
  Min_Res (0.0005),
  Notes (
    “This file represents a panel assembly.  There are six ”,
    “board assembly instances, but no subpanels or component ”,
    “instances on the panel itself.  However, each board assembly ”,
    “instance has an associated footprint, which represents a milling ”,
    “path.  For brevity, the file only contains the assembly instances.  ”,
    “The corresponding board assembly can be found in the Board Assembly” ,
    “sample file.”
  )


); /* End IDF_Header Section */


/*************************************************************************/
/* Assemblies Section                                                    */
/*************************************************************************/

Assemblies (
                             /********************************************/
                             /* Sample Panel Assembly Definition         */
                             /********************************************/
  Panel_Assembly (
    Entity_ID (#5001),
    Assy_Name (“Sample_Panel_Assembly”),
    Part_Number (“PN-862”),
    Units (“Global”),
    Type (“Step_Repeat”),
    Panel_Inst (
      Panel_Part_Instance (                       /* Panel Part Instance */
        Entity_ID (#5002),
        Part_Name (“Sample_Panel”),
        Refdes (“Unassigned”),
        XY_Loc (0.0, 0.0),
        Rotation (0.0)
      ); /* End Panel_Part_Instance */
    ),
    Board_Assys (
      Board_Assembly_Instance (                    /* BD1 Board Instance */
        Entity_ID (#5003),
        Assy_Name (“Sample_Board_Assembly”),
        Refdes (“BD1”),
        XY_Loc (0.10, 1.05),
        Rotation (270.0),
        Mirror (“False”),
        Footprint (#4007)
      ); /* End Board_Assembly_Instance */
      Board_Assembly_Instance (                    /* BD2 Board Instance */
        Entity_ID (#5004),
        Assy_Name (“Sample_Board_Assembly”),
        Refdes (“BD2”),
        XY_Loc (1.05, 1.05),
        Rotation (270.0),
        Mirror (“False”),
        Footprint (#4010)
      ); /* End Board_Assembly_Instance */
      Board_Assembly_Instance (                    /* BD3 Board Instance */
        Entity_ID (#5005),
        Assy_Name (“Sample_Board_Assembly”),
        Refdes (“BD3”),
        XY_Loc (2.00, 1.05),
        Rotation (270.0),
        Mirror (“False”),
        Footprint (#4013)
      ); /* End Board_Assembly_Instance */
      Board_Assembly_Instance (                    /* BD4 Board Instance */
        Entity_ID (#5006),
        Assy_Name (“Sample_Board_Assembly”),
        Refdes (“BD4”),
        XY_Loc (2.60, 1.40),
        Rotation (90.0),
        Mirror (“False”),
        Footprint (#4016)
      ); /* End Board_Assembly_Instance */
      Board_Assembly_Instance (                    /* BD5 Board Instance */
        Entity_ID (#5007),
        Assy_Name (“Sample_Board_Assembly”),
        Refdes (“BD5”),
        XY_Loc (1.65, 1.40),
        Rotation (90.0),
        Mirror (“False”),
        Footprint (#4019)
      ); /* End Board_Assembly_Instance */
      Board_Assembly_Instance (                    /* BD6 Board Instance */
        Entity_ID (#5008),
        Assy_Name (“Sample_Board_Assembly”),
        Refdes (“BD6”),
        XY_Loc (0.70, 1.40),
        Rotation (90.0),
        Mirror (“False”),
        Footprint (#4022)
      ); /* End Board_Assembly_Instance */
    )
  ); /* End Panel_Assembly */


); /* End Assemblies Section */
```