Electrical/Mechanical Design Integration:
An Introduction to IDF 4.0 and What it Can Do for You

David J. Kehmeier
Director, Consulting Services
Intermedius Design Integration, LLC
847 Jansen Avenue
San Jose, California 95125
(408) 279-2903 (Phone)
Dave.Kehmeier@intermedius.com

Abstract

The Intermediate Data Format (IDF) is a specification designed to exchange printed circuit assembly (PCA) data between mechanical design and PCB layout. Initially developed in 1992, the IDF has since become a de facto industry standard, implemented by most CAD vendors and widely used by their customers.

A project funded by vendors and end-users, and managed by Intermedius Design Integration, LLC to develop a new version of the IDF is now complete. The new version, IDF 4.0 Revision A provides significant new capabilities beyond IDF 2.0/3.0. It has a new syntax and data representation, and many new entities.

This paper provides an introduction to IDF 4.0 and its capabilities from a user's perspective. It also describes a project for an industry-wide implementation of IDF 4.0 designed to ensure that vendor translators are developed in a consistent and timely manner, enabling industry to quickly and effectively realize the full potential of IDF 4.0.

Development of IDF 4.0

The project to develop IDF 4.0 was conceived and conducted by Intermedius with funding and support from leading CAD vendors including Bentley Systems, Cadence, CoCreate, Incases, LTX (EDS/Unigraphics), Maya Heat Transfer, Mentor Graphics, OrCAD, PADS, PTC, SDRC, VeriBest, and Zuken-Redac. Funding and support was also provided by Harris, HP, Intel, Delco, Square D, and the Visteon Division of Ford.

In the first phase of the project, (from October, 1997 to February, 1998), Intermedius conducted extensive interviews (mostly on-site) to gather and document requirements from 24 end-user companies that currently use MCAD and ECAD software from a variety of software vendors to design their products. In the second phase of the project (from March, 1998 to July, 1998), Intermedius developed the specification based on these requirements, with review by the participating companies. The specification is officially titled "IDF 4.0, Revision A (Pre-implementation Draft)."

Overview of IDF 4.0

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.

IDF 4.0 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 including:

- 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)

IDF 4.0 does not provide a full product or design representation of a PCA. 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 complete means for manufacturing, assembling, testing, or creating detailed documentation of a PCA

Data Content

The IDF 4.0 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.

IDF 4.0 also provides figures, footprints, and sublayouts to represent associations among instances and features that are useful to maintain during the design of the PCA. These are described below.

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 IDF 4.0 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.

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. As a result, a single IDF part definition can be used to represent several actual physical parts.

Panel and Board Parts

Every panel 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 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 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.

Component Parts

Parts that are assembled to a board part (and occasionally, a panel part) are “components”. IDF 4.0 supports two types of component parts: electrical parts and mechanical parts. Electrical parts have pins 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 and board parts, the shape of a component part is represented by one or more extrusions, along with cutouts and cavities as required. Component parts can have mounting side and opposite side shapes. A component part shape can have multiple extrusions to provide a more complete 3D representation of the part. IDF 4.0 supports actual physical component parts (those that are included in a manufacturing Bill of Materials) as well as “printed” electrical parts that are fabricated into or onto the board itself.

Assemblies and Instances

IDF 4.0 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. In this way, parts and assemblies are re-used rather than recreated for each instance.

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 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.

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 on the panel part of a panel assembly, board assembly instances and panel assembly instances are located in the panel part of a panel assembly.

Features

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 IDF 4.0.

Cavities, Cutouts, and Extrusions

All parts 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. 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.

Holes

Holes are features in panel 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). Holes are typically round but may be square, slots, keyholes, or any other shape. They can be plated or unplated. Plated holes can have associated net names.

Conductors

Conductors represent the physical interconnect among the electrical part instances in a board assembly. They are included in IDF 4.0 to enhance the physical shape of the board part but are not intended to convey the electrical connectivity of the design. Conductors are represented in IDF 4.0 by pads, traces, and filled areas. All conductors can have associated net names.

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 cannot be represented using 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.

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. IDF 4.0 supports several predefined keepin and keepout types. The keepin and keepout types can be extended by the user.

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 IDF 4.0 to transfer conductor shapes on internal layers for reference purposes.

Footprints

A footprint represents a set of board or panel features that are required to support the instance 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 part instance are soldered, exist in the actual board part and are associated with the electrical part instance, so that if the instance is repositioned or removed, the pads “go with it”.

Figures

A figure represents a named instance of a set of features that exist in a panel 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 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.

Sublayouts

A sublayout defines a group of component instances in a board assembly. 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.

Geometry

Geometry supported by IDF 4.0 (for constructing features) consists of planar curves and planar areas. 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). 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). All 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.

Analysis

IDF 4.0 supports the data that is necessary to interface to both thermal and structural analysis applications. In addition to shape information (defined by extrusions, cutouts, cavities, and holes) which is required for analysis, 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.

Properties

IDF entities may 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.

Change Control

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. 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.

Data Representation

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 following data types:

Boolean	Used to represent an attribute value of True or False.
Integer	Used to represent a signed whole number value.
Real	Used to represent a floating point value.
String	Used to represent an ASCII character string value.
Enum	Used to represent an attribute value that must be one of an enumerated list of possible values.
Ext_Enum	Used to represent a user (or vendor) extensible enumeration.
Entity	Used to represent a contained (embedded) entity.
Ent_ID	Used to represent a unique entity ID, within the scope of all entities in the IDF file.
Ent_Name	Used to represent a unique entity name, within the scope of all like entities in the IDF file.
Ref_ID	Used to represent an entity reference by entity ID.
Ref_Name	Used to represent an entity reference by entity name.
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.

If an attribute's value is another entity, the entity is usually contained (the entity and its attributes are embedded) but in some cases, the entity is referenced (the entity and its attributes are located elsewhere in the file).

File Format

The IDF file format is an ASCII format. An IDF file is organized into the following four sections:

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 (boards and panels) in the IDF file, and follows the IDF_Header section.

Parts The Parts section contains all the part (board, panel, component) entities in the IDF file, and follows the Assemblies section, if it exists.

Ref_Entities The Ref_Entities section contains all of the entities that are referenced from the Assemblies and Parts sections.

Sections, entities, and attributes all have predefined names ("keywords"). The entities within a section, attributes within an entity, and values within an attribute are enclosed in parentheses. Commas are used to delimit values in an attribute and attributes within an entity. Semicolons are used to delimit entities within a section and sections within a file.

Although the format is in general a free format, specific formatting rules are defined to enhance the readability of the file and make it easier to parse with a scripting language such as PERL.

Example

The following example of a capacitor definition illustrates a portion of an IDF 4.0 file:

```
Electrical_Part (
  Entity_ID (#1014),
  Part_Name (“Cap”),
  Units (Inch),
  Type (Surface),
  Mnt_Shape (                                          /* Mounting Side Shape */
    Extrusion (
      Entity_ID (#1015),
      Feature_ID (“Unassigned”),
      Top_Height (0.05),
      Bot_Height (0.0),
      Outline (
        Polygon (
          Entity_ID (#1016),
          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 */
  ),
  Properties (
    EL_Capacitance (0.1),
    EL_Tolerance (0.05)
  )
); /* End Electrical Part */
```

IDF 4.0 Implementation Program

With the development of IDF 4.0 complete, Intermedius is now planning an industry-wide implementation program. The program is designed to ensure that vendor translators are developed in a consistent and timely manner, and address real end-user needs. Intermedius believes that this type of program can overcome many of the problems that plague other data exchange standardization efforts, and help industry quickly and effectively realize the benefits of IDF 4.0.

Program Overview

Intermedius will design and manage the IDF V4.0 Implementation Program. The planned duration of the IDF V4.0 Implementation Program is 18 months, and is targeted to start in Q4, 1998. The key elements of the program include capturing and communicating end-user requirements for IDF translation, providing implementation tools and technology, designing and facilitating realistic end-user functionality tests, and establishing a methodology for "characterizing" the capabilities of vendor translators.

Program Goals

The major goals of the IDF V4.0 Implementation Program are to:

- Effectively communicate the end-user requirements for IDF translation to the vendor community.
- Facilitate the development of useable, reliable, and consistent vendor IDF translators.
- Establish IDF V4.0 as a useful and viable de facto PCA design integration standard, and build an infrastructure to maintain and support the IDF on an ongoing basis.

Program Benefits

The key benefits of the IDF V4.0 Implementation Program are:

- Collectively, end-users will "drive" a consistent and timely industry-wide implementation of IDF V4.0, during which the specification will be verified and finalized.
- End-user IDF translation requirements and usage scenarios will be captured, documented, and effectively communicated to the vendor community.
- Vendors will be able to collaborate, test, and verify their IDF translator capabilities in an open and cooperative working environment.
- The costs for developing IDF translator tools, technology, and test cases will be spread across multiple vendors and end-user companies.