Mentor®
A Siemens Business

ODB++ Format Specification

Format Version 8.1 Update 1

July 2017

Note - Viewing PDF files within a web browser causes some links not to function (see MG595892).
Use HTML for full navigation

Mentor Graphics Corporation
8005 S.W. Boeckman Road, Wilsonville, Oregon 97070-7777
Telephone: 503.685.7000
Toll-Free Telephone: 800.592.2210
Website: www.mentor.com
SupportNet: supportnet.mentor.com/

Send Feedback on Documentation: supportnet.mentor.com/doc_feedback_form

Table of Contents

List of Tables

Chapter 2
ODB++ Format Specification

Competing in global markets requires PCB design organizations to continually seek ways to more effectively communicate their product definition data to fabrication, assembly and test organizations. The goal is to simultaneously maximize three aspects:

- Speed in transferring a new product design into the manufacturing process
- Accuracy of communication of the designer's intentions to the manufacturer
- Opportunities for reduction in total manufacturing cost

All the data transferred to manufacturers by a designer that defines the PCB for the purposes of bare board fabrication, assembly and test can be referred to as the manufacturing product model. To the manufacturer, this product model defines 'what' should be manufactured and delivered back to the product owner. The manufacturer's task is to validate the manufacturability of the product, then derive and execute the optimal manufacturing process for its manufacture.

The ODB++ data exchange format is the most widely accepted, integrated product model format for efficiently handing off a PCB from design into manufacturing. The ODB++ format is a simple yet comprehensive description of all entities necessary to answer the question of 'what' needs to be manufactured, covering fabrication, assembly and test of a PCB.

Changes to the ODB++ Format

Changes have been made to the ODB++ format in V8.0 and since V8.0.

ODB++ 8.1 Update 1

These changes were implemented in this version:

- The manual was fixed to specify the correct formula used to round the line ends for a line thermal symbol.

  See “Basic Standard Symbols” on page 169.

- ODB++ now supports HDI netlists.

  See “eda/hdi_netlist (HDI Netlist)” on page 121.

- ODB++ now supports layers containing wire bond information.

  See “Layer Subtype to Support Wire Bonding” on page 54.

- ODB++ now supports dimensions.

  See “<layer_name>/dimensions (Dimensions)” on page 128.

- The manual was fixed to remove an extraneous semicolon from the syntax of the barcode record of the <layer_name>/features entity.

  See “B - Barcode Records” on page 154.

- The manual was fixed to specify the correct limits for measurements such as coordinates and distances. The Cartesian coordinates supported are within the range of (-100, -100) to (100, 100) inches, which is (-2450, -2450) to (2450, 2450) mm.

  See “Units of Measurement” on page 18.

ODB++ 8.1

These changes were implemented in this version:

- Metadata has been added to the ODB++ format.

  See “misc/metadata.xml (Metadata)” on page 60.

- A *shorts* file has been added to the ODB++ format for defining a list of features that are known to create intentional net shorts within the product model.

  See “eda/shorts (Electrical Intention Short Net)” on page 116.

- A *zones* file has been added to the ODB++ format for defining regions of the board where the board layer materials, including the layer types, are identical.

  See “<step_name>/zones (Matrix Broken Into Sections)” on page 94.

- This manual has been reorganized to more closely reflect the ODB++ directory structure:
  - misc/sysattr.* section moved from the Basic Principles chapter to the misc directory description.

    See “misc/sysattr.* (Attribute Definition)” on page 66.

  - misc/userattr section added to the misc directory description.

    See “misc/userattr (User Attribute Definition)” on page 74.

  - A discussion of attribute value assignment was added - using an *attrlist* file or using lookup tables in a file defining features, components, or nets.

    See “Attribute Value Assignment” on page 23.

- Additional supported fields are available for the LAYER array of the matrix file.

  See “Fields in the LAYER Array of a Matrix File” on page 46.

ODB++ 8.0 Update 6

These changes were implemented in this version:

- Section on layer types to support embedded resistors and capacitors added in Product Model Entities.

See “Layer Subtypes to Support Embedded Resistors and Capacitors” on page 52.

- Added test attributes .net_test_type, .net_test_current, .net_test_voltage.
  - See “ODB++ System Attributes - Alphabetical List” on page 189.
  - See “Test (sysattr.test)” on page 249.
- Added amps (for current) and volts (for voltage) to the table of units in which data can be stored.

  See “Units of Measurement” on page 18.
- Added options CURRENT and VOLTAGE to possible options for UNIT_TYPE parameter of an attribute of type FLOAT.

  See “FLOAT Attribute Definition” on page 72.
- Updated the table How the UNITS Directive Determines Units of Measurement, to eliminate ambiguity.

  See “Units of Measurement” on page 18.

ODB++ 8.0 Update 5

These changes were implemented in this version:

- Maximum value for .z0impedance changed from 0.5 to 10000.

  See “ODB++ System Attributes - Alphabetical List” on page 189.
- Definition of the side parameter in netlist file format has been further clarified.

  See “cadnet/netlist (cadnet)” on page 86.
- A comment was added to the effect that surfaces must have at least one polygon defined.

  See “S - Surface Records” on page 156.
- The description of unique IDs was clarified.

  See “Unique ID” on page 25.
- Explanation of record type F (number of features) added to the description of the features file.

  See “<layer_name>/features (Graphic Features)” on page 139.

ODB++ 8.0 Update 4

These changes were implemented in this version:

- Parameter descriptions for comp_num and toep_num corrected, and clarification added to fill_size for toeprint section.

  See “SNT— Subnet Record” on page 110.

- Clarification of layer profile use when a step profile hole is found within or intersecting a layer profile island.

  See “<layer_name>/profile (Outline Shape of Layer)” on page 136.

- Clarification of the relationship between records PKG and CMP.

  See “PKG—Package Record” on page 112 and “CMP—Component Record” on page 161

- Section on Angles added to Basic Principles section to relate to the angle definition in Step and Repeat records and the orient_def field in Pad records, Text records, and Barcode records.

  See “Angles” on page 29.

- DSC parameter in BOM Description Records corrected from ‘up to 5’ to ‘unlimited’.

  See “BOM Description Records” on page 163.

- Clarification of what constitutes a legal net name.

  See the description of $<serial_num> <net_name> in “cadnet/netlist (cadnet)” on page 86 and <net_name> in “NET—Electrical Net Record” on page 110.

ODB++ 8.0 Update 3

These changes were implemented in this version:

- UNIT_TYPE of RESISTIVITY requires a UNITS value.
- Two new UNITS values available when UNIT_TYPE=RESISTIVITY—OHM and NANO_OHM.

  See “FLOAT Attribute Definition” on page 72.

- UNITS value NANO_OHM added to system attribute .bulk_resistivity.
- COPPER_WEIGHT added to the list of available UNIT_TYPE values.

  See “FLOAT Attribute Definition” on page 72.

- In the cadnet/netlist (cadnet) step entity, the definition of the radius field was expanded.

  See “cadnet/netlist (cadnet)” on page 86.

- Two product attributes (sysattr) .centroid_correction_x and .centroid_correction_y added.

See “ODB++ System Attributes - Alphabetical List” on page 189.

- These attributes have been updated to include the required **UNITS** tag, in the relevant sysattr file:

sysattr	sysattr.assy	sysattr.dfm	sysattr.fab
.board_thickness	.spo_h_val	.ar_pad_drill_bottom_max	.et_adjacency
.bulk_resistivity	.spo_move_center	.ar_pad_drill_bottom_min	.fill_dx
.comp_height	.spo_p_val	.ar_pad_drill_inner_max	.fill_dy
.comp_height_max	.spo_s_val	.ar_pad_drill_inner_min	.image_dx
.comp_htol_minus	.spo_w_val	.ar_pad_drill_top_max	.image_dy
.comp_htol_plus		.ar_pad_drill_top_min	
.layer_dielectric		.ar_sm_drill_bottom_max	
.pitch		.ar_sm_drill_bottom_min	
.testprobe_diameter		.ar_sm_drill_top_max	
.thickness_over_cu		.ar_sm_drill_top_min	
.thickness_over_sm		.ar_sm_pad_bottom_max	
		.ar_sm_pad_bottom_min	
		.ar_sm_pad_top_max	
		.ar_sm_pad_top_min	
		.dpair_gap	
		.drc_max_height	
		.drc_min_height	
		.eclass_max_stub_length	
		.eclass_min_stub_length	
		.local_fiducial_dist	
		.min_line_width	
		.net_length_max	
		.net_length_min	
		.toep_spacing_req	

For a description of each attribute, see “ODB++ System Attributes - Alphabetical List” on page 189.

ODB++ 8.0 Update 2

These changes were implemented in this version:

- Improved the netlist example to include the use of ‘ld’.

  See “cadnet/netlist (cadnet)” on page 86.

- Corrected the syntax for the optimization (H) record.

  See “cadnet/netlist (cadnet)” on page 86.

- Tools (Drill Tools):

  - Corrected to indicate units of measurement for MIN_TOL, MAX_TOL, FINISH_SIZE and DRILL_SIZE.
  - Added in the description of FINISH_SIZE that if this value is not set, the value should be -1.

  See “<layer_name>/tools (Drill Tools)” on page 165.

- Clarified that the decimal character is required to be a period, and no other representation of numbers (scientific or exponential) is allowed.

  See “Coordinates” on page 29.

- Clarified that the decimal character is required to be a period.

  See “Symbols” on page 29.

- The compression attribute was added to *bom*, and the example was modified.

  See “<bom_name>/bom (Bill Of Materials)” on page 99.

- More detailed descriptions were added to the START_NAME and END_NAME fields in the LAYER array of the matrix file.

  See “Fields in the LAYER Array of a Matrix File” on page 46.

ODB++ 8.0 Update 1

These changes were implemented in this version:

- Default color for layer display in the layer list is 0, indicating no preference.

  See the COLOR field in “Fields in the LAYER Array of a Matrix File” on page 46.

- Netlist definition enhanced to include staggered.

  See the example for “cadnet/netlist (cadnet)” on page 86.

- Corrected format syntax.

  See “NET—Electrical Net Record” on page 110.

- SNT VIA and SNT TRC records were clarified to not contain any additional parameters.

  See “SNT— Subnet Record” on page 110.

- PKG—Package Record — Corrected format syntax.

  See “PKG—Package Record” on page 112.

- PIN—Pin Record — Clarified the determination of pin #1.

  See “PIN—Pin Record” on page 113.

ODB++ 8.0

These changes were implemented in this version:

- Attributes are divided according to area: DFM, Product, or Process. The Process attributes are further classified as Fab, Assem, Test, or Generic.

  See “ODB++ Attributes” on page 22.

- Includes Flex and Flex-rigid PCB modeling.

  See “Layer Subtypes to Support Flex/Rigid Flex Manufacturing” on page 52.

- Symbols added for Solder Stencil Design:
  - Home Plate
  - Inverted Home Plate
  - Flat Home Plate
  - Radiused Inverted Home Plate
  - Radiused Home Plate
  - Cross
  - Dogbone
  - D-Pack

  See “Symbols Suitable for Solder Stencil Design” on page 184.

- Default metric/imperial measurement used throughout product model definition. The units directive UNITS=MM|INCH can be added at the beginning of any file that contains measurable entities.

  See “Units of Measurement” on page 18.

- Drill span direction.
- Creation of profile with holes enabled.

- Net name length enlarged. There is no limit on the number of characters in a net name.
- Support for test probes. Test Probe Diameter attribute introduced to provide information on the size of test probes.
- PCB build up information stored in product model.
- Support for increased number of BOM Desc attributes. Number of descriptions in BOM (DESC<index> or PART_DESC<index>) is unlimited.
- Introduction of package attributes.
- It is now possible to implement component Bill of Material (BOM) information within the component file.

File System

The ODB++ format uses a standard file system structure. A product model stored in the ODB++ format is represented by an independent and self-contained directory tree. This provides the ability to transfer the product model from design systems into manufacturing systems in an efficient manner, without any loss of information.

The advantages of a directory tree, compared to one large file, are apparent when a product model is being read from disk or saved to disk. The flexible tree structure allows you to read or save exactly the required part of the product model, avoiding the overhead of reading and writing a large file, if only a subset of the information is required.

When a product model tree must be transferred to another system, standard tar and compression utilities can be used to convert a directory tree into a single flat file.

These files and directories are mandatory. Other files and directories discussed in this document are included only when needed.

- *<product_model_name>/matrix/matrix*
- *<product_model_name>/misc/info*
- *<product_model_name>/fonts/standard*
- *<product_model_name>/steps/<step_name>/stephdr*
- *<product_model_name>/steps/<step_name>/layers/<layer_name>/features* (or *features.Z*)

There are links between files, defined implicitly in the ODB++ definition, that create dependencies between files.

For example, the file */<step_name>/layers/comp_+_top/components* contains links to */<step_name>/eda/data*, and the */<step_name>/layers/<layer_name>/features* file contains links to user-defined symbols located in *<product_model_name>/symbols*.

Legal Entity Names

The names of these ODB++ entities must comply with the rules for legal entity names: product model, step, layer, symbol, and attribute.

- The length of an entity name must not exceed 64 characters.
- An entity name may contain only these characters:
  - Lower case letters (a through z).
  - Digits (0 through 9).
  - Punctuation—dash (-), underscore (_), dot (.) and plus (+).
- Entity names must not start with a dot (**.**), hyphen (-), or plus (+).

  The exception is system attribute names, which start with a dot. Names of user-defined attributes must not start with a dot.
- Entity names must not end with a dot (.).

Units of Measurement

Unlike traditional CAM formats such as Gerber, ODB++ format represents a complete product model rather than a collection of unrelated geometric entities. All ODB++ format compatible applications must be able to represent the ODB++ product model correctly. For example, if two individual features do not touch, this relationship must be maintained by the ODB++ compatible application.

Minimum resolution for maintaining feature representation is 1/400 micron (1/10160 mil). Minimum line/arc width is 1/400 micron. Minimum measurable distance—or placement tolerance for any feature—is 1/400 micron.

Measurements such as coordinates and distances must be within the supported range. The Cartesian coordinates supported are within the range of (-100, -100) to (100, 100) inches, which is (-2450, -2450) to (2450, 2450) mm.

Files that contain measurable entities must include the UNITS directive, to specify whether measurements in the file are given in metric units (UNITS=MM) or in imperial units (UNITS=INCH). The UNITS directive is placed before the first line that uses units. Only one UNITS directive can appear in a file.

The default units of measurement for the product model are as defined in the UNITS directive in the file *misc/info* of the product model. If it is not defined for the product model, the default is imperial.

For newly created files, units are based on the default defined for the product model.

These files must contain the UNITS directive:

File Name	Path
info	*<product_model_name>/misc*
sysattr *sysattr.dfm* *sysattr.fab* *sysattr.assy* *sysattr.test* *sysattr.gen* *userattr*	*<product_model_name>/misc*
attrlist	*<product_model_name>/misc* *<product_model_name>/steps/<step_name>* *<product_model_name>/steps/<step_name>/layers/<layer_name>* *<product_model_name>/symbols/<symbol_name>* *<product_model_name>/wheels/<wheel_name>*
stephdr	*<product_model_name>/steps/<step_name>*
profile	*<product_model_name>/steps/<step_name>* *<product_model_name>/steps/<step_name>/layers/<layer_name>*
data	*<product_model_name>/steps/<step_name>/eda*
features	*<product_model_name>/steps/<step_name>/layers/<layer_name>* *<product_model_name>/symbols/<symbol_name>*
tools	*<product_model_name>/steps/<step_name>/layers/<layer_name>*
components	*<product_model_name>/steps/<step_name>/layers/comp_+_top* *<product_model_name>/steps/<step_name>/layers/comp_+_bot*
netlist	*<product_model_name>/steps/<step_name>/netlists/cadnet* *<product_model_name>/steps/<step_name>/netlists/refnet*

For each file described in this document, there is a table of information. The UNITS row of the file information table indicates whether the UNITS directive is required for that file:

<file_name>	
Type	Line Record Text or Structured Text
Compression	None or Yes
Path	Path to the file.
UNITS	The UNITS directive is required or The UNITS directive is not relevant for the *<file_name>* file.

According to the unit of measurement selected, data will be stored in these units:

Table 2-1. How Measurements Are Stored

Type of Data	Metric (UNITS=MM)	Imperial (UNITS=INCH)
area	square mm	square inch
copper weight	µm In metric units, copper weight is actually copper thickness. The application converts it accordingly into commonly used neutral units (thickness or weight).	ounce/square foot
current	amps	amps
distances, sizes, coordinates	mm	inch
resistance	ohm	ohm
temperature	centigrade	fahrenheit
voltage	volts	volts
weight	gram	ounce

This table describes how units of measure are defined:

Table 2-2. How the UNITS Directive Determines Units of Measurement

Location of UNITS Directive	Scope	Example
If there is no UNITS directive in *<product_model_name>/misc/info*.	The default units of measurement for the product model is imperial units.	None

If there is a UNITS directive in *<product_model_name>/misc/info*.	The units of measurement specified in this UNITS directive is the default for the product model.	*info* file example:
		
		`JOB_NAME=odbv8_demo_mm`
		`ODB_VERSION_MAJOR=8`
		`ODB_VERSION_MINOR=0`
		`ODB_SOURCE=PADS-POWERPCB-V9.5-METRIC-250L`
		`CREATION_DATE=20140913.130051`
		`SAVE_DATE=20120914.132825`
		`SAVE_APP=Valor NPI 10.0Dev`
		`SAVE_USER=roni`
		`UNITS=MM`
		`MAX_UID=431990`

If there is a UNITS directive in a file that supports it, and there are no other specifications of units of measurement in the file.	The units of measurement specified in the UNITS directive is used for all measurements in the file.	*stephdr* file example:
		
		`UNITS=INCH`
		`X_DATUM=0`
		`Y_DATUM=0`
		`X_ORIGIN=0`
		`Y_ORIGIN=0`
		`TOP_ACTIVE=0`
		`BOTTOM_ACTIVE=0`
		`RIGHT_ACTIVE=0`
		`LEFT_ACTIVE=0`
		`AFFECTING_BOM=`
		`AFFECTING_BOM_CHANGED=0`
		`ID=0`

If there is a UNITS directive in a file that supports it, and individual records in the file also specify units of measurement.	The units of measurement specified in the UNITS directive is used except where this is overridden by the specification in an individual record in the file.	*features* file example:
		The "I" at the end of the symbol definition line specifies imperial units. This overrides the specification of UNITS=MM.
		
		`UNITS=MM`
		`ID=616`
		`#`
		`#Num Features`
		`#`
		`F 12397`
		`#`
		`#Feature symbol names`
		`$0 oval210x170` **`I`**

ODB++ Attributes

Attributes in the ODB++ format provide a flexible way to assign product related and process related information to ODB++ entities. The use of attributes by the ODB++ format has been standardized and is supported in a unified manner, independent of any application-specific references to attributes. All system attributes are optional.

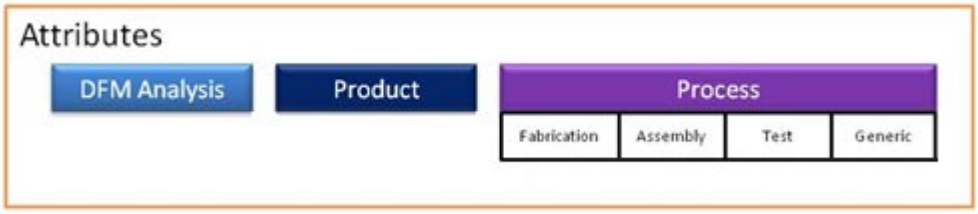
Attribute definition files for all attributes in the product model are stored under *<product_model_name>/misc*. They all have the same syntax.

Attributes starting with the dot (.) character are system attributes. See “misc/sysattr.* (Attribute Definition)” on page 66.

User-defined attributes cannot have the dot character before their name. See “misc/userattr (User Attribute Definition)” on page 74.

Attribute Classes

At the highest level, attributes within the ODB++ format are divided into these classes: DFM Analysis, Product, and Process.



- **DFM Analysis** attributes — Used to enable or enhance analysis. An example is .center_fiducial for components that require a center fiducial.
- **Product** attributes — Used to describe the finished product. An example is .gold_plated on a copper features to indicate that gold will be delivered at these locations.
- **Process** attributes — Used to add further definition to the manufacturing process applied to the product. Process attributes are divided into process categories:
  - **Fabrication** — For example, the fabrication process attribute .plating_bar describes the manufacturing purpose for a piece of copper that is later removed by the milling/routing process and is not delivered to the customer as part of the final product.
  - **Assembly** — For example, the assembly process attribute .fs_direction_top is used to describe the flow solder assembly direction.

- **Test** — These attributes are for the functional, optical, and other testing of the PCBA product before, during or following the assembly process. An example would be the HP3070 attributes used to deliver programs for this test machine.
- **Generic** — The generic attributes related to the production process.

The attribute definitions for each attribute class are stored in a separate file.

In addition to the files that contain system attribute definitions, the file *userattr* can be used to define user attributes for all entities. It is stored in the library when the product model is created. It is read each time the product model is opened.

These files contain attribute definitions:

Attribute Definition File	Class	List of System Attributes in the Class
sysattr	Product	"Product (sysattr)" on page 238
sysattr.dfm	DFM analysis	"DFM (sysattr.dfm)" on page 242
sysattr.fab	Fabrication	"Fabrication (sysattr.fab)" on page 246
sysattr.assy	Assembly	"Assembly (sysattr.assy)" on page 248
sysattr.test	Test	"Test (sysattr.test)" on page 249
sysattr.gen	Generic	"Generic (sysattr.gen)" on page 250
userattr	(none)	(Contains user-defined attributes.)

Attribute Value Assignment

You can assign system attributes and user-defined attributes to ODB++ entities. For some entities, attributes are assigned using an *attrlist* file. For some entities, attributes are assigned using a lookup table.

Attrlist Files

For some entities, an *attrlist* file is placed under the entity, containing a list of attribute names and values. For example, this *attrlist* file assigns attributes to the layer named <*layer_name*>: <*product_model_name*>/steps/<*step_name*>/layers/<*layer_name*>/*attrlist:*

```
UNITS=MM
.out_mirror = no
.inp_file =
.eda_layers = "signal_2","signal","VIA"
.out_angle = 0.0
.out_polarity = positive
.out_x_scale = 1.000000
.out_y_scale = 1.000000
.out_comp = 0.000000
```

Lookup Tables

For entities for which multiple entities are defined in the same file—such as features, components, or nets—each entity definition line can contain attribute assignments for the entity being defined. Lookup tables are used to allow the entity assignment syntax to use shortcuts to refer to attributes and their values. This is an example of a feature attribute lookup table, and a feature definition record that assigns attributes by pointing to attributes and their values in the string `3=2,4=0:`

```
#
#Feature attribute names
#
@0 .smd
@1 .nomenclature
@2 .test_point
@3 .geometry
@4 .pad_usage
#
#Feature attribute text strings
#
&0 9796334
&1 fid_0_0_0
&2 moire
#
#Layer features
#
P -0.198 1.62 16 P 0 3;3=2,4=0;ID=123456
```

ODB++ Entities that can have Attributes Assigned

These ODB++ entities can have attributes assigned using lookup tables:

ODB++ Entity	File with Attribute Lookup Tables	Explanation
Symbol Feature	*/symbols/<symbol_name>/features*	"<symbol_name>/features (Symbol Features)" on page 43
Net	*/steps/<step_name>/eda/data*	"eda/data (EDA Data)" on page 105
Feature	*/steps/<step_name>/layers/<layer_name>/features*	"<layer_name>/features (Graphic Features)" on page 139
Component	*/steps/<step_name>/layers/<layer_name>/components*	"<layer_name>/components (Components)" on page 158

These ODB++ entities can have attributes assigned in an *attrlist* file:

ODB++ Entity	attrlist File	Explanation
Product Model	*<product_model_name>/misc/attrlist*	"misc/attrlist (Product Model Attribute List)" on page 56
Symbol	*/symbols/<symbol_name>/attrlist*	"<symbol_name>/attrlist (Symbol Attribute List)" on page 42
Wheel	*/wheels/<wheel_name>/attrlist*	"<wheel_name>/attrlist (Wheel Attribute List)" on page 74
Step	*/steps/<step_name>/attrlist*	"<step_name>/attrlist (Step Attribute List)" on page 85
Layer	*/steps/<step_name>/layers/ <layer_name>/attrlist*	"<layer_name>/attrlist (Layer Attribute List)" on page 127

Unique ID

The ODB++ format supports the permanent reference of entities through the use of indices for unique IDs. The unique ID is optional, but once assigned, is never changed, and no other entity within the product model can bear the same ID.

IDs are represented by a positive number between 0 and 4294967295.

Unique IDs are generated for use in the product model definition of these files: *matrix*, *stephdr*, *features*, *components*, and *profile*. The largest unique ID used in the product model is stored in the *info* file. See "misc/info (Basic Product Model Information)" on page 58.

The *stephdr*, *features*, *components*, and *profile* files can also have their own individual ID record to uniquely identify the file itself.

A feature file with an ID of 216 would look like this:

```
UNITS=INCH
ID=216
…..
#
#Feature symbol names
#
$0 r4
$1 r5
```

A component file with an ID of 843 would look like this:

```
UNITS=INCH
ID=843
#
#Component attribute names
```

```
#
@0 .comp_height
@1 .package_type
"
```

Format Definition

All files in the ODB++ format are ASCII files.

In all files, the hash (#) character specifies a comment. Lines that start with this character should be ignored, and are only used for readability.

The line separator can be either <LF> or <CR><LF>, depending on operating system and platform.

Extensible Content

Third parties are free to place additional content in the ODB++ file structure. Third parties are not constrained to using ASCII format, and are free to use encrypted or non-encrypted formats such as binary, ASCII, XML, and others. But that content is outside the format specification described for ODB++.

Large File Compression

One of the reasons vendors have chosen binary data in the past was the need to conserve disk space. Modern compression techniques provide excellent compression ratios, especially for ASCII files with repetitive patterns. Large files in ODB++ format are saved in standard UNIX compression format. The compression is optional. Software reading the ODB++ product model structure should expect some files to be compressed (*.Z*) and some not.

For each file described in this document, there is a table of information. Files that are potentially compressed are identified, in this document, in the Compression row of the file information table in the description of the file:

<file_name>	
Type	Line Record Text or Structured Text
Compression	None or Yes
Path	Path to the file.
UNITS	The UNITS directive is required or The UNITS directive is not relevant for the <*file_name*> file.

Structured Text Files

Many text files in ODB++ format contain expressions of the format:

<variable>=<value>

The main advantages of this structure is readability. The user can open a file and understand its contents without needing to refer to external sources.

A more elaborate structure, which appears in some structured text files, describes arrays. Arrays are lists of elements, each one containing several fields. The expression <variable>=<value> appears multiple times, each time defining an element of the array:

```
<array_name> {
 <variable>=<value>
 <variable>=<value>
 ....
}
```

For each file described in this document, there is a table of information. Structured text files are identified, in this document, in the Type row of the file information table in the description of the file:

<file_name>	
Type	Line Record Text or Structured Text
Compression	None or Yes
Path	Path to the file.
UNITS	The UNITS directive is required or The UNITS directive is not relevant for the <*file_name*> file.

Example of a <variable>=<value> expression (from the *stephdr* file):

```
X_DATUM=0.3
```

Example defining the LAYER array (from the *matrix* file):

```
LAYER {
   ROW=1
   CONTEXT=BOARD
   TYPE=COMPONENT
   NAME=COMP_+_TOP
   POLARITY=POSITIVE
   START_NAME=
   END_NAME=
   OLD_NAME=
   ADD_TYPE=MICRO_VIA
   COLOR=606090
}
```

Line Record Text Files

In some files, each line contains multiple fields, typically separated by space characters. Typically, the first character or word in each line defines the type of record that the line describes. In many cases, the line order is important.

The ODB++ format permits line length of up to 1,400 characters. A line longer than this can be truncated.

For each file described in this document, there is a table of information. Line record text files are identified, in this document, in the Type row of the file information table in the description of the file:

<file_name>	
Type	Line Record Text or Structured Text
Compression	None or Yes
Path	Path to the file.
UNITS	The UNITS directive is required or The UNITS directive is not relevant for the <*file_name*> file.

Example of a line record text file (from the feature file):

```
#
#Feature symbol names
#
$0 r50
$1 r70
$2 r80
$3 r93
$4 ths80x60x0x4x15
#
#Feature attribute names
#
@0 .geometry
@1 .pad_usage
#
#Feature attribute text strings
#
&0 systest_board
&1 term_1
&2 via_1
#
#Layer features
#
L -0.4 -3.6 -0.4 -4 0 P 0;0=0
L -0.4 -4 0 -4 0 P 0;0=0
L -0.4 -1.4 -0.4 -1 0 P 0;0=0
L -0.4 -1 0 -1 0 P 0;0=0
```

Coordinates

From ODB++ format v8.0, the UNITS directive must be present in every file that contains coordinates or measurements.

In ODB++ format v7.x, the units directive described the units applied in the feature file; and when not present, imperial units were assumed. Therefore, coordinates were expressed in inches; symbol sizes were expressed in mils.

When specifying a fractional coordinate, the separator for decimal numbers must be a period (.) For example, s3.25 2.456. No other representation of numbers is allowed (such as scientific notation, exponential, or others).

Angles

All angles are defined by a positive value between 0 and 360 degrees.

Rotation and Mirroring

These are the rules for rotation and mirroring.

- Feature pads can be rotated. Rotation is expressed in degrees or in multiples of 90 degrees, and is always clockwise.
- Mirroring occurs only along the x-axis (left to right, changing x coordinates).
- If both rotation and mirroring are performed, the feature is first rotated, and then mirrored.

Symbols

Symbols define a wide variety of shapes that are mostly used to represent pads.

Standard Symbols

Standard symbols are rendered dynamically from their names. They do not require that a graphic symbol entity be saved. They are round, square or parametric shapes.

The names of standard symbols have this structure:

<symbol_name_prefix>[<parameter_value>[x<parameter_value>[...]]]

where parameter_value is a decimal number, integer or real. Parameter values are delimited with an x character.

For example, a rounded square thermal is specified using a name like this:

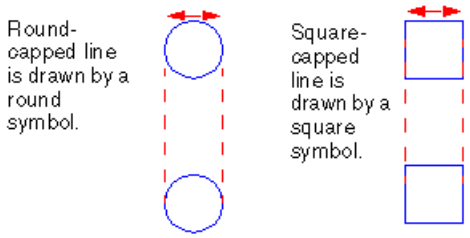
s_ths200x180x15x5x10xr20

- s_ths — Symbol name prefix.
- 200, 180, 15, 5, 10, r20 — Parameter values presented in the required order.

Note

When specifying a fractional symbol size, the separator for decimal numbers must be a period (.). For example, s3.25.

Symmetric symbols (round symbols, or square symbols where width=height) are required to create lines and arcs. The width of a line or arc is the width of the symmetric symbol used to draw it. See “Lines and Arcs Drawn with Symbols” on page 32.



Units for parameters that represent size are expressed as a thousandth part of an inch or as a micron, depending on the current UNITS directive. See “Units of Measurement” on page 18 for a discussion of the UNITS directive.

ODB++ format guarantees precision for standard symbol dimensions in non-integers with resolutions of up to 1/100 mils.

Symbol r0 is a legal entity. Its area is 0, and it is recommended to display it with a single display pixel.

Standard symbols are all positive filled shapes. Holes in symbols are see-through by definition. The internal implementation of complex symbols uses arc or contour data with cutouts.

User-Defined Symbols

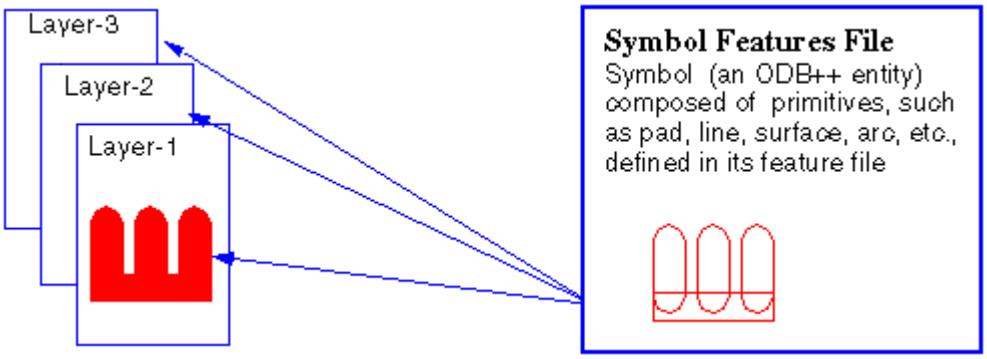
User-defined symbols have a full graphic description stored in *features* files in the product model *symbols* directory. They can contain any number of features, including any number of other user-defined symbols. Circular references are prohibited. User-defined symbols are created for a product model for shapes not found among the standard symbols.

A user-defined symbol is an ODB++ entity that is defined once and can be used many times in order not to repeat the definition of a group of features on a layer. A symbol definition contains a *features* file that specifies the primitive features (such as pad, line, arc, surface, and text) that comprise the symbol on the layer. See "<symbol_name>/features (Symbol Features)" on page 43.

Note

It is preferable to use a standard symbol, where possible. User-defined symbols require more memory than standard symbols and having a large number of them will slow down processing.

A symbol can be referenced from multiple layers in the product model, at different coordinates. Changing the symbol definition will automatically cause all its representations in the layers to change accordingly.



Standard symbol name prefixes are reserved for use in standard symbol names and cannot be assigned to user-defined symbols. For a list of these reserved names, see "Basic Standard Symbols" on page 169.

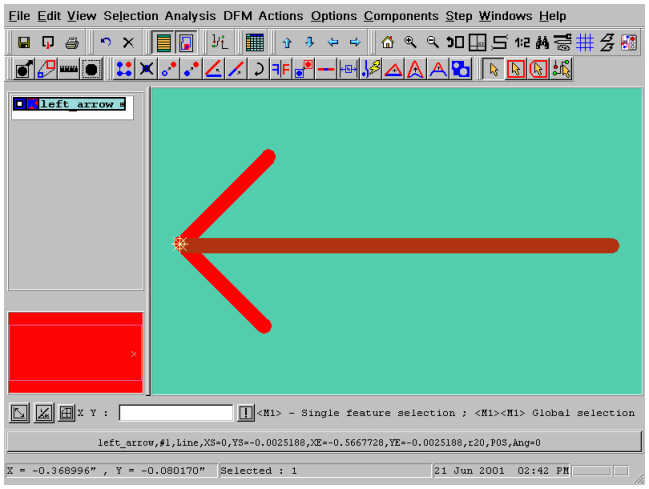
Note

Standard symbol name prefixes cannot be used for user-defined symbols even if the name of the user-defined symbol has a different format from that of the standard symbol, or if the user-defined symbol allows parameter values that are illegal in the standard symbol with that prefix.

Like standard symbols, user-defined symbols are scalable.

See "<layer_name>/features (Graphic Features)" on page 139 for an example of the UNITS directive in the *features* file.

The figure shows an arrow whose origin (0,0) is at the head of the arrow on the left.



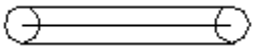
This user-defined symbol is drawn from lines (standard symbols). When inserting this symbol, the insertion point is at (0, 0).

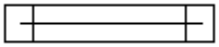
Lines and Arcs Drawn with Symbols

Lines and arcs are created by a symbol and two points—a start point and an end point. Lines and arcs are drawn by dragging the symbol by its center along a line or circular curve from one end to the other.

For arcs, only round symbols can be used.

If the symbol used to draw a line is symmetrical (square or round) the width of the generated line is the side or the diameter of the symbol.

A round symbol (whose name starts with r) generates round line ends  .

A square symbol (whose name starts with s) generates square line ends  .

Note

Using non-symmetric symbols (standard or user-defined) for the drawing of lines is not recommended, because their implementation tends to be inefficient. If such lines are found in a product model, the application must decide how to handle them. The result should resemble the sweep of the symbol from its start point to its end point.

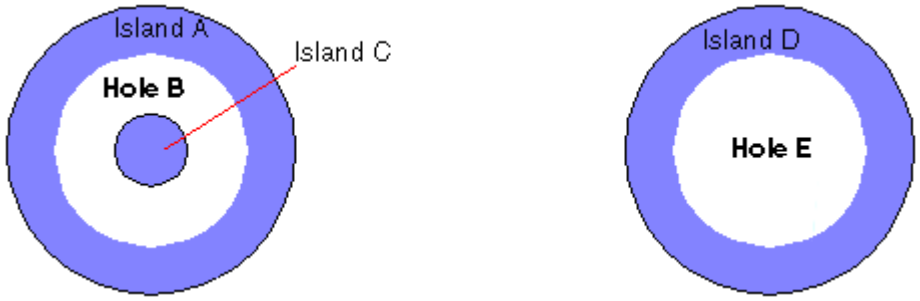
Surfaces

A surface is a feature that represents a legal contour having these properties:

- The islands of a contour do not overlap.

- Holes inside islands do not intersect.
- Islands and holes are ordered in a specific way.

  The order of containment of holes and islands within surfaces determines their natural order. The outermost island comes first. Islands precede holes that are contained in them. Holes precede islands that are contained in them. Take, for example, the following containment order:



  The natural order would be **A B C D E**. Island D is separate from Island A.

Polygons must meet these restrictions:

- Self-intersecting polygons (SIPs) are not allowed.
- Polygons, both the island and holes, must be sequenced to form a closed shape. That means that all polygons must contain at least three segments, and the last record point must be the same as the first.
- Edge lengths should be greater than zero.
- Holes must be graphically contained inside island polygons.
- The direction of segments representing the edges of an island must be clockwise, and the direction of segments representing the edges of holes must be counter-clockwise.
- The curves must be consistent—the start, end, and center point must construct a legal curve.

Geometric Entities

The ODB++ format uses geometric primitives called features to describe product model geometry. Features are stored in the feature containers of layers, profiles, and symbols. The types of geometric entities supported by the ODB++ format are the pad, line, arc, surface, and text.

These primitives are defined by the concept of a symbol and a contour.

- **Symbol** — A symbol is either one of the geometric primitives supported by the ODB++ format described in “Basic Standard Symbols” on page 169, or a combination of them.
- **Contour** — A contour is represented by single or multiple islands. An island is the area inside a boundary that consists of straight and circular segments. Islands in a contour do not touch or intersect with one another. Each island can contain holes that are also defined by a boundary. Those boundaries are referred to as polygons. The boundary of an island is oriented clockwise; the boundary of a hole is oriented counter-clockwise.

Features can be positive or negative. Positive features cover all underlying parts of features previously added to the feature container. Negative features erase all underlying parts of features previously added to the feature container.

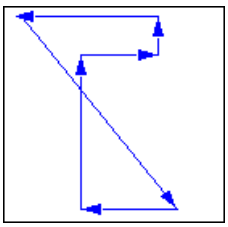
These geometric entities are supported by the ODB++ format:

- **Arc** — Created by dragging a symbol along a circular segment. Arcs are defined by their start point, end point, center, direction (clockwise or counter-clockwise), and symbol.

  When the start-point and end-point of an arc coincide, it is considered a 360 degree arc. There are no single-point arcs in the ODB++ format.
- **Surface** — A solid polygon that can contain non-intersecting islands, that themselves can contain any number of holes.
- **Text** — Consists of letters drawn with a font or barcode. Text is defined by a text string, font, and text size. See “fonts/standard (Standard Font)” on page 77 and “B - Barcode Records” on page 154.
- **Pad** — The most primitive geometric entity. A pad is a point made with a symbol. Pads are defined by their location and their symbol.
- **Line** — Created by dragging a symbol along a straight segment. Lines are defined by their start point, end point, and symbol.

Self-Intersecting Polygons (SIP)

In all polygon representations, a self-intersecting polygon (SIP) is illegal. A SIP is a polygon with a boundary that intersects itself at one or more points. Polygons can represent, for example, graphic surfaces or component package bodies.



Legal polygons are those whose edges intersect only at endpoints of consecutive edges.

SIPs impede the correct representation of required product model elements and hinder their mathematical representation. These are some examples of problematic operations:

- Correct rendering of layer graphics.
- Resizing (enlarging, shrinking, changing shape).
- Calculation of copper areas, where unambiguous definition of the copper location is essential.

Component Variant Implementation

The definition of component variants within ODB++ is based on the active BOM applied to a step (*stephdr*: AFFECTING_BOM) contained in the product model. The existence of multiple variants indicates which components should be used based on the currently applied BOM. The ODB++ format displays a variant by applying the attribute Not Populated per BOM (.no_pop) for components that are not to be used.

See “<step_name>/boms (Bill of Materials)” on page 99.

These attributes control the use of component variants:

Attribute Name	Internal name	Explanation
Product Model Variant List	.variant_list	This product model attribute contains a list of all variants available within the product model, using a colon (:) as a separator. For example, this value would represent three variants, build_1, build_2 and build_3: .variant_list=build_1:build_2:build_3

Attribute Name	Internal name	Explanation
Active Variant	.current_variant	This step attribute can contain one of the variant names listed in Product Model Variant List, if one has been selected. If no variant is currently applied, then its value would be empty.
Variant List	.comp_variant_list	When Active Variant is applied, this component attribute should already be applied to each component. This attribute contains the list of variants of which the component is a member. The list contains the names of all relevant variants, separated by colons (:). When the attribute Active Variant contains a variant name, components that are not members of that variant should bear the attribute Not Populated per BOM (.no_pop).

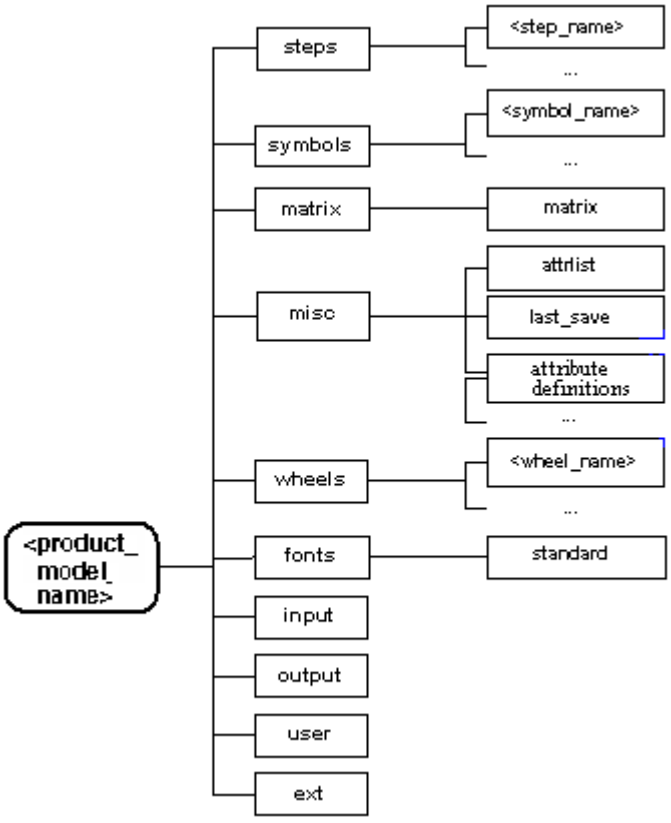
Chapter 3
Product Model Tree

An ODB++ product model is a directory containing sub-directories that represent ODB++ entities. These charts illustrate the directory hierarchy.

Users can create additional files in the *user* directory related to each entity as long as they do not conflict with ODB++ standard directories. Directories that are no longer supported might appear in ODB++ models containing legacy data.

<product_model_name>

This is the top level product model directory tree:



These sub directories of the <*product_model_name*> directory are described:

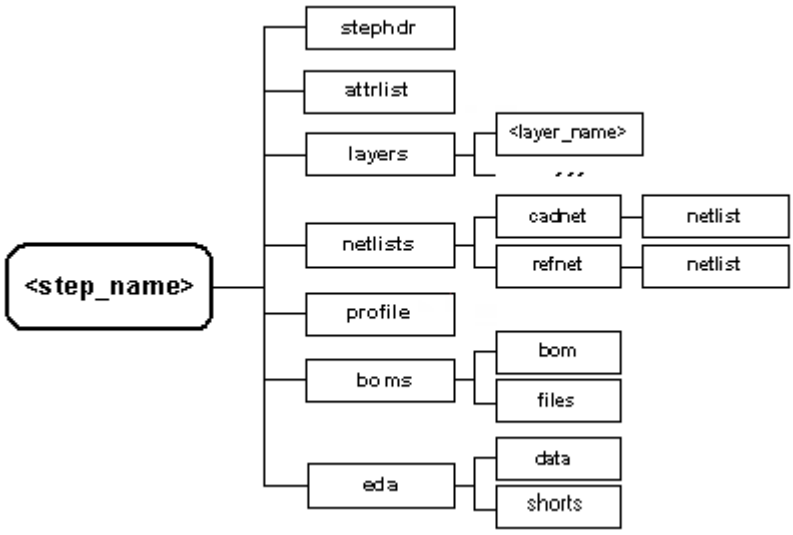
Sub Directory	File	Link to Explanation
steps		"<product_model_name>/steps" on page 38.
symbols	*attrlist* *features*	"<symbol_name>/attrlist (Symbol Attribute List)" on page 42 "<symbol_name>/features (Symbol Features)" on page 43
matrix	*matrix*	"matrix/matrix (Matrix)" on page 44
misc	*attrlist* *last_save* *info* *metadata.xml* *sysattr files* *userattr*	"misc/attrlist (Product Model Attribute List)" on page 56 "misc/last_save (Last Time Product Model Saved)" on page 57 "misc/info (Basic Product Model Information)" on page 58 "misc/metadata.xml (Metadata)" on page 60 "misc/sysattr.* (Attribute Definition)" on page 66 "misc/userattr (User Attribute Definition)" on page 74
wheels	*attrlist* *dcodes*	"<wheel_name>/attrlist (Wheel Attribute List)" on page 74 "<wheel_name>/dcodes (Dcodes)" on page 75
fonts	*standard*	"fonts/standard (Standard Font)" on page 77

<product_model_name>/steps

Each step entity can contain, in addition to general information and the list of layers, these subentities:

- Step and repeat information in the *stephdr* file — Specifying any steps that are included in this one and their relative location and orientation.
- Netlist description of the step — CAD netlist and reference netlist.
- An EDA object — This contains information about the component packages and pins. It also contains information about the relation of features in the board layers to specific design nets and properties imported from the EDA system. It can contain a definitions of intentional shorts.
- A profile — A border around the step representing the board edge.

This chart represents the directory tree under each step:

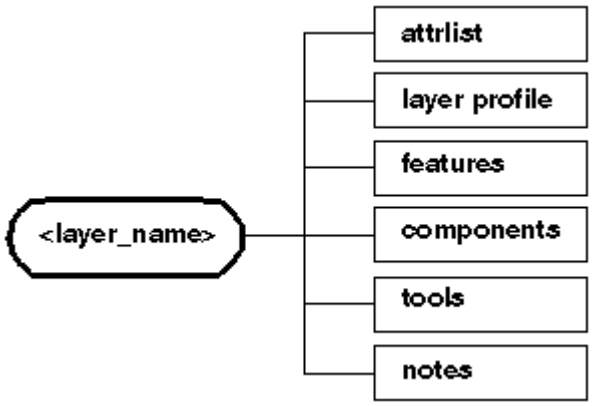


These files and sub directories of the <*step_name*> directory are described:

Sub Directory	File	Link to Explanation
	stephdr	"<step_name>/stephdr (Step Header)" on page 82
	attrlist	"<step_name>/attrlist (Step Attribute List)" on page 85
layers		See "Layer Entities" on page 127.
netlists	*cadnet* *refnet*	"cadnet/netlist (cadnet)" on page 86 "refnet/netlist (Reference)" on page 92
	profile	"<step_name>/profile (Outline Shape of Step)" on page 92
boms	*bom* *files*	"<bom_name>/bom (Bill Of Materials)" on page 99 "<bom_name>/files (Source Files)" on page 105
eda	*data* *shorts*	"eda/data (EDA Data)" on page 105 "eda/shorts (Electrical Intention Short Net)" on page 116

<product_model_name>/steps/<step_name>/layers

This chart represents the directory tree under each layer:



These files of the <*layer_name*> directory are described:

File	Link to Explanation
attrlist	"<layer_name>/attrlist (Layer Attribute List)" on page 127
profile	"<layer_name>/profile (Outline Shape of Layer)" on page 136
features	"<layer_name>/features (Graphic Features)" on page 139
components	"<layer_name>/components (Components)" on page 158
tools	"<layer_name>/tools (Drill Tools)" on page 165
notes	"<layer_name>/notes (Electronic Notes)" on page 167

Chapter 4
Product Model Entities

Several ODB++ entities are defined at the product model level.

For a list of mandatory files and directories, see “File System” on page 17.

steps

Steps are multi-layer entities such as a single image, an assembly panel, a fabrication panel, or a fabrication coupon. Each step contains a collection of layers. Layers are two-dimensional sheets, containing graphics, attributes and annotation. For information on the *steps* directory of the product model, see “Step Entities” on page 81.

symbols (User-Defined Symbols)

Symbols are single layer graphic entities that can be referenced from within any graphic layer in a step. ODB++ format can handle both standard symbols and user-defined symbols.

Standard symbols are rendered dynamically from their name. Standard symbols do not require any files. See “Standard Symbols” on page 29.

User-defined symbols require these files:

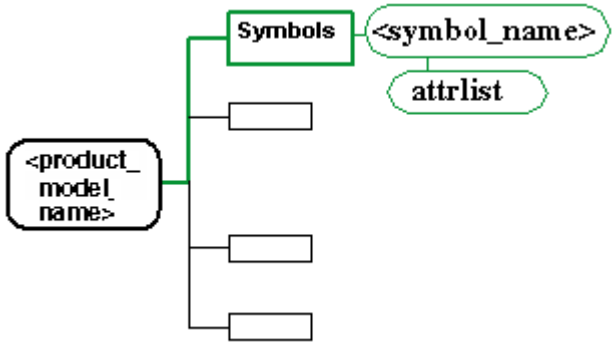
<symbol_name>/attrlist (Symbol Attribute List)

This file contains the values for system attributes and user-defined attributes assigned to a symbol.

You can provide *attrlist* files at these levels: product model, symbol, wheel, step, layer.

attrlist (Attribute List)	
Type	Structured Text
Compression	None
Path	*<product_model_name>/symbols/<symbol_name>/attrlist*
UNITS	The UNITS directive is required.

System attributes that can be placed in this file are listed with Entity = Symbol in the attribute description. See "System Attributes" on page 189.



The file contains lines of this form: <attribute> = <value>.

Example of an attrlist file for a Symbol

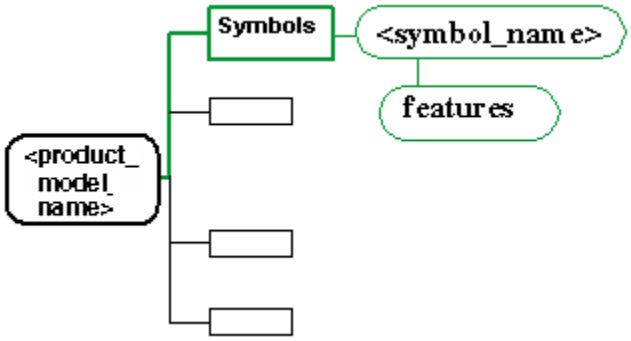
```
UNITS=INCH
.out_break = no
.out_scale = no
.break_away = no
.fill_dx = 0.100000
.fill_dy = 0.100000
.image_dx = -1.000000
.image_dy = -1.000000
connector = no
target = no
component =
comment =
hole_type = plated
serial_number = 15
```

<symbol_name>/features (Symbol Features)

Each user-defined symbol has a *features* file to define its shape.

features (Symbol Features)	
Type	Line Record Text
Compression	Yes
Path	*<product_model_name>/symbols/<symbol_name>/features*
UNITS	The UNITS directive is required.

The symbol *features* file describes the graphic shape of the symbol. It is similar in structure to the layer features file.



See "<layer_name>/features (Graphic Features)" on page 139.

Example

See the example in "<layer_name>/features (Graphic Features)" on page 139.

matrix (Product Model Matrix)

The *matrix* directory of the product model contains the *matrix* file.

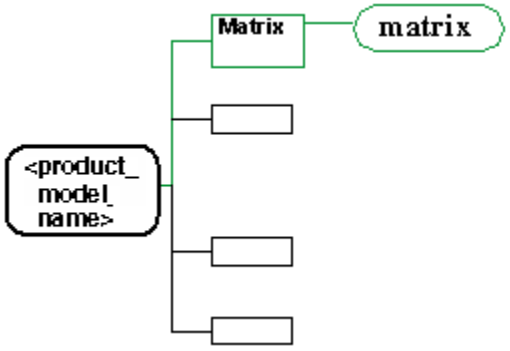
matrix/matrix (Matrix)

The matrix is a representation of the product model in which the rows are the product model layers—sheets on which elements are drawn for plotting, drilling and routing or assembly; and the columns are the product model steps—multi-layer entities such as single images, assembly panels, production panels and coupons.

matrix (Matrix)	
Type	Structured Text
Compression	None
Path	*<product_model_name>/matrix/matrix*
UNITS	The UNITS directive is not relevant for the *matrix* file.

Each row of the matrix contains additional information such as the type, polarity and context of the layer.

The matrix defines the physical order of the layers and the relation of drill layers (through, blind, buried, and so forth). A product model can contain only one matrix file.



Example of a Matrix File

```
STEP {
COL=1
ID=67890
NAME=PCB
}
STEP {
COL=2
NAME=PANEL
}
...
LAYER {
ROW=1
CONTEXT=BOARD
TYPE=COMPONENT
ID=123456
NAME=COMP_+_TOP
POLARITY=POSITIVE
START_NAME=
END_NAME=
OLD_NAME=
ADD_TYPE=
COLOR=606090
}
LAYER {
ROW=2
CONTEXT=BOARD
TYPE=SILK_SCREEN
ID=123457
NAME=SST
OLD_NAME =
POLARITY=POSITIVE
REF=123458
START_NAME=
END_NAME=
OLD_NAME=
ADD_TYPE=
COLOR=606090
}
...
LAYER {
    ROW=10
    CONTEXT=BOARD
    TYPE=SIGNAL
    ID=123458
    NAME=top
    POLARITY=POSITIVE
    FORM=RIGID
    START_NAME=
    END_NAME=
    OLD_NAME=
    ADD_TYPE=
    COLOR=0
}
LAYER {
    ROW=11
    CONTEXT=BOARD
    TYPE=DIELECTRIC
    ID=123459
    NAME=DIELECTRIC_4
    POLARITY=POSITIVE
    DIELECTRIC_TYPE=CORE
    DIELECTRIC_NAME=FR4 CORE Material
    FORM=RIGID
    CU_TOP=123458
    CU_BOTTOM=123460
    START_NAME=
    END_NAME=
    OLD_NAME=
    ADD_TYPE=
    COLOR=0
}
LAYER {
    ROW=12
    CONTEXT=BOARD
    TYPE=SIGNAL
    ID=123460
    NAME=signal_2
    POLARITY=POSITIVE
    FORM=RIGID
    START_NAME=
    END_NAME=
    OLD_NAME=
    ADD_TYPE=
    COLOR=0

}
...
LAYER {
ROW=16
CONTEXT=BOARD
TYPE=DRILL
NAME=DRILL_4_4
POLARITY=POSITIVE
START_NAME=sig4
END_NAME=sig5
OLD_NAME=
}
```

These topics are discussed:

Fields in the STEP Array of a Matrix File

These are the fields in the STEP array of the matrix file:

Field	Explanation
COL	The number of the column in the matrix. Column numbers must be unique positive numbers (1 and above). Gaps are allowed between columns.
ID	A unique ID for the step definition in the matrix (optional). See “Unique ID” on page 25.
NAME	A legal entity name for the step. Each named step must have a step entity defined under the *steps* directory of the product model. See “Legal Entity Names” on page 18.

Fields in the LAYER Array of a Matrix File

These are the fields in the LAYER array of the matrix file:

Field	Explanation
ROW	The number of the row in the matrix. Rows must be unique positive numbers (1 and above). Order is not important, and the numbering need not be sequential. Gaps are allowed between row numbers. See “Recommended Buildup” on page 50 for a suggested buildup of base type layers.
CONTEXT	The layer context must be one of these values: **BOARD** — A layer that participates in the actual board production. **MISC** — Any other layer that is used for drawings, testing, and so forth.

Field	Explanation
TYPE	The type of layer. For some layer types, sub-layer types can be supplied in the ADD_ TYPE field. See “Layer Subtypes” on page 51. The layer type must be one of these values: • **SIGNAL** — A layer used for regular signal transfer. • **POWER_GROUND** — A plane layer, used for power or ground signals. • **DIELECTRIC** — A dielectric layer that separates two copper layers. • **MIXED** — A combination of a signal and a plane layer. • **SOLDER_MASK** — A layer used for solder mask application. • **SOLDER_PASTE** — A layer used for depositing solder paste for assembly. • **SILK_SCREEN** — A layer used for application of text legend. • **DRILL** — A layer used to produce drill programs. Additional information pertaining to type of drill layer can optionally be applied to this base type. See “Layer Subtypes to Support Backdrill and Dual Diameter Drill” on page 51. • **ROUT** — A layer used to produce rout program. • **DOCUMENT** A layer used for drawings, testing, auxiliary processes, and so forth. • **COMPONENT** — A layer containing components locations and outlines. • **MASK** — A layer containing additional information used as the base type for all mask types. See “Layer Subtypes” on page 51. • **CONDUCTIVE_PASTE** — A layer used to represent the depositing of conductive materials.
ID	A unique ID for the layer definition in the matrix (optional). See “Unique ID” on page 25.
NAME	A legal entity name for the layer. Each named layer MUST have a layer entity defined under the layers directory of each step in the product model, otherwise the product model might be unreadable. See “Legal Entity Names” on page 18.
OLD_NAME	Previous name of the layer. When this field has a value it means that a matrix layer has been renamed and this value is its old name. If the field is blank it means the layer has not been renamed.

Field	Explanation
POLARITY	POSITIVE or NEGATIVE Controls whether the graphic representation of features is to be considered t be solid. In ODB++ format, for POSITIVE copper layers—like SIGNAL, POWER_GROUND, MIXED—the features represent copper; for SOLDER_PASTE layers, features represent paste. For NEGATIVE layers, features represent laminate. This is most commonly used on POWER_GROUND layers where pad features represent clearance, and background content is copper. One exception is the SOLDER_MASK and MASK layers when polarity is required to be POSITIVE and features represent clearances in the physical mask. COMPONENT, SILKSCREEN, DRILL, ROUT, and DOCUMENTATION layers should always have POSITIVE polarity. This polarity value in the matrix is in no way connected to the intended artwork generation polarity.
DIELECTRIC_ TYPE	Applies to layers of type DIELECTRIC to define the purpose behind the material. (optional) (default = NONE) • **NONE** — Not defined. • **PREPREG** — Material that is constructed with fiberglass impregnated with resin. The resin is pre-dried, but not hardened, and when heated, flows to adhere to neighboring material. • **CORE** — Material that is pre-pressed layers according to a pattern of copper foil - prepreg - copper foil.
DIELECTRIC_ NAME	A text field that is intended to carry the material name as reference in the design application. (optional)
FORM	Defines the layer as ridge or flexible. (optional) (default = RIGID) • **RIGID** — Layer is defined as rigid. • **FLEX** — Layer is defined as flexible. Layers of the subtype SIGNAL_FLEX, PG_FLEX and MIXED_ FLEX have a default FORM = FLEX. Layers of type DIELECTRIC are to be defined as RIGID or FLEX.
CU_TOP	Used with layer type DIELECTRIC to indicate that a copper board layer is to be manufactured on top of the DIELECTRIC material. The value is the UID of the layer being referenced. This field would be used as part of a DIELECTRIC_TYPE=CORE definition. (optional)
CU_BOTTOM	Used with layer type DIELECTRIC to indicate that a copper board layer is to be manufactured on bottom of the DIELECTRIC material. The value is the UID of the layer being referenced. This field would be used as part of a DIELECTRIC_TYPE=CORE definition. (optional)

Field	Explanation
REF	Establishes a connection for layers within the matrix for the layer types SOLDER_MASK, SOLDER_PASTE, SILK_SCREEN, MASK, or CONDUCTIVE_PASTE with any layer whose type is SIGNAL, POWER_GROUND, or MIXED. This relationship is established through the use of the REF directive, and is to contain the UID of the reference layers. Only layers of type SOLDER_MASK, SOLDER_PASTE, SILK_SCREEN, MASK, or CONDUCTIVE_PASTE can have this tag. For example, consider a build sequence where a solder mask is being applied to an inner layer of a board. Within ODB++ itself, the only relationship between a solder mask layer and the applied outer copper layer is the position of the solder mask in the matrix. If the solder mask layer was to be positioned within a matrix with copper above and below, either could be viewed as the recipient of the mask. The Related Layers (ref) tag would be used to establish the connection to the appropriate copper layer. (optional)
START_NAME, END_NAME	These fields are active only for drill and rout layers. They specify the span of the drill or rout, in case it is partial (for example, blind or buried via layers). Each field must be a valid board layer name. When the fields are empty, START_NAME is assumed to be the first board layer (that is not a drill or rout layer) and END_NAME is assumed to be the last board layer (that is not a drill or rout layer). The START_NAME and END_NAME are based on the name of the board layers in sequence from the top of the matrix to the bottom, regardless of actual drill direction. The layer attribute .drill_layer_direction is used to specify actual drill direction from the ‘top to bottom’ or ‘bottom to top’. Default drill direction is ‘top to bottom’. Therefore, blind drills or backdrills coming up from the bottom would actually start from within the matrix and end on the bottom, and the attribute .drill_layer_direction would be ‘bottom2top’. See “ODB++ System Attributes - Alphabetical List” on page 189.
ADD_TYPE	Contains the layer subtype name. (optional) For example: • PUNCH for layers with base type ROUT. • COVERLAY for layers with base type SOLDER_MASK. • BACKDRILL for layers with base type DRILL. • SILVER_MASK for layers with base type CONDUCTIVE_PASTE. • WIRE_BONDING for layers with base type MASK. See “Layer Subtypes” on page 51.

Field	Explanation
COLOR	RGB defines the recommended color for display of this layer in the layer list. The representation is created using a percentage of red, green and blue. The syntax is **COLOR=RRGGBB** where **RR**, **GG** and **BB** are numbers between 00-99. For example, **COLOR=304050**. Default=0, indicating no preference. (optional)

Recommended Buildup

The ROW element represents the position of the layers in the buildup of the board.

Mask and dielectric layers related to any of the base layers should be placed above and below the relevant layers, as in the actual buildup.

This is the recommended buildup of base type layers:

Table 4-1. Recommended Buildup for Layer Matrix

Layer Type	Recommended Buildup
physical layers	• component (top) • solder paste (top) • silk screen (top) • solder mask (top) • all copper layers as ordered in the step • solder mask (bottom) • silk screen (bottom) • solder paste (bottom) • component (bottom)
drill layers	Drill layers according to top to bottom sequence and depth. For example: • Drill penetrating the top layer and extending to the greatest depth. • Drill penetrating the top layer and extending to the second greatest depth, and so forth, until all drills from the top layer are represented. • Drill beginning at the second layer and extending to the greatest depth. • Drill beginning at the second layer and extending to the second greatest depth, and so forth, until all drills from the second layer are represented. • Continuing until all drills are represented by a layer according to the initial point of penetration and the depth.
rout layer	Rout layer.
documentation, misc layers	Documentation or Misc layer.

Layer Subtypes

For some of the base layer types assigned in the TYPE field of the LAYER array of the *matrix* file, you can assign a layer subtype in the ADD_LAYER field.

These types of layer subtypes are discussed:

- Layer Subtypes for General Needs
- Layer Subtypes to Support Backdrill and Dual Diameter Drill
- Layer Subtypes to Support Embedded Resistors and Capacitors
- Layer Subtypes to Support Flex/Rigid Flex Manufacturing
- Layer Subtype to Support Wire Bonding

Layer Subtypes for General Needs

These layer subtypes support general needs in the representation of product models:

Layer Subtype ADD_LAYER	**Base Type TYPE**	**Description**
PUNCH	ROUT	Pattern to be punched by a die-cut fixture.

Layer Subtypes to Support Backdrill and Dual Diameter Drill

These layer subtypes support the possible unique needs of drill representation in product models. They are subtypes of the DRILL base type:

Layer Subtype ADD_LAYER	**Base Type TYPE**	**Description**
BACKDRILL	DRILL	Represents a backdrill procedure where the plating within a hole is removed from the unused portion of the matrix using a drill hole slightly larger than the original. This removes the connection from one side of the PCB.
DUAL_DIAMETER	DRILL	Represents a condition prior to plating where a slightly larger hole is drilled part way through the matrix at the same location as an existing hole.

Layer Subtypes to Support Embedded Resistors and Capacitors

These layer types support the unique needs of product models containing embedded resistors and embedded capacitors. They are layer subtypes of the MASK base type:

Layer Subtype **ADD_LAYER**	**Base Type** **TYPE**	**Description**
EMBEDDED_R	MASK	Feature shapes on the layer represent embedded resistors.
EMBEDDED_C	MASK	Feature shapes on the layer represent embedded capacitors.

Layer Subtypes to Support Flex/Rigid Flex Manufacturing

These layer types support the unique needs of Flex/Rigid Flex product models. They are subtypes of the indicated base type:

Layer Subtype **ADD_LAYER**	**Base Type** **TYPE**	**Description**
COVERLAY	SOLDER_MASK	Clearances of a coverlay layer.
COVERCOAT	SOLDER_MASK	Clearances of a covercoat layer.
STIFFENER	MASK	Shapes and locations where stiffener material is placed on the PCB.
BEND_AREA	MASK	For labeling areas on the PCB bent when the PCB is in use.
PSA	MASK	Shapes and locations where PSA (Pressure Sensitive Adhesive) material is placed on the PCB.
AREA	DOCUMENT	Area definition.
SIGNAL_FLEX	SIGNAL	Signal (copper) layer on flex laminate. Used to distinguish from signal on rigid laminate in rigid-flex boards.
PG_FLEX	POWER_GROUND	Power and ground (copper) layer on flex laminate. Used to distinguish from power and ground layer on rigid laminate in rigid-flex boards.

Layer Subtype **ADD_LAYER**	**Base Type** **TYPE**	**Description**
MIXED_FLEX	MIXED	Mixed copper layer—a layer containing both power/ground planes and regular signal pads or traces—on flex laminate. Used to distinguish from a Mixed copper layer on rigid laminate in a rigid-flex board.
DRAWING	DOCUMENT	Drawing layer definition.
PLATING_MASK	MASK	Defines which features in the adjacent copper layer should be plated.
IMMERSION_MASK	MASK	Defines which features are to be covered during immersion in the gold process.
OSP_MASK	MASK	Defines which features are to be covered with **OSP** finish.
SILVER_MASK	CONDUCTIVE_PASTE	Defines the silver mask of the adjacent copper layer.
CARBON_MASK	CONDUCTIVE_PASTE	Defines the location where carbon ink is applied to copper layers.

Layer Subtype to Support Wire Bonding

This layer type supports wire bonding. It is a subtypes of the MASK base type:

Layer Subtype **ADD_LAYER**	**Base Type** **TYPE**	**Description**
WIRE_BONDING	MASK	Wire bond connections are stored in separate layers for connections at the top and bottom sides, named wire_bond_+_top and wire_bond_+_bot. In a wire bond layer, die bumps (pins of source components) are represented by pads bearing attributes .net_name, .comp_name, and .pin_name (used as a link to the netlist file). Wire bonds are represented by lines bearing attribute .bonding_profile. The value of this attribute specifies the bonding order. Direct connection of die bump to copper pad (for flip chip) is represented by lines of zero length.

Example of a Matrix File Defining a Wire Bond Layer

These matrix file lines define a layer containing wire bond connections.

```
LAYER {
    ROW=1
    CONTEXT=BOARD
    TYPE=MASK
    NAME=WIRE_BOND_+_TOP
    POLARITY=POSITIVE
    START_NAME=
    END_NAME=
    OLD_NAME=
    ADD_TYPE=WIRE_BONDING
    COLOR=9999
    ID=6190
}
```

Example of a Matrix File Defining Layer Subtypes for Flex/Rigid Flex Manufacturing

These matrix file lines define layers with layer subtypes SILVER_MASK, COVERLAY, PG_FLEX, MIXED_FLEX, AREA, and PUNCH:

```
LAYER {
     ROW=2
     CONTEXT=BOARD
     TYPE=MASK
     NAME=SILVER_TOP
     POLARITY=POSITIVE
     START_NAME=
     END_NAME=
     OLD_NAME=MODPNL.46
     ADD_TYPE=SILVER_MASK
     COLOR=909098
}
LAYER {
     ROW=3
     CONTEXT=BOARD
     TYPE=SOLDER_MASK
     NAME=CVL1
     POLARITY=POSITIVE
     START_NAME=
     END_NAME=
     OLD_NAME=
     ADD_TYPE=COVERLAY
     COLOR=208020
}
LAYER {
     ROW=9
     CONTEXT=BOARD
     TYPE=POWER_GROUND
     NAME=L5
     POLARITY=POSITIVE
     START_NAME=
     END_NAME=
     OLD_NAME=PG5
     ADD_TYPE=PG_FLEX
     COLOR=856750
}
LAYER {
     ROW=10
     CONTEXT=BOARD
     TYPE=MIXED
     NAME=L6
     POLARITY=POSITIVE
     START_NAME=
     END_NAME=
     OLD_NAME=MODPNL.06
     ADD_TYPE=MIXED_FLEX
     COLOR=948775
}
```

```
LAYER {
     ROW=21
     CONTEXT=BOARD
     TYPE=DOCUMENT
     NAME=STIFFENER1_AREA
     POLARITY=POSITIVE
     START_NAME=
     END_NAME=
     OLD_NAME=
     ADD_TYPE=AREA
     COLOR=994171
}
LAYER {
     ROW=10
     CONTEXT=MISC
     TYPE=ROUT
     NAME=ADH-PUNCH_L2-L3
     POLARITY=POSITIVE
     START_NAME=
     END_NAME=
     OLD_NAME=
     ADD_TYPE=PUNCH
     COLOR=757575
}
```

misc (Miscellaneous)

These files relate to the misc entity:

misc/attrlist (Product Model Attribute List)

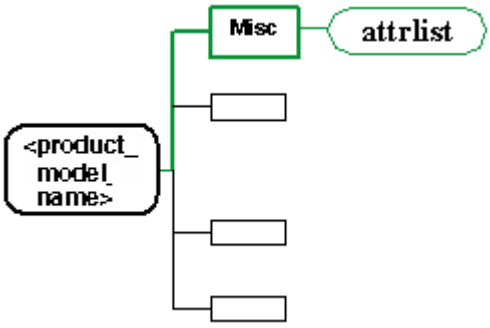
This file contains the values for system attributes and user-defined attributes assigned to a product model. Only attributes that have been defined are stored in the product model.

You can provide *attrlist* files at these levels: product model, symbol, wheel, step, layer.

attrlist (Attributes Used in Product Model)	
Type	Structured Text
Compression	None

attrlist (Attributes Used in Product Model)	
Path	*<product_model_name>/misc/attrlist*
UNITS	The UNITS directive is required.

System attributes that can be placed in this file are listed with Entity = Product Model in the attribute description. See “System Attributes” on page 189.



The file contains lines of this form: <attribute> = <value>

Example of a product model attrlist file

System attribute names begin with a dot. The attribute connector in this example is a user-defined attribute, and therefore does not begin with a dot. For a list of system attributes defined with Entity = Product Model, see “System Attributes” on page 189.

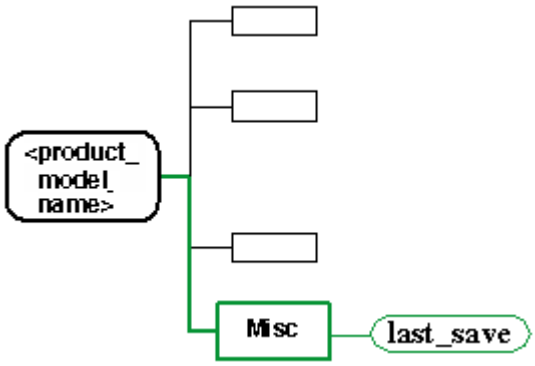
```
UNITS=INCH
.customer = abc
connector = no
target = no
```

misc/last_save (Last Time Product Model Saved)

This file is optional. It is written each time a save operation is performed on a product model. It records the time of the save operation.

last_save (Last Time Product Model Saved)	
Type	Line Record Text
Compression	None
Path	*<product_model_name>/misc/last_save*
UNITS	The UNITS directive is not relevant for the *last_save* file.

The file consists of a single line that contains the timestamp in format **yyyymmdd.hhmmss**.



Example of the last_save file

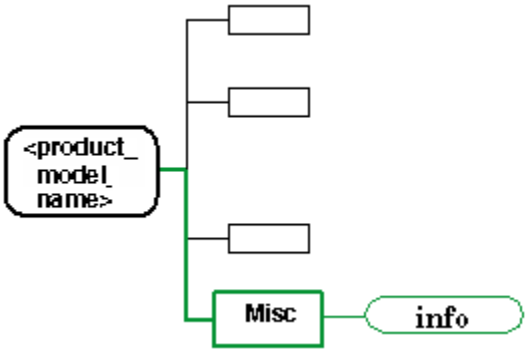
```
20141224.183210
```

misc/info (Basic Product Model Information)

This file records basic information about the product model. It is written each time a save operation is performed.

info (Basic Product Model Information)	
Type	Line Record Text
Compression	None
Path	*<product_model_name>/misc/info*
UNITS	The UNITS directive is required.

The UNITS directive in this file governs all product model units unless specifically overridden within the create file or element declaration. See “Units of Measurement” on page 18.



This information is stored in the *info* file:

Field	Explanation
PRODUCT_MODEL_NAME	Legal name of the product model.
ODB_VERSION_MAJOR	Major version designation such as 8 in Version 8.0, expressed as a numeric string. This can provide the application with a way to verify compatibility.
ODB_VERSION_MINOR	Minor version designation such as 0 in Version 8.0, expressed as a numeric string.
ODB_SOURCE	Source of data, typically a CAD/EDA system name of up to 500 characters.
CREATION_DATE	Creation date in the format yyyymmdd.hhmmss.
SAVE_DATE	Save date in the format yyyymmdd.hhmmss.
SAVE_APP	Name and number of the application in which the product model was saved, along with the currently running software version in up to 100 characters.
SAVE_USER	Login name of the user saving the file, in up to 50 characters (optional).
UNITS	Units used by default for this product model. **(UNITS=MM|INCH).**
MAX_UID	The largest numerical unique ID in the product model (optional). See “Unique ID” on page 25.

Example of the info file

```
UNITS=MM
PRODUCT_MODEL_NAME=k10025_cd2
ODB_VERSION_MAJOR=8
ODB_VERSION_MINOR=0
ODB_SOURCE=Expedition
CREATION_DATE=20110727.091213
SAVE_DATE=20110727.091230
SAVE_APP=Valor NPI 9.2
SAVE_USER=mikel
MAX_UID=57394
```

misc/metadata.xml (Metadata)

Product description information is provided in a file named *metadata.xml*, stored in the *misc* directory of the product model.

metadata (Metadata)	
Type	XML
Compression	None
Path	*<product_model_name>/misc/metadata.xml*
UNITS	The UNITS directive is not relevant for the *metadata.xml* file.

The purpose of metadata is to support manufacturing process-cost calculation, and the generation of the Manufacturing Product Definition (MPD) during the early stage of the New Product Introduction (NPI) process. This data is made available immediately, without needing to process the ODB++ product model.

The file is not case sensitive.

See “Metadata Example” on page 277.

Data is added to the metadata as the product model moves through the flow processes.

Structure of the Metadata File

A metadata file can contain several types of element:

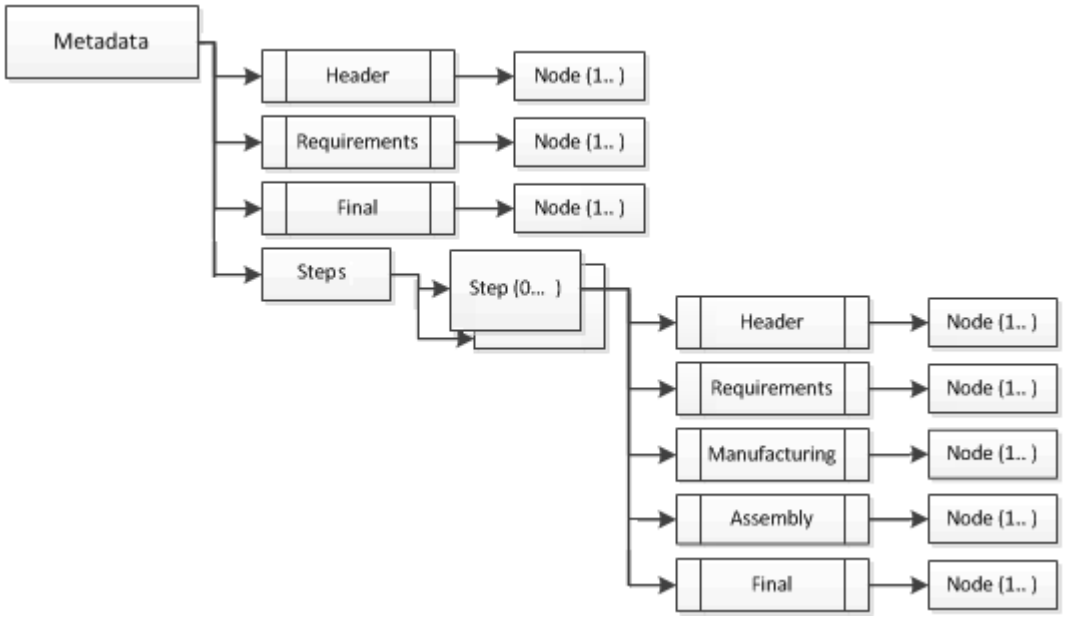
Element	Explanation
metadata section	The metadata is divided into sections that indicate the type of metadata: • Header • Requirements • Manufacturing • Assembly • Final Metadata that applies to the product model is placed in metadata sections at the top level of the hierarchy. Metadata that applies to a specific step is placed in metadata sections that are child elements of a step section. See “Metadata Sections” on page 62.

steps section	The steps section contains one or more step sections.
	This is the syntax of the steps element:
	`<Steps description = “Steps”>`
	`</Steps>`
	The description is optional.

step section	The start-tag of each step section indicates the name of the step. Between the start-tag of the step and its end-tag, are metadata sections containing metadata that applies to that step.
	This is the syntax of the step element:
	`<step name="nnn" description="ddd" >`
	`</step>`
	The description is optional.

node	These elements contain the metadata. Each node element must be placed in the appropriate metadata section. See “Metadata Types” on page 63. See “Attributes of a Metadata Node” on page 63.
classify_by	Some nodes can contain be classified by one or more levels of classes. See “Collective Data” on page 64.

This illustrates the structure of a metadata file:



Metadata Sections

The metadata is divided into sections that indicate the type of metadata.

Some sections can be specified at the level of the product model, and some sections can be specified only in a step section of the file.

The information that can be specified in each section is listed in "Metadata Content" on page 265. For each node, there is an indication of whether the node can be specified at the product model level or at the step level:

Section	Explanation	Level	Content
Header	General information about the product or the step.	product model or step	"Header Metadata Section" on page 265
Requirements	Overall requirements specified by the designer.	product model or step	"Requirements Metadata Section" on page 267
Manufacturing	Manufacturing information.	step	"Manufacturing Metadata Section" on page 271
Assembly	Assembly information.	step	"Assembly Metadata Section" on page 276

Section	Explanation	Level	Content
Final	Quality assurance and delivery information.	product model or step	"Final Metadata Section" on page 277

The metadata section elements have this syntax. The description attribute is optional.

Element	Syntax
Header	`<header description="ddd" /> </header>`
Requirements	`<requirements description="ddd" /> </requirements>`
Manufacturing	`<manufacturing description="ddd" /> </manufacturer>`
Assembly	`<assembly description="ddd" /> </assembly>`
Final	`<final description="ddd" /> </final>`

Metadata Types

Each value stored in the metadata file is of one of these data types:

Type	Description
Boolean	A value of either yes or no.
Integer	A value expressed as an integer.
Float	A value expressed as a decimal number. A node of this type can be of one of these unit types: • **distance** — Represents distance. • **size** — Represents the size of a feature. • **temperature** — Represents temperature. • **area** — Represents area.
Text	Text strings of printable ASCII characters.
Option	One of the legal values.
TextWithOptions	One of the suggested values or another text string.

Attributes of a Metadata Node

Matadata is specified in node statements having this syntax:

```
<node name="nnn", value="vvv", units="uuu", layer="lll", display="ddd" />
```

See “Metadata Content, Example, and Schema” on page 265.

These are the attributes of the node element:

Attribute	Description	Legal Values	Required
name	Name of the information.	A name in “Metadata Content” on page 265.	required
value	A value stored for future reference by its metadata content name.	Legal values based on the type of content stored.	
units	Units of measure. Permitted values for units correspond to those defined in “Units of Measurement” on page 18.	**distance** — inch or mm **size** — inch or mm **temperature** — fahrenheit or centigrade **area** — square mm or square inch	Required only for float data.
layer	The name of the layer to which the name/value pair is assigned.	A layer name located in the ODB++ matrix file.	optional
display	Suggested text for the application display.	ASCII text characters.	optional

Collective Data

In some cases, metadata content can represent a collection of similar data—for example, the number of components classified by type. In such a case, classify_by elements are included in the file, with this syntax:

```
<classify_by name="nnn", value="vvv", units="uuu", layer="lll", display="ddd"
/>
```

Classification is recursive, and can include an additional level of classification. component_ count can be classified by two parameters: layer and technology. The corresponding segment of the metadata file looks like this:

```
<node name=component_count value= 30 display=Component Count >
  <classify_by name=top value=10 display="Components in Top Layer:" >
    <classify_by name=bga value=5 />
    <classify_by name=axial value=5 />
  </classify_by>
  <classify_by name=Layer3 value=10 display="Components in Inner Layer 3" >
    <classify_by name=bga value=5 />
    <classify_by name=axial value=5 />
  </classify_by>
  <classify_by name=bottom value=10 display="Components in Bottom Layer:" >
    <classify_by name=bga value=5 />
    <classify_by name=axial value=5 />
  </classify_by>
</node>
```

This is an example report that could be generated from the component_count segment.

	Layer	**Technology**	**Value**
Component Count	Total for all layers		30
Components in layer	Top Layer		10
		bga axial	5 5
Components in layer	Inner Layer 3		10
		bga axial	5 5
Components in layer	Bottom Layer		10
		bga axial	5 5

This metadata content requires classification:

Name	**Display Name**	**Classified by**
component_count	Components per Step	layer, technology
number_of_nets_with_probes	Number of Nets with Probes	layer, size
number_of_fiducials	Number of Fiducials	layer
total_number_of_smd_pads	Total Number of SMD Pads	layer
number_of_test_points	Number of Test Points	side

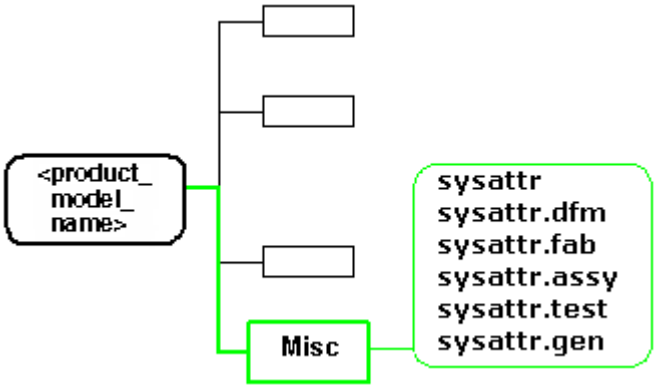
Name	Display Name	Classified by
number_of_drill_holes	Number of Drill Holes	layer
number_of_tooling_holes	Number of Tooling Holes	layer
number_of_un-connected_ nets	Number of Unconnected Nets	layer
number_of_single_pin_nets	Number of Single Pin Nets	layer
number_of_multi_pin_nets	Number of Multi-pin Nets	layer
cad_package_analysis	CAD Package Analysis	layer, CAD package, pin_ count and pitch

misc/sysattr.* (Attribute Definition)

Attribute definition files for all attributes in the product model are stored under *<product_ model_name>/misc*. They all have the same syntax. Attributes starting with the dot (.) character are system attributes (some system attributes must be defined for certain processes, such as analysis). User-defined attributes cannot have the dot character before their name.

sysattr, sysattr.dfm, sysattr.fab, sysattr.assy, sysattr.test, sysattr.gen	
Type	Line Record Text
Compression	None
Path	*<product_model_name>/misc/sysattr* *<product_model_name>/misc/sysattr.dfm* *<product_model_name>/misc/sysattr.fab* *<product_model_name>/misc/sysattr.assy* *<product_model_name>/misc/sysattr.test* *<product_model_name>/misc/sysattr.gen*
UNITS	The UNITS directive is required.

Every attribute definition file contains a list of definitions, defined at the time the product model was created. It is read each time the product model is opened.



For a discussion of ODB++ attributes, see “ODB++ Attributes” on page 22.

For a discussion of user defined attributes, see “misc/userattr (User Attribute Definition)” on page 74.

These topics are discussed:

Attribute Value Defaults

When defining an attribute, it is possible to establish a default value by using the DEF keyword. The purpose of this keyword is to suggest a default value to be applied to an element without further intervention. If the DEF keyword is missing, or not defined, these are the suggested defaults:

Table 4-2. Attribute Type Default Values

Attribute Type	Default
BOOLEAN	NO
TEXT	empty string
OPTION	first value in **OPTIONS** list
INTEGER	MIN_VAL
FLOAT	MIN_VAL

Specifying Units for a FLOAT Attribute

For an attribute of type FLOAT, you can use the UNITS line to specify the units in which the values of MIN_VAL and MAX_VAL are given. See "FLOAT Attribute Definition" on page 72.

The way in which the UNITS line of the attribute is interpreted depends on the UNITS directive at the top of the file.

Example: UNITS directive = MM and UNITS line = MIL_MICRON

In this example, MAX_VAL is interpreted as 100 micron. The directive UNITS=MM specifies that measurements in the file are given in metric units. The attribute definition contains the line UNITS=MIL_MICRON, which specifies that if units of measurement are metric, the measurements are to be interpreted as microns:

```
UNITS=MM

...

FLOAT {
NAME=.spacing_reg
ENTITY=FEATURE
MIN_VAL=0.0
MAX_VAL=100.0
UNITS=MIL_MICRON
GROUP=DFx Feature
PROMPT=Spacing Required
DEF=25.0
}
```

Example: UNITS directive = INCH and UNITS line = MIL_MICRON

In this example, MAX_VAL is interpreted as 100 mil (0.1 inch). The directive UNITS=INCH specifies that measurements in the file are given in imperial units. The attribute definition contains the line UNITS=MIL_MICRON, which specifies that if units of measurement are imperial, the measurements are to be interpreted as mils:

```
UNITS=INCH

...

FLOAT {
NAME=.spacing_reg
ENTITY=FEATURE
MIN_VAL=0.0
MAX_VAL=100.0
UNITS=MIL_MICRON
GROUP=DFx Feature
PROMPT=Spacing Required
DEF=0.025
}
```

Attribute Definitions

Each attribute is defined by a set of definitions specific to the attribute type:

Attribute Type	Explanation	Attribute Definition
BOOLEAN	Attributes that have values of either YES or NO.	"BOOLEAN Attribute Definition" on page 69
TEXT	Attributes that are text strings have values comprised of printable ASCII characters, and with a length range specified by the attribute definition.	"TEXT Attribute Definition" on page 70
OPTION	Attributes that have values limited to a list of options supplied as part of the attribute definition.	"OPTION Attribute Definition" on page 71
INTEGER	Attributes that have a numerical integer value within the range defined by the attribute definition.	"INTEGER Attribute Definition" on page 72
FLOAT	Attributes that have numeric values that can also contain decimal notation within the range defined in the attribute definition.	"FLOAT Attribute Definition" on page 72

BOOLEAN Attribute Definition

Fields for attributes of type BOOLEAN:

Field	Explanation
NAME	Name of the attribute.
PROMPT	The name used on screen when this attribute is displayed (Optional).
ENTITY	The entities for which this attribute is applicable. A semicolon (;) separated list of entity types: JOB, STEP, SYMBOL, LAYER, BUILDUP, WHEEL, FEATURE, COMPONENT, or PACKAGE.
GROUP	Name of the class to which this attribute belongs. This could be DFM Analysis, Product, Process: Fabrication, Assembly, Test, Generic, or other. See "Attribute Classes" on page 22.
DEF	The default value of the attribute, if no value is specified. See "Attribute Value Defaults" on page 67.

Example

```
BOOLEAN
{
  NAME    = .string_mirrored
  ENTITY  = FEATURE
  DEF     = NO
  GROUP   = Fabrication
  PROMPT  = String Mirror
}
```

TEXT Attribute Definition

Fields for attributes of type TEXT:

Field	Explanation
NAME	Name of the attribute.
PROMPT	The name used on screen when this attribute is displayed (Optional).
OPTIONS	A semicolon (;) separated list of options. Unlike other Options lists, the text attribute is used to provide a selection of the most commonly used values.
MIN_LEN	Minimum number of characters allowed in the value of the attribute.
MAX_LEN	Maximum number of characters allowed in the value of the attribute.
ENTITY	The entities for which this attribute is applicable. A semicolon (;) separated list of entity types: **JOB**, **STEP**, **SYMBOL**, **LAYER**, **BUILDUP**, **WHEEL**, **FEATURE**, **COMPONENT**, or **PACKAGE**.
GROUP	Name of the class to which this attribute belongs. This could be DFM Analysis, Product, Process: Fabrication, Assembly, Test, Generic, or other. See “Attribute Classes” on page 22.
DEF	The default value of the attribute, if no value is specified. See “Attribute Value Defaults” on page 67.

Example

```
TEXT
{
  NAME    = .string
  OPTIONS =
  ENTITY  = FEATURE
  MIN_LEN = 0
  MAX_LEN = 1000
  GROUP   = Feature
  DEF =
  PROMPT  = Copper Text String
}
```

OPTION Attribute Definition

Fields for a structure of type OPTION:

Field	Explanation
NAME	Name of the attribute.
PROMPT	The name used on screen when this attribute is displayed (Optional).
OPTIONS	A semicolon (;) separated list of options.
DELETED	A semicolon (;) separated list of the values YES and NO. Only an index number to the list of options in the OPTIONS field is stored within the format. Therefore, the removal of an existing option is not allowed. The list of corresponding YES or NO values indicates whether an option is available for use. A value of YES indicates that the corresponding OPTIONS value is to be ignored.
ENTITY	The entities for which this attribute is applicable. A semicolon (;) separated list of entity types: **JOB**, **STEP**, **SYMBOL**, **LAYER**, **BUILDUP**, **WHEEL**, **FEATURE**, **COMPONENT**, or **PACKAGE**.
GROUP	Name of the class to which this attribute belongs. This could be DFM Analysis, Product, Process: Fabrication, Assembly, Test, Generic, or other. See "Attribute Classes" on page 22.
DEF	The default value of the attribute, if no value is specified. See "Attribute Value Defaults" on page 67.

Example

In this example, option both is defined as being logically deleted.

```
OPTION {
    NAME=.viacap_layer
    PROMPT=Via Capping on Layer
    OPTIONS=top;bottom;both;none
    DELETED=NO;NO;YES;NO
    ENTITY=step
    DEF=none
    GROUP=Custom
}
```

INTEGER Attribute Definition

Fields for a structure of type INTEGER:

Field	Explanation
NAME	Name of the attribute.
PROMPT	The name used on screen when this attribute is displayed (Optional).
MIN_VAL	Minimum value for the integer attribute.
MAX_VAL	Maximum value for the integer attribute.
ENTITY	The entities for which this attribute is applicable. A semicolon (;) separated list of entity types of: JOB, STEP, SYMBOL, LAYER, BUILDUP, WHEEL, FEATURE, COMPONENT, or PACKAGE.
GROUP	Name of the class to which this attribute belongs. This could be DFM Analysis, Product, Process: Fabrication, Assembly, Test, Generic, or other. See “Attribute Classes” on page 22.
DEF	The default value of the attribute, if no value is specified. See “Attribute Value Defaults” on page 67.

Example

```
INTEGER {
  NAME     = .num_local_fiducials
  ENTITY   = COMPONENT
  DEF      = 0
  MIN_VAL  = 0
  MAX_VAL  = 20
  GROUP    = DFx Component
  PROMPT   = Number Local Fiducials Required
}
```

FLOAT Attribute Definition

Fields for a structure of type FLOAT:

Field	Explanation
NAME	Name of the attribute.
PROMPT	The name used on screen when this attribute is displayed (Optional).
MIN_VAL	Minimum value for the float attribute.
MAX_VAL	Maximum value for the float attribute.

Field	Explanation
ENTITY	The entities for which this attribute is applicable. A semicolon (;) separated list of entity types of: JOB, STEP, SYMBOL, LAYER, BUILDUP, WHEEL, FEATURE, COMPONENT, or PACKAGE.
GROUP	Name of the class to which this attribute belongs. This could be DFM Analysis, Product, Process: Fabrication, Assembly, Test, Generic, or other. See “Attribute Classes” on page 22.
DEF	The default value of the attribute, if no value is specified. See “Attribute Value Defaults” on page 67.
UNIT_TYPE	DISTANCE, WEIGHT, TEMPERATURE, AREA, RESISTANCE, COPPER_WEIGHT, VOLTAGE, CURRENT. Determines the units used in accordance with the type of measurement. (Default = DISTANCE) See “Units of Measurement” on page 18.
UNITS	Default = NO_UNITS • When UNIT_TYPE=DISTANCE — Possible values are INCH_MM or MIL_MICRONS. Affects the way the value is interpreted—inch or mil, millimeter or micron. • When UNIT_TYPE=RESISTANCE — Possible values are OHM or NANO_OHM. Affects the way the value is interpreted—ohms or nano-ohms. (Default for RESISTANCE is OHM) The UNITS field does not apply to other UNIT_TYPE values. See “Specifying Units for a FLOAT Attribute” on page 68.

Example

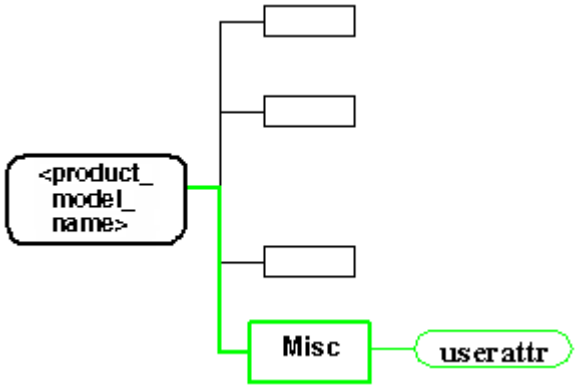
```
FLOAT {
   NAME     = .local_fiducial_dist
   ENTITY   = COMPONENT
   MIN_VAL = 0.0
   MAX_VAL = 100.0
   UNITS    = INCH_MM
   PROMPT   = Search Distance for Local Fiducial
   GROUP = Assembly
   DEF = 50.0
}
```

misc/userattr (User Attribute Definition)

User-defined attributes cannot have the dot character before their name.

userattr	
Type	Line Record Text
Compression	None
Path	*<product_model_name>/misc/userattr*
UNITS	The UNITS directive is required.

The user attribute definition file contains a list of definitions, defined at the time the product model was created. It is read each time the product model is opened.



The format of user attribute definitions is the same as the format of system attribute definitions. See “misc/sysattr.* (Attribute Definition)” on page 66.

For a discussion of ODB++ attributes, see “ODB++ Attributes” on page 22.

wheels (Gerber and Tool Wheels)

Wheels are aperture tables used for the correct translation of Gerber 274D files and/or drill/rout files with external tool definitions. These files relate to the *wheels* entity of the product model:

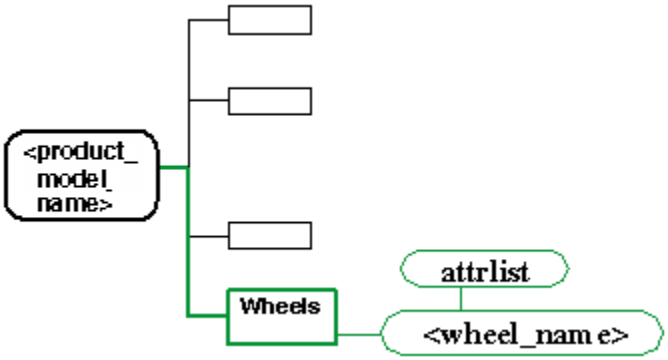
<wheel_name>/attrlist (Wheel Attribute List)

This file contains the values for system attributes and user-defined attributes assigned to a wheel.

You can provide *attrlist* files at these levels: product model, symbol, wheel, step, layer.

attrlist (Attributes Values)	
Type	Structured Text
Compression	None
Path	*<product_model_name>/wheels/<wheel_name>/attrlist*
UNITS	The UNITS directive is required.

System attributes that can be placed in this file are listed with Entity = Wheel in the attribute description. See "System Attributes" on page 189.



The file contains lines of this form: <attribute> = <value>

Example of a wheel attrlist file

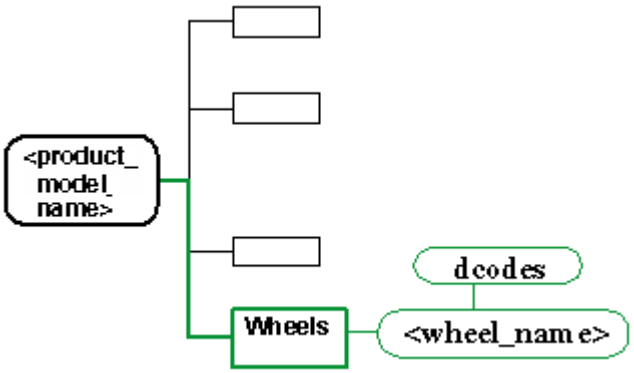
```
UNITS=MM
comment=Mentor wheel
```

<wheel_name>/dcodes (Dcodes)

This file lists dcodes. A dcode is a Gerber dcode number or an Excellon tool number.

dcodes	
Type	Line Record Text
Compression	None
Path	*<product_model_name>/wheels/<wheel_name>/dcodes*
UNITS	The UNITS directive is not relevant for the *dcodes* file.

This file saves the wheel used during Gerber input for translation purposes. Once the file is processed, both the dcodes and Gerber file are no longer referenced.



Example of a dcodes file

This is a typical dcodes file:

```
dcode10   r12   0    no_mirror
dcode11   r50   0    no_mirror
dcode12   r60   0    no_mirror
dcode13   r10   0    no_mirror
dcode14   r70   0    no_mirror
dcode15   r80   0    no_mirror
dcode17   r5    0    no_mirror
```

Each line in the file has the format: **dcode<n>** **<symbol_name>** **<angle>** **<mirror>**, where:

Field	Explanation
n	Dcode number.
symbol_name	Symbol name.
angle	Always 0.
mirror	Always no_mirror.

fonts (Fonts Used in Product Model)

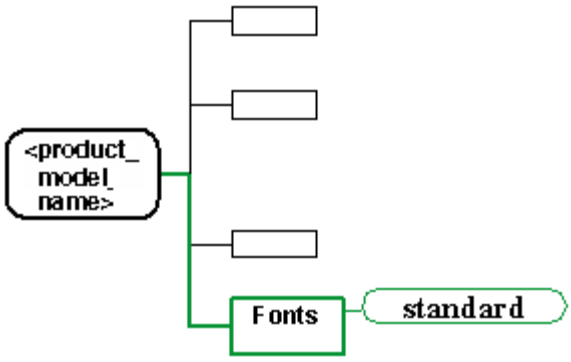
The one file relating to fonts is *standard*.

fonts/standard (Standard Font)

This file describes the vector representation of all the characters that can be a part of a text feature inside a layer.

standard (Standard Font)	
Type	Line Record Text
Compression	None
Path	*<product_model_name>/fonts/standard*
UNITS	The UNITS directive is not relevant for the *standard* file.

Size is scaled according to text requirements. The system supports fully the font named standard.



The origin of each character is at the lower left corner. For best results, font definition should include all ASCII characters.

The file consists of a header containing global parameters, followed by a collection of character blocks.

The header block consists of these lines:

Line	**Explanation**
XSIZE <size>	Horizontal size of a character.
YSIZE <size>	Vertical size of a character.
OFFSET <size>	Horizontal distance between the end of one character block and the beginning of the next one.

A character block consists of these lines:

Line	Explanation
CHAR <char>	Defines the ASCII character defined by this block.
LINE <xs> <ys> <xe> <ye> <pol> <shape> <width>	A character specification contains one or more LINE records, each defining a line used to construct the character. • (**xs**, **ys**) — Starting point of the line. • (**xe**, **ye**) — Ending point of the line. • **pol** — Polarity of the line (P for positive, N for negative). • **shape** — The shape of the ends of the line (R for rounded, S for square). • **width** — Line width in mm or inches. All coordinates are in mm or inches. Because fonts are scaled to specific text style, units in font definition are irrelevant. Coordinates and width can be expressed in any units.
ECHAR	Ends the definition of a character.

Standard Font Characters

The standard font, fully supported by the ODB++ format, contains these characters:

Table 4-3. Standard Font Characters

!	"	#	$	%	&	'	(	)	*	+	,
-	.	/	0	1	2	3	4	5	6	7	8
9	:	;	<	=	>	?	@	A	B	C	D
E	F	G	H	I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X	Y	Z	[	\
]	^	_	`	a	b	c	d	e	f	g	h
i	j	k	l	m	n	o	p	q	r	s	t
u	v	w	x	y	z	{	|	}	~		

Example of the standard font file

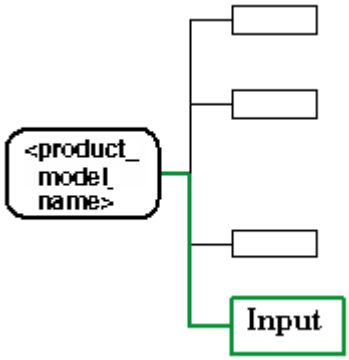
The file consists of a header containing global parameters, followed by a collection of character blocks.

```
XSIZE 0.302000
YSIZE 0.302000
OFFSET 0.000000

CHAR!
LINE 0.000000 0.000000 0.000000 0.200000 P R 0.012000
```

```
LINE 0.000000 -0.100000 0.000000 -0.100000 P R 0.012000
ECHAR

...

CHAR ~
LINE -0.100000 0.150000 -0.050000 0.200000 P R 0.012000
LINE -0.050000 0.200000 0.050000 0.100000 P R 0.012000
LINE 0.050000 0.100000 0.100000 0.150000 P R 0.012000
ECHAR
```

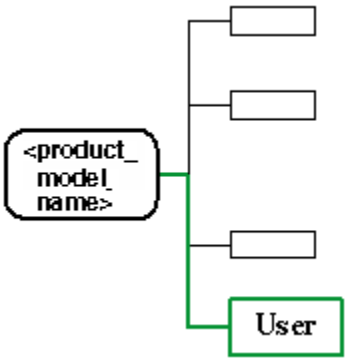
input (Input)

This directory is the default location of files for input packages (not ODB++ files) in the product model.



user (User)

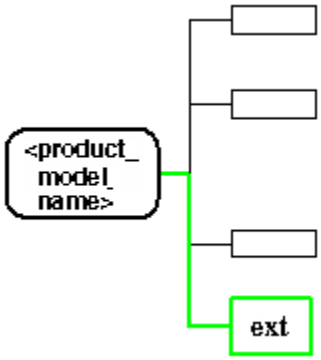
This directory is the location to store user files.



ext (Extension)

This directory is the location to store third party data files.

See “Extensible Content” on page 26.



Chapter 5
Step Entities

Steps are multi-layer entities such as a single image, an assembly panel, a fabrication panel, or a fabrication coupon. Each step contains a collection of layers. Layers are two-dimensional sheets, containing graphics, attributes and annotation. Layers express physical board layers, mask layers, NC drill and rout layers and miscellaneous drawings. All steps in one product model have the same list of layers, even though the contents might be different.

A step is capable of referencing the content of another step, thereby graphically including its content. The referenced step must be within the product model. When this occurs, the layer content for the referenced step appears on the primary step. This process can be repeated any number of times. The most commonly used term for describing this capability is step and repeat, used to create fabrication and assembly panels. Within the primary step, feature display is first defined by the sequence within the referenced step's feature container, then the sequence the steps are added to the primary step, and finally the feature sequence as defined by the primary step's feature container. For more information about feature sequence, see “Geometric Entities” on page 34.

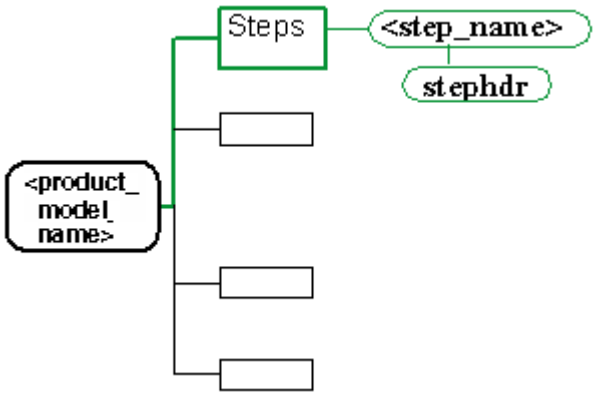
This section describes each element of the step entity.

<step_name>/stephdr (Step Header)

This file contains data that is common to the whole step. This includes the step and repeat array for nested steps.

stephdr (Step Header)	
Type	Structured Text
Compression	None
Path	*<product_model_name>/steps/<step_name>/stephdr*
UNITS	The UNITS directive is required.

The graphic order of precedence for features is as defined in "Geometric Entities" on page 34. When using step repetitive format content within *stephdr* (DX, DY, NX, NY), the referenced step begins at location (X, Y). Subsequent reference steps are first shifted the distance of DX—NX number of times—to complete the first row of reference steps. The format then returns to the previous (X, Y) location, shifts the distance of DY (new Y location), and begins the NX repeat again, completing the next row. This is completed NY number of times. The sequence in which the reference steps are added determines order of precedence for each repeated step, as defined also in "Geometric Entities" on page 34.



The file consists of general fields, and an array of STEP-REPEAT records that exists only if there are nested steps.

These are the general fields:

Field	Explanation
X_DATUM	X datum point (used for step and repeat).
Y_DATUM	Y datum point (used for step and repeat).
ID	A unique ID (optional). See "Unique ID" on page 25.

Field	Explanation
X_ORIGIN	X origin point. Defines the location of the x-zero point (0, 0) in relation to the feature positions.
Y_ORIGIN	Y origin point. Defines the location of the y-zero point (0, 0) in relation to the feature positions.
TOP_ACTIVE BOTTOM_ACTIVE RIGHT_ACTIVE LEFT_ACTIVE	Active area is the area allowed for step and repeat of substeps. It is used to define the border width on the panel edges that should be kept free for placement of coupons and targets. These fields are used only when the step is a panel or array that is meant to contain other steps. (optional)
AFFECTING_BOM	The product model can contain several relevant BOMs. Only one BOM describes the current state of the product model. This field contains the integer **0** to represent that BOM.
AFFECTING_BOM_CHANGED	When a product model is modified to use a different BOM, but the necessary update processes have not been run (for example, package geometries have not been updated to the new BOM) then this field contains the integer **1**.
ONLINE_<online_type>	Obsolete parameters maintained for backward compatibility that may be any number of the DRC_* or NET_* parameters (Obsolete)

These are the fields in the STEP-REPEAT array:

Field	Explanation
NAME	Name of the step to be included in the current step. It must be a valid step in the same product model, without nesting or circular referencing.
X	Start X coordinate for placement of the datum point of the nested step. See the X_DATUM field among the general fields.
Y	Start Y coordinate for placement of the datum point of the nested step. See the Y_DATUM field among the general fields.
DX	Horizontal distance between datum points (when ANGLE = 0).
DY	Vertical distance between datum points (when ANGLE = 0).
NX	Number of repetitions horizontally.

Field	Explanation
NY	Number of repetitions vertically.
ANGLE	Rotation angle of the steps (see “Angles” on page 29).
FLIP	When set to **YES**, all layers in the step instance are mirrored and the layer order is reversed. This can be done only with steps that have a symmetrical buildup (both layer polarities and layer types should be symmetrical).
MIRROR	**YES** for mirror (around x-axis), **NO** for no mirror.

Example of the stephdr file

The file consists of general fields, and an array of STEP-REPEAT records that exists only if there are nested steps.

```
UNITS=INCH
X_DATUM=0
Y_DATUM=0
ID=123456

STEP-REPEAT {
NAME=1UP
X=1.5
Y=1.6
DX=1.2
DY=1.2
NX=6
NY=6
ANGLE=0
FLIP=NO
MIRROR=NO
}
TOP_ACTIVE=1
BOTTOM_ACTIVE=1
RIGHT_ACTIVE=1
LEFT_ACTIVE=1
AFFECTING_BOM=
AFFECTING_BOM_CHANGED=0
```

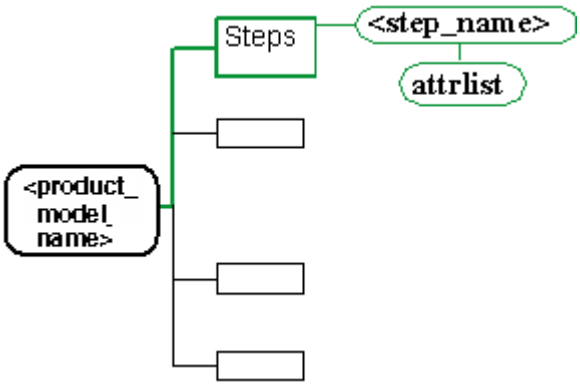
<step_name>/attrlist (Step Attribute List)

This file contains the values for system attributes and user-defined attributes assigned to a step.

You can provide *attrlist* files at these levels: product model, symbol, wheel, step, layer.

attrlist (Attribute List)	
Type	Structured Text
Compression	None
Path	*<product_model_name>/steps/<step_name>/attrlist*
UNITS	The UNITS directive is required.

System attributes that can be placed in this file are listed with Entity = Step in the attribute description. See "System Attributes" on page 189.



The file contains lines of this form: <attribute> = <value>

Example of the attrlist file

In the example, comment is a user-defined attribute, so it does not begin with a dot.

```
UNITS=MM
.board_thickness=50
.out_drill_full = no
.out_drill_optional = no
.out_rout_optional = no
.fs_direction_top = left2right
.fs_direction_bottom = right2left
comment = Production Step
```

<step_name>/layers

Layers express physical board layers, and also mask layers, NC drill and rout layers, and miscellaneous drawings. See “Layer Entities” on page 127 for a discussion of the files relating to the layers.

<step_name>/netlists

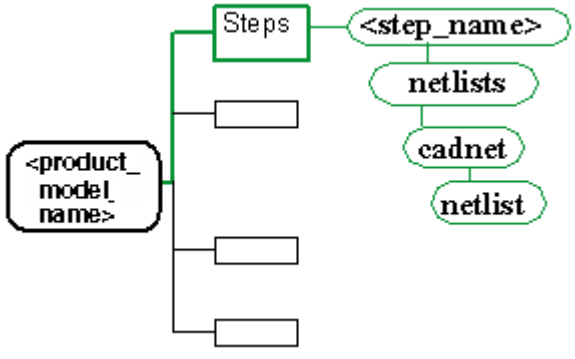
These files are related to the netlists entity:

cadnet/netlist (cadnet)

This file represents a netlist as it was read from an external CAD system. A netlist is a collection of nets, each one referencing a group of points.

cadnet/netlist (cadnet)	
Type	Line Record Text
Compression	None
Path	*<product_model_name>/steps/<step_name>/netlists/cadnet/netlist*
UNITS	The UNITS directive is required.

When a feature does not have a net defined, it is assigned to NET $NONE$. All non-assigned features are defined as NET $NONE$. See “NET—Electrical Net Record” on page 110.



The files consists of these sections:

Header Line

The first line of the file has the form:

H optimize <y|n> [staggered <y|n>]

Field	Explanation
optimize	y — Netlist was optimized for bare board testing. n — Netlist was not optimized. For CAD netlists, the net will always be non-optimized.
staggered (optional)	y — Netlist points have been staggered for testing. n — Netlist points have not been staggered for testing.

List of Nets

The net section of the file contains the nets, in this format:

$<serial_num> <net_name>

Field	Explanation
serial_num	The net serial number, starting with 0.
net_name	A unique name representing the name or a net. A typical practice is to use the original net name used in the design system.

Net names are for reference only. Every <serial_num> net is considered a different electrical net, and should be electrically isolated from all different <serial_num> nets. Ideally, every <serial_num> net should have a different <net_name>. There is no limit on the length of a <net_name>, but <net_name> may contain only printable ASCII characters, excluding the blank space and the semi-colon (;) character.

List of Net Points

The net point section of the file lists the net points in this format:

<net_num> <radius> <x> <y> <side> [<w> <h>] <epoint> <exp> [<"c">] ["staggered" <sx> <sy> <sr>] [<"v">] [<"f">] [<"t">] [<"m">] [<"x">] [<"e">] ["by="<by>] ["arsize_top="<arsize_top>] ["arsize_bot="<arsize_bot>] ["is_shrink="< is_shrink>] ["ld="<ld>]

In the syntax line, these conventions are used:

- < > — Angled brackets represent a single field or value.
- [ ] — Square brackets represent optional fields.
- " " — Quotation marks represent literal strings.

Positional parameters must be placed in the correct order on the line. Keyword parameters can be placed at any position on the line, after the positional parameters.

Fields are space delimited. There should be no spaces before or after an equals sign (=).

Field	Explanation

net_num	Index of the net as defined in the list of nets. See “List of Nets” on page 87.
	For example, if the list of nets includes this line: `$0 MYNET`
	Then these net point lines refer to the net MYNET because the first value in these lines is a zero:
	`0 0.002 4.96 -2.64 T e e staggered 0 0 0`
	`0 0.002 4.94 -2.4775 T e e staggered 0 0 0`
	`0 0.002 4.945 -2.575 B e e staggered 0 0 0`
	A net index value of -1 can be used to represent tooling holes. When a feature is found with no reference to a net index, the net name $NONE$ should be assigned to the feature.

radius	Drill radius (inches or mm). For SMD pads: • In cadnet — 0.002 (independent of units) • In reference net — 0 inch.
x, y	Point coordinates (inches or mm).

Field	Explanation
side	The side from which the net point is accessible. • **T** (top) — Indicates net points accessible from the TOP side of the ODB++ product model. This includes net points placed on inner layers, but accessible from the TOP side through the use of a layer profile hole (cavity). The layer on which a net point resides is defined by the ld parameter. If the ld field is missing, the TOP layer UID is assumed. • **D** (down) — Indicates net points accessible from the BOTTOM side of the ODB++ product model. This includes net points placed on inner layers, but accessible from the BOTTOM side through the use of a profile hole (cavity). The layer on which a net point resides is defined by the **ld** parameter. If the ld field is missing, the BOTTOM layer UID is assumed. • **B** (both) — Indicates net points accessible from TOP and BOTTOM sides of the ODB++ product model. This includes net points placed on inner layers, but accessible from the TOP side and the BOTTOM side through the use of a profile hole (cavity). In this case the net points are accessible from both sides of the product model by means of plated through holes, and non_plated and via holes are excluded. The layers on which a net point resides is defined by the ld value of the drill/rout layer defining the hole location. The matrix file information for the associated drill/rout layer provides the layer accessibility from the TOP and BOTTOM side. If the ld field is missing or does not represent a drill/rout layer, the TOP and BOTTOM layer UIDs are used to define the layer on which to place the net point. • **I** (inner) — Indicates net points that are not accessible from the TOP or BOTTOM side of the ODB++ product model. These are net points resulting from completely buried inner layer surface connections, not exposed by a profile hole (cavity). In this case the net points exclude all internal plated, non_plated and via holes. The layer on which the net point resides is based on the ld parameter. Previous format definitions included following the letter I by an integer representing the layer position within the matrix file. This use is now obsolete.
w, h	Width and height of non-drilled pads (only when radius = 0).
epoint	**e** — Net end point. **m** — Net mid point.
exp	**e** — Solder mask exposed point. **c** — Solder mask covered point. **p** — Solder mask covered point on top side of product model. **s** — Solder mask covered point on bottom side of product model.
c	Comment point flag used to mark net points that are irrelevant to the product model. For example, these points can be ignored during the net comparison process.

Field	Explanation
sx, sy	Coordinates of staggered point (where probe location is other than net point location).
sr	Represents the radius of a probe to be used at this location.
v	Via point.
f	Fiducial point.
t	Test point.
m	Present when a netlist point is designated as a test point by assigning it the **.critical_tp** attribute. Normally this is applied to mid-points that need to be tested. If both **.non_tp** and **.critical_tp** are assigned to the same point, **.critical_tp** takes precedence and the mid point is tested. In case of a drilled pad, the attribute must be added to the drill hole.

x	This indicates that there are extension records in the netlist file. Extension records contain the **e** keyword and have the same location.
	A net point description for an extended point does not have to be grouped together in the netlist file.
	`0 0.00675 0.8 3.3 B m e v x by=b`
	`0 0 2.5 3.214393 T 0.04242 0.04242 e s staggered 0 0.01325 0 e by=c ld=111`

e	This indicates that this is an extension record for a line containing the x keyword.
	A net point description for an extended point does not have to be grouped together in the netlist file.
	`0 0.00675 0.8 3.3 B m e v x by=b`
	`0 0 2.5 3.214393 T 0.04242 0.04242 e s staggered 0 0.01325 0 e by=c ld=111`

by	Defines how a test is executed: **(c s b a n)** **c** — Test from component side. **s** — Test from solder side. **b** — Test from both sides. **a** — Test from any one side. **n** — Side not defined (default). If **<by>** value not defined, **n** is assumed.

arsize_top	The annular ring size for top represents the minimum width of exposed copper (from solder mask) around a drill hole on the top outer layer.
	If the hole does not go through the top layer, arsize_top should not be defined or should be set to 0. Example:
	`4 0.023622 0.726 0.3351969 B m e arsize_top=0.016378 arsize_bot=0.0161873`
	`4 0.015748 0.7460787 0.5300787 B e e by=a arsize_top=0.011752 arsize_bot=0.0116406`
	`4 0.011811 0.0358425 0.1450394 B m c  arsize_top=0 arsize_bot=0`

Field	Explanation

arsize_bot	The annular ring size for bottom represents the minimum width of exposed copper (from solder mask) around a drill hole on the bottom outer layer.
	If the hole does not go through the bottom layer, arsize_bot should not be defined or should be set to 0. Example:
	`4 0.023622 0.726 0.3351969 B m e arsize_top=0.016378 arsize_bot=0.0161873`
	`4 0.015748 0.7460787 0.5300787 B e e by=a arsize_top=0.011752 arsize_bot=0.0116406`
	`4 0.011811 0.0358425 0.1450394 B m c  arsize_top=0 arsize_bot=0`

is_shrink	Indicates that the radius field, which should contain the diameter of the drill, was shrunk to fit the solder mask opening or pad size. (y[es]/[no]) **y** — Point size was shrunk to fit solder mask opening. **n** — Point size is limited only by pad size.
ld	Defines to which layer a net point belongs. The layer is defined by its unique ID. See “Unique ID” on page 25. Optional for top and bottom side, and obligatory for inner points.

Example of a cadnet netlist file

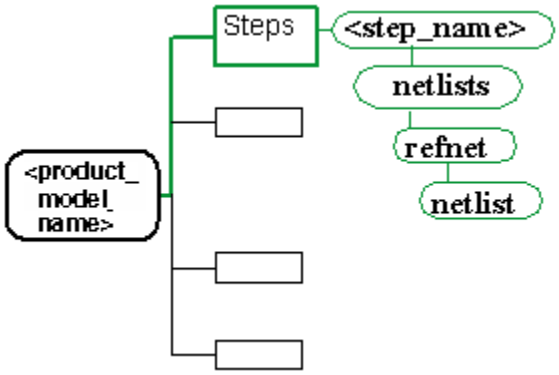
```
UNITS=MM
H optimize n staggered n
$0 &1N1096
$1 &1N1526
$2 &1N289
$3 &1N312
$4 &1N338
$5 &1N340
$6 $NONE$
...
#
#Netlist points
#
0 0.002 4.96 -2.64 T e e staggered 0 0 0
0 0.002 4.94 -2.4775 T e e staggered 0 0 0
0 0.002 4.945 -2.575 B e e staggered 0 0 0
1 0.002 4.31 -4.045 T e e staggered 0 0 0
1 0.002 4.27 -3.893 T e e staggered 0 0 0
```

refnet/netlist (Reference)

This file describes the reference netlist for the step. A reference netlist describes net connectivity as it should be, derived from actual connectivity of the board. This netlist file should be generated and used as the source netlist of choice.

refnet/netlist (Reference)	
Type	Line Record Text
Compression	None
Path	*<product_model_name>/steps/<step_name>/netlists/refnet/netlist*
UNITS	The UNITS directive is required.

The format of the file is identical to that of "cadnet/netlist (cadnet)" on page 86.

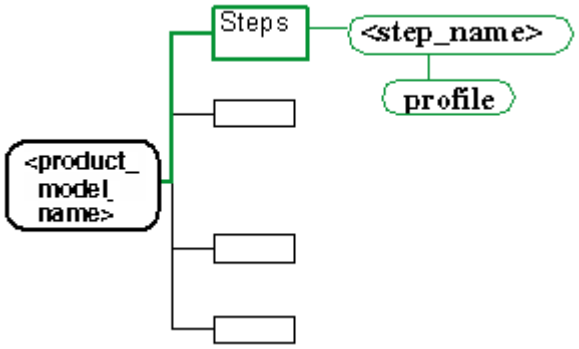


<step_name>/profile (Outline Shape of Step)

The step profile defines the outline shape of the step. It is required by many operations.

profile (Outline Shape of Step)	
Type	Structured Text
Compression	None
Path	*<product_model_name>/steps/<step_name>/profile*
UNITS	The UNITS directive is required.

A profile is created using a single surface feature representing the outline of the step with optional internal holes (cutouts). There can be only one island, but any number of holes inside the island. Holes cannot touch the island boundary or one another.



The *profile* file is a *features* file defining a single surface feature. Surface features are defined as described in “<layer_name>/features (Graphic Features)” on page 139.

Example of a Step Profile File

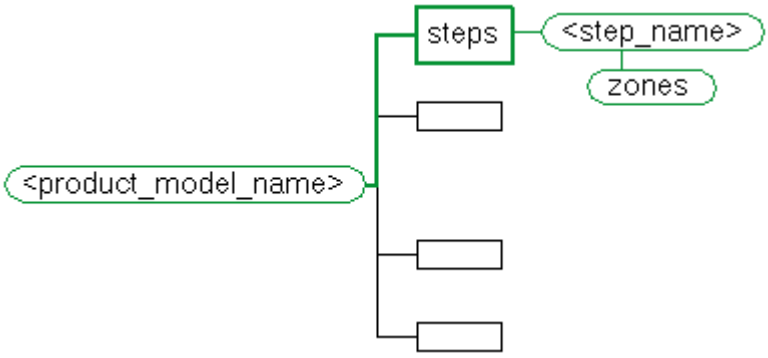
```
UNITS=MM
ID=58
#
#Num features
#
F 1
#
#Layer features
#
S P 0
OB -49.9341975 -64.2563075 I
OS -49.9341975 -0.5988325
OS -0.518735 -0.5988325
OS -0.518735 -64.2563075
OS -49.9341975 -64.2563075
OE
OB -44.3237075 -43.97223 H
OS -15.839685 -43.97223
OS -15.839685 -13.7619025
OS -44.3237075 -13.7619025
OS -44.3237075 -43.97223
OE
SE
```

<step_name>/zones (Matrix Broken Into Sections)

The *zones* file is optional. When the file exists the content provides a cross-sectional view of areas within the profile of a step that are the same.

zones	
Type	Structured Text
Compression	None
Path	*<product_model_name>/steps/<step_name>/zones*
UNITS	The UNITS directive is required.

The file is comprised of multiple regions, or zones, that are defined by surfaces where the board layer materials, including the base layer types of SIGNAL, POWER_GROUND, MIXED, DIELECTRIC, SOLDER_MASK, SOLDER_PASTE, MASK and CONDUCTIVE_PASTE and the associated subtype for each, through the matrix, are identical. An air gap between layers does not require the creation of an additional zone.



A *zones* file has these types of records:

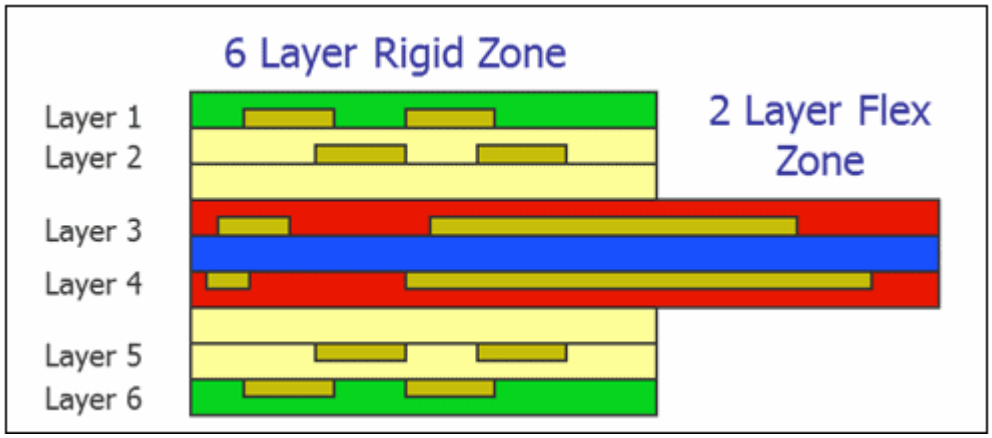
Example of a zones file

For each zone, the *zones* file contains information about the zone, a list of layers in the zone, properties of each of the layers (optional), and a surface feature that describes the outline, islands, and holes of the zone.

```
UNITS=INCH
ID=2411100

#
# Zone 1
#
ZONE area1;UID=2411111
LYR 23 45 78 97
PRP copper_area 2.23 INCH 23
PRP copper_area 5.53 INCH 45
PRP copper_area 6.23 INCH 78
PRP copper_area 2.73 INCH 97
S P 0;;ID=2452187
OB -2 -2 I
OS -2 4
OS 4 4
OS 4 -2
OS -2 -2
OE
OB -1 -1 H
OS 3 -1
OS 3 3
OS -1 3
OS -1 -1
OE
SE
#
# Zone 2
#
ZONE area2;UID=2411112
LYR 45 75
PRP copper_area 1.23 INCH 45
PRP copper_area 3.53 INCH 75
S P 0;;ID=2503253
OB 2 1 I
OS 2 -1
OS 0 -1
OS 0 1
OS 2 1
OE
SE
```

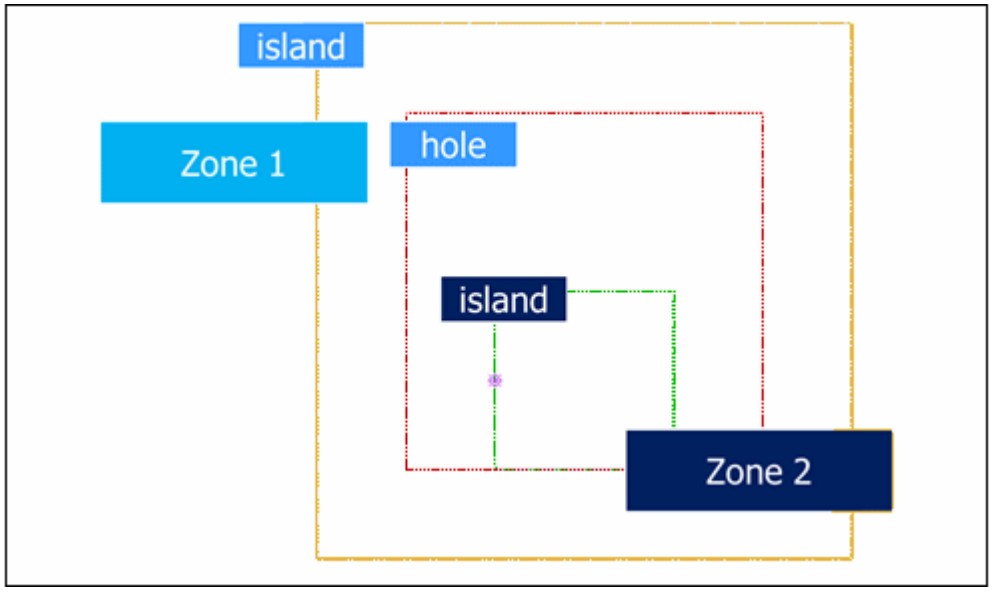
Graphic Illustrating the Layer Cross Section

This graphic illustrates a cross section of a board and the layers for which zones are defined.



Graphic Illustrating the Surfaces Defined for the Zones

This graphic illustrates the surfaces defined for the zones. Zone 1 has an island and a hole. Zone 2 has an island.



ZONE - Start Record

Start zone record: ZONE <name>;zone_UID

Defines the start of a zone by providing a name by which the zone will be referenced.

ZONE Record Parameter	Explanation
name	The name of each zone must be unique and follow ODB++ standard entity guidelines.
zone_UID	A unique ID for this zone.

LYR - Layer Record

Layer record: LYR <layer_UIDs>

Lists the layers that belong to the zone.

LYR Record Parameter	Explanation
layer_UIDs	The list of layer_UIDs to which the ZONE applies. This includes dielectric UIDs. These layer UIDs must be defined in the matrix. See “matrix (Product Model Matrix)” on page 44.

PRP - Properties Records

Properties Record: PRP <name> <value> <units> <layer_UIDs>

Optional. Defines properties of specific layers, such as the enlargement factor or the copper area.

PRP Record Parameter	Explanation
name	One of these names: • **enlarge_zone** — The amount by which the zone will be enlarged. This would apply in situations where the coverlay intrudes into the rigid region of the board (known as a bikini coverlay). • **copper_area** — The copper area for a layer within the zone. This would typically be used with a single layer_UID.
value	The values appropriate to the specified name. • **name = enlarge_zone** — The enlargement amount. • **name = copper_area** — The copper area.

PRP Record Parameter	Explanation
units	Units follow unit of measurements ODB++ format guidelines.
layer_UIDs	The list of layer_UIDs to which the property is assigned. The layer_UIDs must be among those listed in the LYR record for this zone.

S - Surface Defining the Zone Area

The records following the LYR record, and possibly the optional PRP records, define a zone as a single surface feature representing the outline of the zone with its optional internal holes (cutouts). Each zone can be constructed using multiple islands with any number of holes inside each island.

The zones follow the rules that apply to the ODB++ contour definition. These specific rules apply:

- An island must completely surround its holes.
- Holes must not touch one another or intersect within the island.
- No islands defined by two different zones enforced on single layer can intersect, but an island from one zone can be tangent to an island of another zone on a single layer.
- No features other than a surface are legal in the *zones* file.
- A zone may not exist outside of the area defined by the board profile.
- In order for a zone to be defined within a zone, a hole must first be created in the encompassing island.

A surface is defined in a *zones* file with the same syntax used in a *features* file. See “S - Surface Records” on page 156.

<step_name>/boms (Bill of Materials)

The *boms* directory of the step contains a directory for each BOM associated with the step. The name of the directory is the name of the BOM.

The directory for each BOM contains a file named *bom* and a directory named *files*.

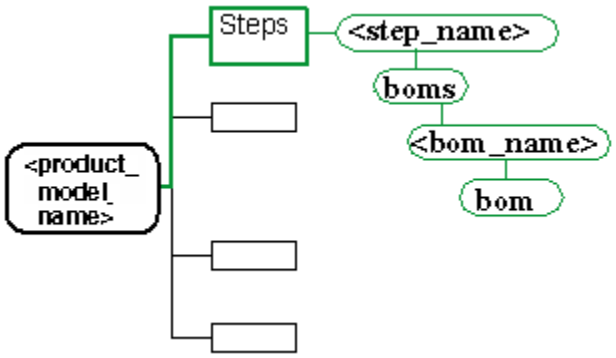
See "Component Variant Implementation" on page 35.

<bom_name>/bom (Bill Of Materials)

The BOM file describes the Bill of Materials for a board. It is a list of the components that can be used in assembling the electronic product.

bom	
Type	Line Record Text
Compression	None
Path	*<product_model_name>/steps/<step_name>/boms/<bom_name>/bom*
UNITS	The UNITS directive is not relevant for the *bom* file.

Each component contains a list of attributes and its uses or locations in the electronic product being assembled.



Each LNFILE line in the *bom* file indicates that source information for the preceding line is stored in the indicated file in the *files* directory. The LNFILE line provides the line number and the file name. For example, LNFILE 5 Rev14.v1 indicates that information is stored in line 5 of file *Rev14.v1*.

Each section of the file starts with a line containing the name of the section, and ends with a line of the format <section_name>_END. For example, RD_CPN and RD_CPN_END.

The file ends with a FILE_END record.

The *bom* file contains these sections:

HEADER Section

The HEADER section starts with a HEADER record and ends with a HEADER_END record.

The header section of the *bom* file contains these parameters:

Parameter	Description
BRD	Board number.
REV	Revision.

DESC_ALIASES Section

The DESC_ALIASES section starts with a DESC_ALIASES record and ends with a DESC_ALIASES_END record.

The DESC_ALIASES section of the *bom* file provides aliases for CPN and MPN descriptions to enable terms to be displayed in the language of the product model creator.

Parameter	Description
LANG	The language code of the product model creator: • **c**—English. • **ja**—Japanese. • **gb**—Simplified Chinese. • **big5**—Traditional Chinese. • **ko**—Korean.

Parameter	Description
INDEX	A number that provides the link between an alias and a description in the CP section. See “CP Section” on page 102. The index starts at 1 and increases. For example, an alias with an index of 3 corresponds to the third description (DSC) of every CPN in the CP section.
CPN|MPN	The alias for the CPN field or the MPN field in either English or a specified language.

RD_CPN Section

The RD_CPN section starts with a RD_CPN record and ends with a RD_CPN_END record.

The RD_CPN section of the *bom* file lists the Reference Designators and their matching Customer Part Numbers:

Parameter	Description
REF	Reference designator name.
CPN	Customer part number.

A LNFILE entry is saved for all parameters.

CPN_MPN Section

The CPN_MPN section starts with a CPN_MPN record and ends with a CPN_MPN_END record.

The CPN_MPN section of the *bom* file lists Customer Parts and their matching Manufacturer Parts:

Parameter	Description
CPN	Customer part number.
VPL_MPN	MPN from an external vendor part library corresponding to the original MPN (as determined in BOM Validation).
VPL_VND	Manufacturer from an external vendor part library corresponding to the original vendor (as determined in BOM Validation).
MPN	Manufacturer part number.
VND	Manufacturer (vendor) name.

Parameter	Description
QLF	Indicates whether the part (MPN+VND) is qualified for production: **-1** — Not qualified. **0** — Unknown. **1** — Qualified.
CHS	Indicates whether this part is chosen from among the alternate parts for the CPN. Only one alternate can be designated as chosen.

A LNFILE entry is saved for CPN, MPN, and VND.

CP Section

The CP section starts with a CP record and ends with a CP_END record.

The CP section of the *bom* file lists Customer Part Numbers and their descriptions:

Parameter	Description
CPN	Customer part number.
IPN	Internal part number.
DSC	Unlimited number of descriptions.
PKG	CAD package name.
QNT	Reference designator quantity.
ITEM	Item number.

A LNFILE entry is saved for CPN, IPN, DSC, and PKG.

Example of a bom file

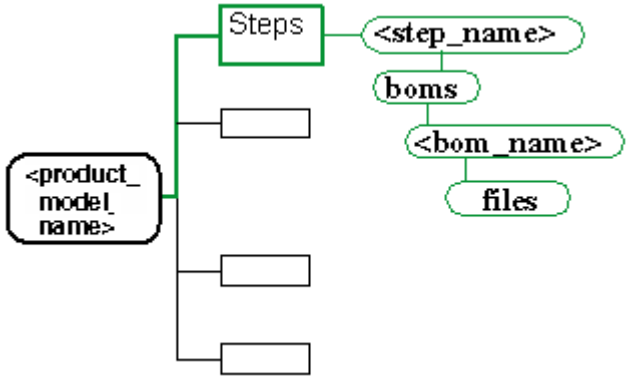
```
# Header Parameters
HEADER
BRD
REV
HEADER_END

# Description Aliases
DESC_ALIASES
LANG fr
INDEX 1
CPN Cout
CPN Cost
INDEX 1
MPN Benefice
MPN Profit Margin
DESC_ALIASES_END
```

```
# Reference Descriptors and matching Customer Parts
RD_CPN
REF XTAL1
LNFILE 5 Rev14.v1
CPN 004-020-101
LNFILE 5 Rev14.v1

REF Y8
LNFILE 7 Rev14.v1
CPN 004-040-101
LNFILE 7 Rev14.v1

RD_CPN_END

# Customer Parts and Matching Manufacturer Parts
CPN_MPN
CPN 004-020-101
LNFILE 5 Rev14.v1
VPL_MPN
VPL_VND TOYOCOM
MPN TQC-216C-6R
LNFILE 5 Rev14.v1
VND TOYOCOM
LNFILE 5 Rev14.v1
QLF 0
CHS 1

CPN 004-020-101
LNFILE 5 Rev14.v1
VPL_MPN
VPL_VND VF
MPN VM6S-20.0000-16PF
LNFILE 6 Rev14.v1
VND VALPEY-FISHER
LNFILE 6 Rev14.v1
QLF 0
CHS 0

CPN_MPN_END

# Customer Parts and Description
CP
CPN 004-020-101
LNFILE 5 Rev14.v1
IPN
LNFILE 5 Rev14.v1
DSC 5 OHM
LNFILE 5 Rev14.v1
DSC UL Listed
LNFILE 5 Rev14.v1
DSC Weight 4
LNFILE 5 Rev14.v1
DSC Price 18
LNFILE 5 Rev14.v1
DSC
LNFILE 10 OHM Rev14.v1
PKG
LNFILE 5 Rev14.v1
```

```
QNT 1
ITEM 0

CPN 004-040-101
LNFILE 7 Rev14.v1
IPN
LNFILE 7 Rev14.v1
DSC 5 OHM
LNFILE 7 Rev14.v1
DSC Weight 3
LNFILE 7 Rev14.v1
DSC Price 24
LNFILE 7 Rev14.v1
DSC Height 0.05
LNFILE 7 Rev14.v1
DSC UL Listed
LNFILE 7 Rev14.v1
PKG
LNFILE 7 Rev14.v1
QNT 1
ITEM 0

CP_END
FILE_END
```

<bom_name>/files (Source Files)

The *files* directory contains the source files that generated the bom entity (BOM and AVL).



LNFILE lines in the *bom* file point to information stored in these files.

<step_name>/eda (Electronic Design Automation)

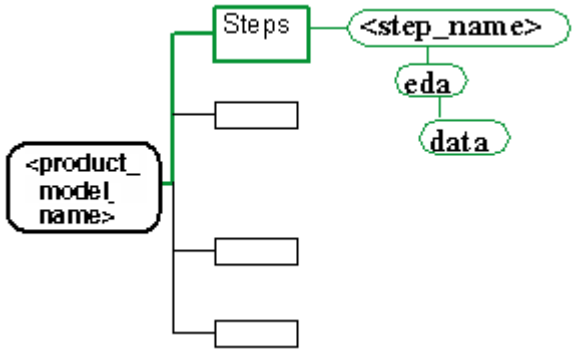
These files are related to the eda entity:

eda/data (EDA Data)

This file contains information that is read from the EDA system directly.

data	
Type	Line Record Text
Compression	Yes
Path	*<product_model_name>/steps/<step_name>/eda/data*
UNITS	The UNITS directive is required.

It covers the library of CAD and user-defined component packages, their outlines and properties, net connectivity information and more.



The file contains these types of records:

Example of an eda data file

```
#
HDR Mentor Boardstation database
UNITS=MM
LYR sst sst+1 sigt sig2 sig3 sig4 sig7 sig8 sig9 sigb smt smb drill1 spt ssb
spb ssb+1 pg6 pg5
#
PRP MILLING_ORIGIN 'MILLING 0 0.0 0' 0 0
PRP DRILL_ORIGIN '' 0 0
PRP FIXED_COMPONENT_LOCATION 'M3,m1' 1
....
#
#Net attribute names
#
#@0 .critical_net
#@1 .diff_pair
#@2 .net_type
#@3 .electrical_class
```

```
#@4 .dpair_gap
#@5 .eclass_rise_time
#@6 .eclass_max_stub_length
#
#
#Net attribute text strings
#
#&0 DEFAULT
#&1 clocks
#&2 EC_PUA
#&3 local
...

# NET 0
NET /D_CL_TX_CLK ;0,2=1,3=2,5=1.500000;ID=1030
SNT TOP T 16 0
FID C 2 33
FID C 10 33
FID C 13 30
...
# PKG 1
PKG *PDXC-L10/HX-L127W51T97 0.1 -0.25 -0.145 0.25 0.145
RC -0.25 -0.1 0.5 0.2
PIN 1 T -0.2 -0.1105 0 E S ID=1034
RC -0.2125 -0.145 0.025 0.069
PIN 3 T -0.1 -0.1105 0 E S ID=1038
RC -0.1125 -0.145 0.025 0.069
PIN 5 T 0 -0.1105 0 E S ID=1041
RC -0.0125 -0.145 0.025 0.069
PIN 7 T 0.1 -0.1105 0 E S ID=1044
RC 0.0875 -0.145 0.025 0.069
PIN 9 T 0.2 -0.1105 0 E S ID=1049
RC 0.1875 -0.145 0.025 0.069
PIN 2 T -0.2 0.1105 0 E S ID=1053
RC -0.2125 0.076 0.025 0.069
PIN 4 T -0.1 0.1105 0 E S ID=1054
RC -0.1125 0.076 0.025 0.069
PIN 6 T 0 0.1105 0 E S ID=1059
RC -0.0125 0.076 0.025 0.069
PIN 8 T 0.1 0.1105 0 E S ID=1073
RC 0.0875 0.076 0.025 0.069
PIN 10 T 0.2 0.1105 0 E S ID=1075
RC 0.1875 0.076 0.025 0.069
#
# PKG 2
PKG *MBCY-T2/XC-L80W80T115 0.1377953 -0.1574803 -0.1574803 0.1574803
0.1574803
CR 0 0 0.1574803
PIN P T 0.0688976 0 0 E T
CR 0.0688976 0 0.011811
PIN N T -0.0688976 0 0 E T
CR -0.0688976 0 0.011811

# FGR 1907
FGR TEXT
PRP string '030'
FID C 14 11018
FID C 14 11018
```

HDR — File Header

This record contains the name of the EDA system that was the source of the data.

HDR <source>

HDR Parameter	Explanation
source	**A** description of the authoring tool from which the product model originated. For example, it can be one of these values: • Mentor Boardstation neutral file • Mentor Boardstation database • Cadence Allegro extract file • Zuken Redac CADIF file • Mentor PADS

LYR — Layer Names

This record contains the names of the layers that are referenced in FID records. See “FID—Feature ID Record” on page 112.

LYR <name1>.... <namen>

LYR Parameter	Explanation
name1, name2, ...	A legal layer name listed in the product model matrix.

PRP — Property Record

A property is a value associated with EDA entities—board, net, package, feature—that specifies some of their characteristics.

PRP <property_name> '<property_value>' n1 n2 ...

PRP Parameter	Explanation
property_name	Name of the property. The property name defines how to interpret the value.
property_value	String of the property (between single quotes).
n1,n2,...	Floating numbers to be kept in the property.

Net Attribute Lookup Tables

This section of the file contains a table of net attribute names and a table of all net attribute values that are strings.

- **Net attribute names** — Each record contains an at sign (@), and index, and an attribute name. For example, @0 .critical_net.
- **Net attribute text strings** — Each record contains an ampersand (&), an index, and a text string that can be used as the value of an attribute of type TEXT. For example, &3 local.

System attributes that can be placed in this table are listed with Entity = Net in the attribute description. See "System Attributes" on page 189

This section must be placed before the first NET record in the file.

The NET records and PKG records in the file can assign net attributes by referencing the index of a net attribute and, for attributes of type TEXT, the index of the text string that is the value assigned to the attribute.

Note

For backward compatibility, net attributes can be preceded by the number sign (#).

This is an example of a net attribute lookup table, and a NET record that assigns attributes.

```
#Net attribute names
#
@0 .critical_net
@1 .diff_pair
@2 .net_type
@3 .electrical_class
@4 .dpair_gap
@5 .eclass_rise_time
@6 .eclass_max_stub_length
#
#Net attribute text strings
#
&0 DEFAULT
&1 clocks
&2 EC_PUA
&3 local
...

# NET 0
NET /D_CL_TX_CLK ;0,2=1,3=2,5=1.500000;ID=1030
```

The NET record in the example should be interpreted as assigning these attributes to a net named /D_CL_TX_CLK:

Assignment in the NET Record	Line in the Table of Attribute Names	Attribute Type	Line in the Table of Attribute Text Strings	Result
0	@0 .critical_net	BOOLEAN	-	.critical_net = TRUE

Assignment in the NET Record	Line in the Table of Attribute Names	Attribute Type	Line in the Table of Attribute Text Strings	Result
2=1	@2 .net_type	TEXT	&1 clocks	.net_type = clocks
3=2	@3 .electrical_class	TEXT	&2 EC_PUA	.electrical_class = EC_PUA
5=1.500000	@5 .eclass_rise_time	FLOAT	-	.eclass_rise_time = 1.5

NET—Electrical Net Record

This record contains a start record of an electrical net. Each net consists of one NET line and 0 or more SNT records.

NET <net_name>; <attributes>;ID=<id> where:

NET Parameter	Explanation
net_name	The same net_name found in the netlist file with which the following records are associated. Net names are for reference only. Every NET record is considered a different electrical net, and should be electrically isolated from all other NET records. Ideally, the <net_name> should be unique across all NET records. <net_name> may contain only printable ASCII characters, excluding the blank space and the semi-colon (;) character. When a feature does not have a net defined it is assigned to NET $NONE$. All un-assigned outer layer pads are defined as NET $NONE$. With more than one $NONE$ net, each is disconnected from the other. Any two points of a $NONE$ net can be connected or disconnected, depending on the design.
attributes	A comma separated list of attribute assignments. The attribute is indicated by its index in the attribute name lookup table. • **BOOLEAN** — n indicating that attribute n is set. • **OPTION** — n=m indicating that attribute n has value m. • **INTEGER** — n=i indicating that attribute n has value i. • **FLOAT** — n=f indicating that attribute n has value f. • **TEXT** — n=s indicating that attribute n has the value associated with index s in the attribute text string lookup table. See “Net Attribute Lookup Tables” on page 108.
ID=<id>	Assigns a unique identifier to the net. See “Unique ID” on page 25.

SNT— Subnet Record

This record defines a portion of a net. The portion can be one of these types of subnet:

- **Toeprint** — A connection of a component pin to the board.
- **Via** — A connectivity padstack between layers.
- **Trace** — A collection of lines/arcs leading from point to point.
- **Plane** — A surface used for connectivity purposes.

Each subnet record is followed by zero or more FID records mapped to the board features that are part of this subnet.

Structure for toeprint: SNT TOP <side> <comp_num> <toep_num>

SNT TOP Field	Explanation
TOP	Indicates that the subnet is a toeprint.
side	**T** — Top side of the product model. **B** — Bottom side of the product model.
comp_num	An index number (0...n-1) corresponding to the component record in the file *comp_+_top/components* or *comp_+_bot/components*
toep_num	An index number (0...n-1) corresponding to a toeprint record of the referenced component.

Structure for via: SNT VIA

SNT VIA Field	Explanation
VIA	Indicates that the subnet is a via.

Structure for trace: SNT TRC

SNT TRC Field	Explanation
TRC	Indicates that the subnet is a trace.

Structure for plane: SNT PLN <fill_type> <cutout_type> <fill_size>, where:

SNT PLN Field	Explanation
PLN	Indicates that the subnet is a plane.
fill_type	**S** — Solid. **O** — Outline.
cutout_type	**C** — Circle. **R** — Rect. **O** — Octagon. **E** — Exact.

SNT PLN Field	Explanation
fill_size	Size in inches or mm of fill brush. (For solid fill, use the value 0.)

FID—Feature ID Record

This record contains a link to the feature file of layer number <lyr_num>. The record is used to connect subnets and feature groups to the board features that are part of them.

Structure: FID <type> <lyr_num> <f_num>

FID Parameter	Explanation
type	**C** — Copper. **L** — Laminate. **H** — Hole.
lyr_num	A layer number (0 ... n-1) corresponding to the names of layers in the LYR record
f_num	A feature number (0 ... n-1) corresponding to the feature record sequence in the features file.

An example of the f_num sequence in a feature file would be:

```
#
#Layer features
#
P 4.057087 4.5 6 P 0 0;1=6,3=0          #f_num = 0
P 4.057087 4.57874 6 P 0 0;1=6,3=0      #f_num = 1
P 4.057087 3.633858 6 P 0 0;1=6,3=0     #f_num = 2
P 4.057087 3.712598 6 P 0 0;1=6,3=0     #f_num = 3
```

PKG—Package Record

This record contains a definition of a package that is the generic shape of a component. Each component refers to a package.

Each PKG line must be followed immediately by one or more outline records, 0 or more property (PRP) records, and 0 or more pin records. See “RC, CR, SQ, CT, OB, OS, OC, OE, CE — Outline Records” on page 114.

The order in which the PKG records occur is used by the CMP record (pkg_ref) of the component file to determine the desired package. The order is referenced starting from 0 and up. See “CMP—Component Record” on page 161.

ODB++ requires closed geometries. That is, polygons must be closed and not intersect.

Structure: PKG <name> <pitch> <xmin> <ymin> <xmax> <ymax>; <attributes>;ID=<id>

PKG Parameter	Explanation
name	Name of the package as defined in the EDA system
pitch	Distance between center of closest pins, in inches or mm, based on the current unit of measure
xmin, ymin, xmax, ymax	Bounding box of package, relating to package datum
attributes	A comma separated list of attribute assignments. The attribute is indicated by its index in the attribute name lookup table. • **BOOLEAN** — n indicating that attribute n is set. • **OPTION** — n=m indicating that attribute n has value m. • **INTEGER** — n=i indicating that attribute n has value i. • **FLOAT** — n=f indicating that attribute n has value f. • **TEXT** — n=s indicating that attribute n has the value associated with index s in the attribute text string lookup table. See "Net Attribute Lookup Tables" on page 108.
ID=<id>	Assigns a unique identifier the package. See "Unique ID" on page 25.

PIN—Pin Record

This record contains a definition of a pin that belongs to a package.

Each Pin record is followed by at least one outline record. See "RC, CR, SQ, CT, OB, OS, OC, OE, CE — Outline Records" on page 114.

By default, pin #1 is determined by the first PIN record in the package. If the integer attribute .polarity_marker is assigned for the package, then that value overrides the default behavior by assigning pin #1 to the occurrence of the PIN record indicated by that integer.

Structure: **PIN <name> <type> <xc> <yc> <fhs> <etype> <mtype> ID=<id>**, where:

PIN Parameter	Explanation
name	Name of the pin as defined in the EDA system.
type	**T** — Through-hole (top to bottom). **B** — Blind (top to inner or inner to bottom). **S** — Surface (top to top or bottom to bottom).
xc, yc	Center of pin, relating to package datum.
fhs	Finished hole size (currently not used—should be 0).

PIN Parameter	Explanation
etype	Pin electrical type: **E** — Electrical. **M** — Non-electrical (mechanical). **U** — Undefined (default).
mtype	Pin mount type: **S** — SMT **D** — Recommended SMT pad (where the pin size is the recommended pad size and not the pin size). **T** — Through-hole. **R** — Through-hole where the pin size is the recommended hole size and not the pin size. **P** — Pressfit. **N** — Non-board, pins without contact area with the board. Used in components with lead forms of types: solder lug, high cable, or quick connect. **H** — Hole, for physical holes that appear without the physical pin **U** — Undefined (default).
ID=<id>	Assigns a unique identifier to the pin. See “Unique ID” on page 25.

FGR—Feature Group Record

This record contains the definition of a group of related features—for example, the strokes of a text record. Each **FGR** line is followed by zero or more **FID** records mapped to the board features that are part of this subnet.

Structure: **FGR <type>**, where **<type>** is the string **TEXT**—the only value allowed.

RC, CR, SQ, CT, OB, OS, OC, OE, CE — Outline Records

Outline records must follow a PKG or PIN record. They describe the shape of the package or pin.

A PKG record must have an outline record as the immediate next entry (an outline record can be more than one line). A PIN record does require an outline record but not immediately after.

A shape can consist of a simple shape (rectangle, circle, square) or a complex contour.

- **RC** — Rectangle record.

  RC <lower_left_x> <lower_left_y> <width> <height>

- **CR** — Circle record.

  CR <xc> <yc> <radius>

- **SQ** — Square record.

  SQ <xc> <yc> <half side>

- **CT, OB ... CE** — Contour record.

  The structure of a contour record is the same as a surface feature in the features file and is restricted by the same limitations. A contour consists of one or more polygons.

  - Self-intersecting polygons (SIPs) are not allowed.
  - The polygons must form a closed shape, as described:

    A polygon starts with an **OB** command, contains **OS** (segment) or **OC** (arc) commands, and ends with an **OE** command.

    ```
    OB <start_x> <start_y> I/H     (I=island, H=hole)
    OS <end_x> <end_y>
    OC <end_x> <end_y> <center_x> <center_y> <cw> (cw = y or n)
    OE
    ```

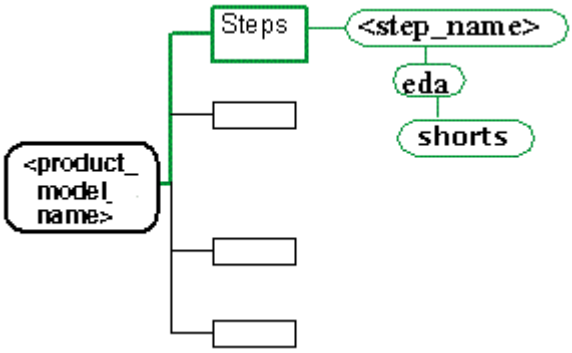
  - Holes must be graphically contained inside island polygons.
  - The arcs must be consistent (the start, end, and center point must construct a legal arc).

eda/shorts (Electrical Intention Short Net)

This file contains information that is read from the EDA system directly.

data	
Type	Line Record Text
Compression	Yes
Path	*<product_model_name>/steps/<step_name>/eda/shorts*
UNITS	The UNITS directive is required.

The file contains a list of features that are known to create intentional net shorts within the product model.



The file contains records of type LYR, SRT, and FID. An example is provided:

- LYR — Layer Names in the Shorts File
- SRT — Electrical Intention Short Net Record
- FID—Feature ID Record in the Shorts File
- Examples of Multiple Nets Shorted Together

Example of a shorts file

```
#
UNITS=MM
LYR sst sigt sig2 sig3 sig4 sig7 sig8 sig9 sigb smt smb drill spt ssb spb
pg6 pg5
# SRT 1
SRT 1 7;ID=20
FID C 4 2
FID C 4 3
# SRT 2
SRT 2 7;ID=21
FID C 4 4
FID C 4 3
```

```
# SRT 3
SRT 3 7;ID=22
FID C 4 6
FID C 4 3
# SRT 4
SRT 4 7;ID=23
FID C 4 8
FID C 4 3
# SRT 5
SRT 5 7;ID=24
FID C 4 10
FID C 4 3
# SRT 6
SRT 6 7;ID=25
FID C 4 12
FID C 4 3
```

LYR — Layer Names in the Shorts File

This record contains the names of the layers that are referenced in FID records. See “FID— Feature ID Record in the Shorts File” on page 118.

LYR <name1>.... <namen>

LYR Parameter	Explanation
name1, name2, ...	A legal layer name listed in the product model matrix.

SRT — Electrical Intention Short Net Record

This record contains a start record for defining an intentional short.

Each short definition contains at least two FID records, or one from each net name. The need for multiple FID records that go through layers would result from a via or through hole location participating in a short.

If the same nets are shorted together in multiple locations for a single product model, then the multiple short locations should be recorded separately and as each would contain their own FID related records. This also means that there might be more than one SRT record containing the same NET net_name IDs.

If more than two nets are short together, each instance of the resulting net shorts must be defined separately. For example, if there are three nets, net1, net2 and net3 that are shorted together, then the short of net1 to net2, net2 to net3 and net3 to net1 will need to be listed in the file. See “Examples of Multiple Nets Shorted Together” on page 119.

SRT <net_name1_uid> <net_name2_uid> ID=<id>

SRT Parameter	Explanation
net_name1_uid	The ID from the NET record from within the *eda/data* file referencing the first net shorted together.
net_name2_uid	The ID from the NET record from within the *eda/data* file referencing the second net shorted together.
ID=<id>	A fixed string to indicate the following value is a unique ID, and its numerical value within the legal range of 0 - 4294967295 (optional).

FID—Feature ID Record in the Shorts File

This record contains a link to the feature file of layer number <lyr_num>. The record is used to connect subnets and feature groups to the board features that are part of them.

Structure: FID <type> <lyr_num> <f_num>

FID Parameter	Explanation
type	FID records in a *shorts* file can only be of type C or H. **C** — Copper. **H** — Hole.
lyr_num	A layer number (0 ... n-1) corresponding to the names of layers in the LYR record
f_num	A feature number (0 ... n-1) corresponding to the feature record sequence in the features file.

An example of the f_num sequence in a feature file would be:

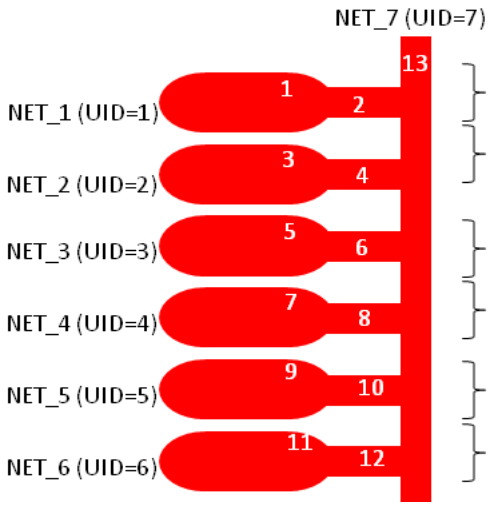
```
#
#Layer features
#
P 4.057087 4.5 6 P 0 0;1=6,3=0          #f_num = 0
P 4.057087 4.57874 6 P 0 0;1=6,3=0      #f_num = 1
P 4.057087 3.633858 6 P 0 0;1=6,3=0     #f_num = 2
P 4.057087 3.712598 6 P 0 0;1=6,3=0     #f_num = 3
```

Examples of Multiple Nets Shorted Together

If more than two nets are short together, each instance of the resulting net shorts must be defined separately. If the copper feature that caused the shorts has a net name, the number or shorts that must be defined is reduced.

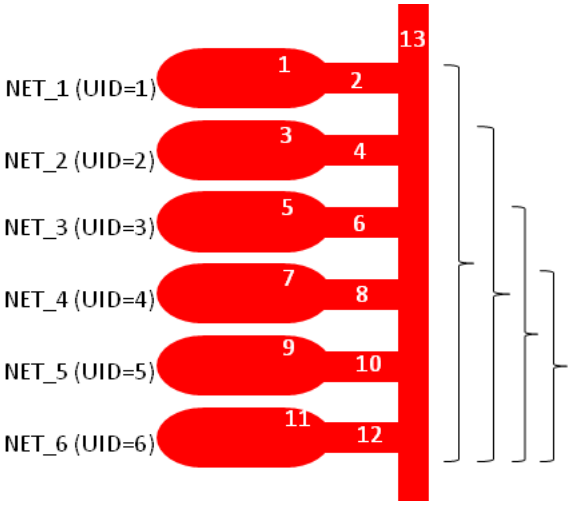
Example of multiple nets shorted together where the copper feature is provided a net name:

Explanation	Definitions in the shorts file	Graphic

In this example, copper feature 13, which causes the short, has a net name with UID = 7.
Six SRT sections are required. Each one describes a short between net 7 and one of the other six nets.
Each SRT section has two FID lines. Each one identifies a feature involved in the short.
	`# SRT 1`
	`SRT 1 7;ID=20`
	`FID C 4 2`
	`FID C 4 13`
	`# SRT 2`
	`SRT 2 7;ID=21`
	`FID C 4 4`
	`FID C 4 13`
	`# SRT 3`
	`SRT 3 7;ID=22`
	`FID C 4 6`
	`FID C 4 13`
	`# SRT 4`
	`SRT 4 7;ID=23`
	`FID C 4 8`
	`FID C 4 13`
	`# SRT 5`
	`SRT 5 7;ID=24`
	`FID C 4 10`
	`FID C 4 13`
	`# SRT 6`
	`SRT 6 7;ID=25`
	`FID C 4 12`
	`FID C 4 13`
		

Example of multiple nets shorted together, where the copper feature is not provided a net name:

Explanation	Definitions in the shorts file	Graphic

In this example, copper feature 13, which causes the short, does not have a net name.
A SRT section is required for each possible pair of the six nets that are shorted together by copper feature 13:
1-2, 1-3, 1-4, 1-5, 1-6
2-3, 2-4, 2-5, 2-6
3-4, 3-5, 3-6
4-5, 4-6
5-6
Each SRT section has three FID lines to define the three features involved in the short.
	`# SRT 1`
	`SRT 1 2;ID=20`
	`FID C 4 2`
	`FID C 4 13`
	`FID C 4 4`
	`# SRT 2`
	`SRT 1 3;ID=21`
	`FID C 4 2`
	`FID C 4 13`
	`FID C 4 6`
	`# SRT 3`
	`SRT 1 4;ID=22`
	`FID C 4 2`
	`FID C 4 13`
	`FID C 4 8`
	`# SRT 4`
	`SRT 1 5;ID=23`
	`FID C 4 2`
	`FID C 4 13`
	`FID C 4 10`
	`# SRT 5`
	`SRT 1 6;ID=24`
	`FID C 4 2`
	`FID C 4 13`
	`FID C 4 12`
	`# SRT 6`
	`SRT 2 3;ID=25`
	`FID C 4 4`
	`FID C 4 13`
	`FID C 4 6`
	
	`# SRT 7`
	`SRT 2 4;ID=26`
	`FID C 4 4`
	`FID C 4 13`
	`FID C 4 8`
	`# SRT 8`
	`SRT 2 5;ID=27`
	`FID C 4 4`
	`FID C 4 13`
	`FID C 4 10`
	`# SRT 9`
	`SRT 2 6;ID=28`
	`FID C 4 4`
	`FID C 4 13`
	`FID C 4 12`
	`...`
		

eda/hdi_netlist (HDI Netlist)

A netlist defining wire bond connection is defined in this file. This netlist defines connections between pins of source and destination components for nets including wire bonds.

hdi_netlist	
Type	Structured Text
Compression	No
Path	*<product_model_name>/steps/<step_name>/eda/hdi_netlist*
UNITS	The UNITS directive is required.

The hdi netlist file contains these sections:

- HDI_NETLIST_DATABASE array structure
- HDI_PACKAGE array structure
- HDI_NET_POINT array structure
- HDI_NET_BRIDGE array structure

HDI_NETLIST_DATABASE array structure

The file contains a single HDI_NETLIST_DATABASE record containing common information:

Table 5-1. HDI_NETLIST_DATABASE Array Structure

Parameter	Explanation
units	Units of coordinates in the file. (INCH / MM)
source	Application used to create ODB++, or input source if created by Valor NPI.
version	Netlist version.

Example of the HDI_NETLIST_DATABASE section:

```
HDI_NETLIST_DATABASE {
    units=MM
    source=ExpeditionPCB
    version=2.0
}
```

HDI_PACKAGE array structure

The file contains a HDI_PACKAGE records for each component involved in the netlist. These components are specified in the die parameter or the dest_comp parameter of the HDI_NET_POINT records.

Table 5-2. HDI_PACKAGE Array Structure

Parameter	Explanation
name	Name of a component used in HDI_NET_POINT records.
ref_des	Reference designator of the corresponding component in the design (usually the same as the value in name).
type	DIE or SRC for a source component; BGA or DST for a destination component.

Example of the HDI_PACKAGE section:

```
HDI_PACKAGE {
   name=U4
   ref_des=U4
   type=DIE
}
```

HDI_NET_POINT array structure

The file contains multiple HDI_NET_POINT records for all pins of source and destination components connected by wire bonds. The records contains the same fields for both source and destinations but some fields are empty depending on the net point and connection type..

Table 5-3. HDI_NET_POINT Array Structure

Parameter	Explanation	Source and Destination
net	Name of die bump net (should be the same as the value of .net attribute of corresponding pad in wire bond layer).	pin to pin pin to net pin to gate
x, y	Die bump coordinates (the same—within a certain tolerance—as the coordinates of the corresponding pad in the wire bond layer).	pin to pin pin to net pin to gate
sym	Similar to the symbol name of corresponding pad in wire bond layer.	pin to pin pin to net pin to gate
die	Name of the source component. An HDI_PACKAGE record must exist for the component.	pin to pin pin to net pin to gate

Table 5-3. HDI_NET_POINT Array Structure

Parameter	Explanation	Source and Destination
bump	Name of the source component pin.	pin to pin pin to net pin to gate
gate	Gate name. This is used as a key to find the corresponding destination gate record. See the examples. ODB++ does not contain a corresponding entity, feature, or attribute.	pin to gate
dest_net	Name of the destination net.	pin to net destination net
dest_comp	Name of the destination component. An HDI_PACKAGE record must exist for the component.	pin to pin pin to net destination net destination gate
dest_pin	Name of destination component pin.	pin to pin destination net destination gate
dest_gate	Name of the destination gate. The gate parameter in a pin to gate record points to a gate specified in the dest_gate parameter of a destination gate record. See the examples.	destination gate

Example of the HDI_NET_POINT section for a source pin connected to a destination pin

```
HDI_NET_POINT {
    net=X_X_ATB1TL
    x=-6.430395
    y=4.160125
    sym=rect60x49
    die=U1
    bump=35
    gate=
    dest_net=
    dest_comp=B1
    dest_pin=B4
    dest_gate=
}
```

Example of the HDI_NET_POINT section for a source pin connected to a destination net

A record specifying a destination net instead of a destination pin is used for a net containing multiple pins of the same destination component (typically power / ground nets).

Each pin of the destination component connected to such a net should be represented by separate HDI_NET_POINT record for a destination net.

```
HDI_NET_POINT {
    net=X_X_ATB1TL
    x=-6.430395
    y=4.160125
    sym=rect60x49
    die=U1
    bump=35
    gate=
    dest_net=X_X_ATB1TL
    dest_comp=
    dest_pin=
    dest_gate=
}
```

Example of the HDI_NET_POINT section for a destination net

Multiple records with the same dest_net are allowed. Typically, the destination net and source net are the same, but different nets are also supported.

```
HDI_NET_POINT {
    net=
    x=
    y=
    sym=
    die=
    bump=
    gate=
    dest_net=X_X_ATB1TL
    dest_comp=B1
    dest_pin=B4
    dest_gate=
}
```

Example of the HDI_NET_POINT section for a source pin connected to a destination gate

```
HDI_NET_POINT {
    net=X_X_ATB1TL
    x=-6.430395
    y=4.160125
    sym=rect60x49
    die=U1
    bump=35
    gate=GATE23
    dest_net=
    dest_comp=
    dest_pin=
    dest_gate=
}
```

Example of the HDI_NET_POINT section for a destination gate

Multiple records with the same dest_gate are allowed. Source and destination net should be the same in the case of a connection to a gate.

```
HDI_NET_POINT {
    net=
    x=
    y=
    sym=
    die=
    bump=
    gate=
    dest_net=
    dest_comp=B1
    dest_pin=B4
    dest_gate=GATE23
}
```

HDI_NET_BRIDGE array structure

The netlist file can contain an optional section for a direct wire bond connection in the same copper layer. No component pins are involved in such a connection.

Table 5-4. HDI_NET_BRIDGE Array Structure

Parameter	Explanation
net	Name of the net with a direct wire bond connection.
x1, y1	Coordinates of one of the connected points. The copper layer should contain pads with similar coordinates or lines / arcs with similar endpoints.
x2, y2	Coordinates of one of the connected points. The copper layer should contain pads with similar coordinates or lines / arcs with similar endpoints.

Example of an HDI_NET_BRIDGE section

```
HDI_NET_BRIDGE {
    net=VCCEL_L
    1=0.37
    y1=0.54
    x2=0.385
    y2=0.544
}
```

Chapter 6
Layer Entities

Product model layers can contain graphics, properties and annotation. Layers represent physical board layers, NC drills and rout layers, miscellaneous drawings, and dielectric separators. The order in which layers are arranged is based on the content in the matrix. Layers of type DIELECTRIC can only contain information representing material attributes, such as thickness. For information on the matrix, see “matrix (Product Model Matrix)” on page 44.

This section describes each element of the layer entity.

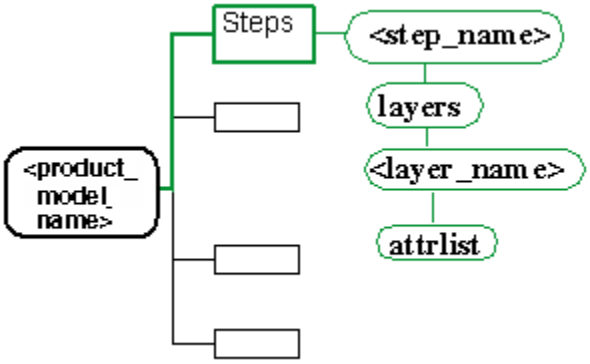
<layer_name>/attrlist (Layer Attribute List)

This file contains the values for system attributes and user-defined attributes of a layer.

You can provide *attrlist* files at these levels: product model, symbol, wheel, step, layer.

attrlist (Attribute List)	
Type	Structured Text
Compression	None
Path	*<product_model_name>/steps/<step_name>/layers/<layer_name>/attrlist*
UNITS	The UNITS directive is required.

System attributes that can be placed in this file are listed with Entity = Layer in the attribute description. See “System Attributes” on page 189.



The file contains lines of this form: <attribute> = <value>

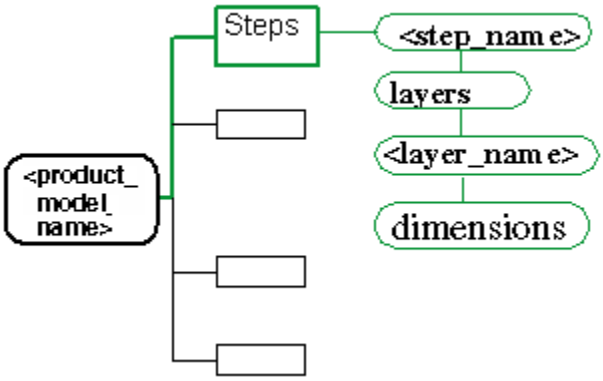
Example of a layer attrlist file

```
UNITS=MM
.out_mirror = no
.inp_file =
.eda_layers = "signal_2","signal","VIA"
.out_angle = 0.0
.out_polarity = positive
.out_x_scale = 1.000000
.out_y_scale = 1.000000
.out_comp = 0.000000
```

<layer_name>/dimensions (Dimensions)

Dimensions that have been added to a layer to indicate measurements of objects on the layer are stored with the product model.

dimensions (Dimensions)	
Type	Structured Text
Compression	Yes
Path	*<product_model_name>/steps/<step_name>/layers/<layer_name>/dimensions*
UNITS	The UNITS directive is required.



The dimensions file contains these sections:

- Global Parameters for Dimensions
- PARAMETERS array structure
- DIMENSION array structure
- PAPER array structure
- TEXT array structure

Global Parameters for Dimensions

These are the global parameters:

Table 6-1. Global Parameters for Dimensions

Field	Explanation
VERSION	File version. <int>, >= 0
UNITS	Units of all coordinates and floating numbers. <string> Possible values: INCH | MM

PARAMETERS array structure

This is the array structure of each parameter set:

Table 6-2. PARAMETERS Array Structure

Field	Explanation
ARROW_STYLE	Controls the appearance of the dimension arrow. Optional. <int>, >= 0
ID	The ID of this parameter set. DIMENSION records reference parameter sets by this number. See “DIMENSION array structure” on page 131. <int>, >= 0
LINE_WIDTH	Dimension line width. <float>, 0 - 500 mil
POST_DECIMAL_DIST	Number of decimal digits for size type dimensions. <int>, 0 - 6
POST_DECIMAL_POS	Number of decimal digits for LOCATION type dimensions. <int>, 0 - 6
POST_DECIMAL_ANGLE	Number of decimal digits for ANGLE type dimensions. <int>, 0 - 6
FONT	Name of a font from the fonts folder of the product model. <string>
FONT_WIDTH	Width of text letters. <float>, 0 - 5000 mil
FONT_HEIGHT	Height of text letters. <float>, 0 - 5000 mil
EXT_OVERLEN	Extension overlength for HORIZONTAL | VERTICAL | PARALLEL | ANGLE. <float>, -100 INCH - 100 INCH
EXT_OFFSET	Extension offset for HORIZONTAL | VERTICAL | PARALLEL | ANGLE. <float>, -100 INCH - 100 INCH
CENTER_MARKER_LEN	Cross size for CENTER dimension. <float>, 0 - 100 INCH
BASELINE_SPACING	Spacing for HORIZONTAL, VERTICAL, PARALLEL. <float>, 0 - 100 INCH
ORIGIN_X	<x coordinate>, -100 INCH - 100 INCH
ORIGIN_Y	<y coordinate>, -100 INCH - 100 INCH

Table 6-2. PARAMETERS Array Structure

Field	Explanation
SCALE	Frontline scaling percent. <float>, 0 - 100
PAPER	The PAPER array. See “PAPER array structure” on page 132.

DIMENSION array structure

This is the dimension array structure:

Table 6-3. DIMENSION Array Structure

Field	Explanation
TYPE	<string> Possible values: HORIZONTAL | VERTICAL | PARALLEL | RADIUS | DIAMETER | CHAMFER | CENTER | LOCATION | ANGLE
PARAMETERS	The ID number of a parameter set. See “PARAMETERS array structure” on page 130. <int>
REF1X	<x coordinate>, -100 INCH - 100 INCH
REF1Y	<y coordinate>, -100 INCH - 100 INCH
REF2X	<x coordinate>, -100 INCH - 100 INCH Not relevant to CENTER.
REF2Y	<y coordinate>, -100 INCH - 100 INCH Not relevant to CENTER.
REF3X	<x coordinate>, -100 INCH - 100 INCH Relevant to RADIUS, DIAMETER, CHAMFER, LOCATION, ANGLE.
REF3Y	<y coordinate>, -100 INCH - 100 INCH Relevant to RADIUS, DIAMETER, CHAMFER, LOCATION, ANGLE.
LINE_PT_X	<x coordinate>, -100 INCH - 100 INCH Not relevant to CENTER.
LINE_PT_Y	<y coordinate>, -100 INCH - 100 INCH Not relevant to CENTER.
OFFSET	Offset for RADIUS, DIAMETER, CHAMFER, LOCATION. <float>, -100 INCH - 100 INCH
ARROW_POS	<string> Possible values: AUTOMATIC | INSIDE | OUTSIDE

Table 6-3. DIMENSION Array Structure

Field	Explanation
MAGNIFY	Frontline magnifcation percent. <float>, 0 - 100
TEXT	The TEXT array. See “TEXT array structure” on page 133.
TO_ARC_CENTER	<string> Possible values: NO | YES Relevant to RADIUS, DIAMETER.
TWO_SIDED_ DIAM	<string> Possible values: NO | YES Relevant to RADIUS, DIAMETER.

PAPER array structure

This is the array structure for the Frontline paper definitions. It is included in the PARAMETERS array.

Table 6-4. PAPER Array Structure

Field	Explanation
ORIENTATION	<string> Possible values: NONE | PORTRAIT | LANDSCAPE
SIZE	<string> Possible values for the Postscript paper size: A0 | A1 | A2 | A3 | A4 | A5 | B4 | B5 | LETTER | LIFE | CUSTOM
WIDTH	<float>
HEIGHT	<float>
X	<float>
Y	<float>
MARGIN	The MARGIN array contains this information: TOP = <float> BOTTOM = <float> LEFT = <float> RIGHT = <float>
ACTIVE	The ACTIVE array contains this information: X00 = <float> Y00 = <float> X11 = <float> Y11 = <float>

Table 6-4. PAPER Array Structure

Field	Explanation
COLOR	The COLOR array contains this information: FEATURE = <string> DIMENS = <string> DIMENS_TEXT = <string> PROFILE = <string> TEMPLATE = <string>

TEXT array structure

The TEXT array structure is not relevant to the CENTER dimension. It is included in the DIMENSIONS array.

Table 6-5. TEXT Array Structure

Field	Explanation
PREFIX	<string>, 0-20 characters
SUFFIX	<string>, 0-20 characters
VALUE	<string>, 0-40 characters
NOTE	<string>, 0-100 characters
UNITS	<string> Possible values: INCH | MM | MIL | MICRON | DEGREE | RADIAN | GRADIAN Options DEGREE, RADIAN, and GRADIAN are relevant to ANGLE only.
VIEW_UNITS	<string> Possible values: NO | YES
TOL_UP	<string>, 0-11 characters
TOL_DOWN	<string>, 0-11 characters
MERGE_TOL	<string> Possible values: NO | YES
OUTSIDE	<string> Possible values: NO | YES Not relevant to LOCATION.
UNDERLINE	<string> Possible values: NO | YES Relevant to RADIUS, DIAMETER, CHAMFER.
X	<x coordinate>, -100 INCH - 100 INCH
Y	<y coordinate>, -100 INCH - 100 INCH

Table 6-5. TEXT Array Structure

Field	Explanation
ANGLE	<float>, 0-360 degrees

Example of a Dimensions File

```
VERSION=1
UNITS=INCH

PARAMETERS {
    ID=0
    LINE_WIDTH=0.01
    POST_DECIMAL_DIST=3
    POST_DECIMAL_POS=3
    POST_DECIMAL_ANGLE=1
    FONT=STANDARD
    FONT_WIDTH=0.07
    FONT_HEIGHT=0.05
    EXT_OVERLEN=0.02
    EXT_OFFSET=0.01
    CENTER_MARKER_LEN=0.05
    BASELINE_SPACING=0.012
    ORIGIN_X=0
    ORIGIN_Y=0
    SCALE=100

    PAPER {
        ORIENTATION=PORTRAIT
        SIZE=A4
        WIDTH=8.3
        HEIGHT=11.7
        X=0
        Y=0
        MARGIN {
            TOP=0.1
            BOTTOM=0.1
            LEFT=0.2
            RIGHT=0.2
        }
        ACTIVE {
            X00=1
            Y00=1
            X11=8
            Y11=11
        }
        COLOR {
            FEATURE=RED
            DIMENS=BLUE
            DIMENS_TEXT=GREEN
            PROFILE=BLACK
            TEMPLATE=GRAY
        }
    }

}
```

```
DIMENSION {
    TYPE=HORIZONTAL
    PARAMETERS=0
    REF1X=0.1
    REF1Y=0.2
    REF2X=1.2
    REF2Y=0.3
    REF3X=0
    REF3Y=0
    LINE_PT_X=1.2
    LINE_PT_Y=0.7
    OFFSET=0
    ARROW_POS=AUTOMATIC
    MAGNIFY=1
    TO_ARC_CENTER=NO
    TWO_SIDED_DIAM=NO
    TEXT {
        VALUE=27.94
        PREFIX=
        SUFFIX=
        NOTE=
        UNITS=MM
        VIEW_UNITS=YES
        OUTSIDE=NO
        UNDERLINE=NO
        TOL_UP=
        TOL_DOWN=
        MERGE_TOL=NO
        X=0.65
        Y=0.7
        ANGLE=0
    }
}
DIMENSION {
    TYPE=VERTICAL
    PARAMETERS=0
    REF1X=0.3
    REF1Y=0.8
    REF2X=0.3
    REF2Y=0.1
    REF3X=0
    REF3Y=0
    LINE_PT_X=-0.1
    LINE_PT_Y=0.1
    OFFSET=0
    ARROW_POS=AUTOMATIC
    MAGNIFY=1
    TO_ARC_CENTER=NO
    TWO_SIDED_DIAM=NO
```

```
    TEXT {
        VALUE=17.78
        PREFIX=
        SUFFIX=
        NOTE=
        UNITS=MM
        VIEW_UNITS=YES
        OUTSIDE=YES
        UNDERLINE=NO
        TOL_UP=
        TOL_DOWN=
        MERGE_TOL=NO
        X=-0.1
        Y=-0.15231
        ANGLE=90
    }
```

```
}
```

```
DIMENSION {
    TYPE=PARALLEL
    PARAMETERS=0
    REF1X=0.3
    REF1Y=-0.2
    REF2X=0.6
    REF2Y=-0.5
    REF3X=0
    REF3Y=0
    LINE_PT_X=0.7
    LINE_PT_Y=-0.4
    OFFSET=0
    ARROW_POS=AUTOMATIC
    MAGNIFY=1
    TO_ARC_CENTER=NO
    TWO_SIDED_DIAM=NO
```

```
    TEXT {
        VALUE=10.776
        PREFIX=
        SUFFIX=
        NOTE=
        UNITS=MM
        VIEW_UNITS=YES
        OUTSIDE=YES
        UNDERLINE=NO
        TOL_UP=
        TOL_DOWN=
        MERGE_TOL=NO
        X=1.0021572
        Y=-0.702157
        ANGLE=45
    }
```

```
}
```

<layer_name>/profile (Outline Shape of Layer)

The layer profile is defined as a multi-island contour according to ODB++ contour definition. See “Geometric Entities” on page 34. Every island can contain several holes. The island must

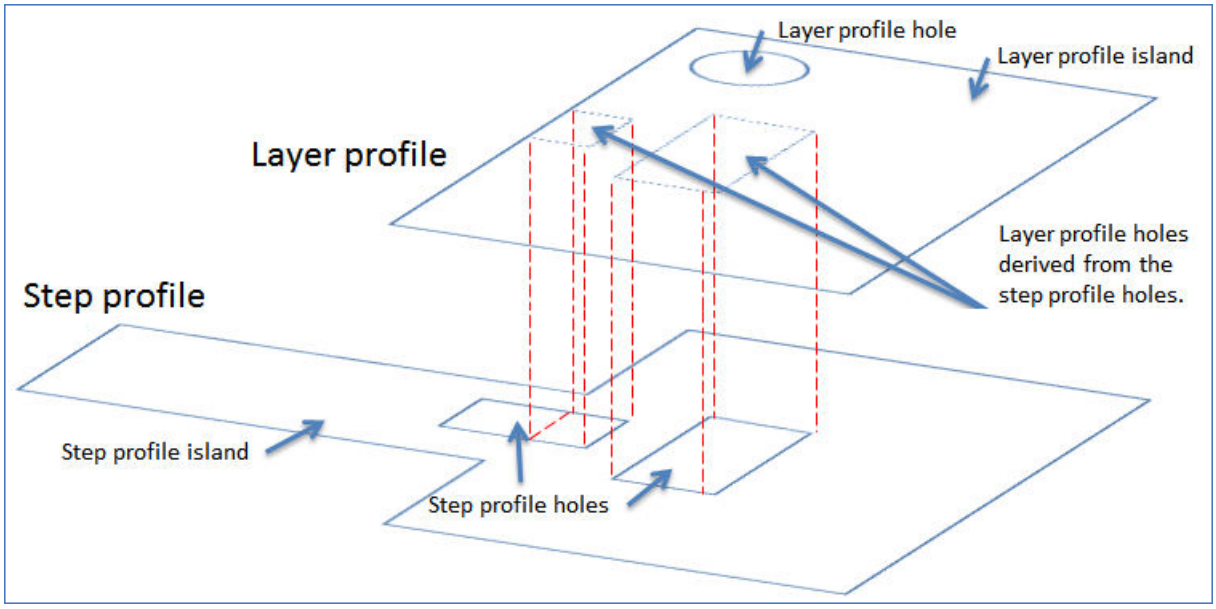
completely surround its holes, and the holes must not touch one another or intersect within the island. No other features are considered legal for the layer profile.

profile	
Type	Line Record Text
Compression	Yes
Path	*<product_model_name>/steps/<step_name>/layers/<layer_name>/profile*
UNITS	The UNITS directive is required.

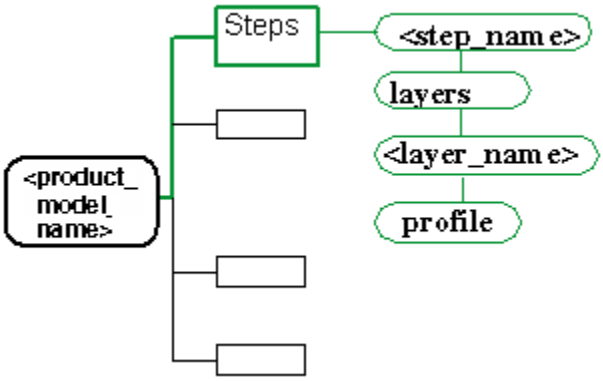
The layer profile is optional. A layer can contain one profile or no profile. The layer profile contains information relating to a single layer profile—the outline of the layer profile and its related attributes. All are optional.

The layer profile is primarily used for rigid-flex boards, where the rigid and flex parts of the board might have different physical limits. A layer profile is used to define these physical limits.

If a step profile hole is found within a layer profile island or if it intersects a layer profile island, the step profile hole is considered to exist along with the layer profile holes. It is not necessary to create the step profile holes in the layer profile. If the step profile hole intersects the layer profile island, the layer profile island shape—with the overlapping hole area of the step removed—defines a reduced layer island polygon shape that becomes the final layer island profile shape.



If no layer profile is defined, the layer profile is assumed to be equal to the step profile of the step in which the layer is defined. It is not necessary to define a layer profile if the step profile defines the shape of each layer. See "<step_name>/profile (Outline Shape of Step)" on page 92.



The *profile* file is a *features* file defining a single surface feature. Surface features are defined as described in "<layer_name>/features (Graphic Features)" on page 139.

Example of a Layer Profile File

```
UNITS=INCH
ID=58
#
#Num features
#
F 1
#
#Layer features
#
S P 0
OB 5.688950787402 7.475950787402 I
OS 5.688950787402 7.122049212598
OS 4.936950787402 7.122049212598
OS 4.936950787402 3.838950787402
OS 5.530950787402 3.838950787402
OS 5.530950787402 3.759950787402
OS 5.786950787402 3.759950787402
OS 5.786950787402 2.303049212598
OS 5.668950787402 2.303049212598
OS 5.668950787402 0.885950787402
OS 5.786950787402 0.885950787402
OS 5.786950787402 -0.290950787402
OS 4.566950787402 -0.290950787402
OS 4.566950787402 -0.554950787402
OS 4.173049212598 -0.554950787402
OS 4.173049212598 -0.290950787402
OS 1.731950787402 -0.290950787402
OS 1.731950787402 -0.633950787402
OS -0.542950787402 -0.633950787402
OS -0.542950787402 2.456950787402
OS 0.957049212598 2.456950787402
```

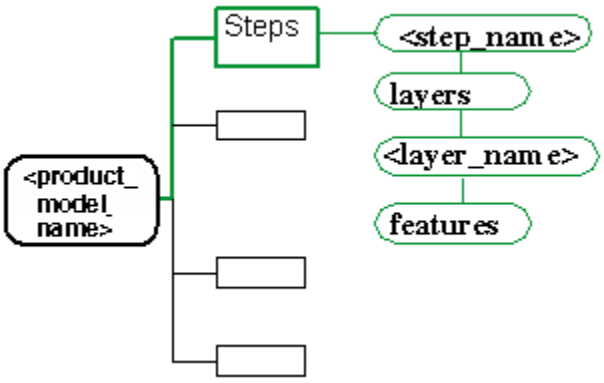
```
OS 0.957049212598 2.834950787402
OS 1.035049212598 2.834950787402
OS 1.035049212598 3.898049212598
OS 0.957049212598 3.898049212598
OS 0.957049212598 5.511950787402
OS 1.035049212598 5.511950787402
OS 1.035049212598 5.984049212598
OS 0.957049212598 5.984049212598
OS 0.957049212598 6.311049212598
OS -1.113950787402 6.311049212598
OS -1.113950787402 7.475950787402
OS 5.688950787402 7.475950787402
OE
SE
```

<layer_name>/features (Graphic Features)

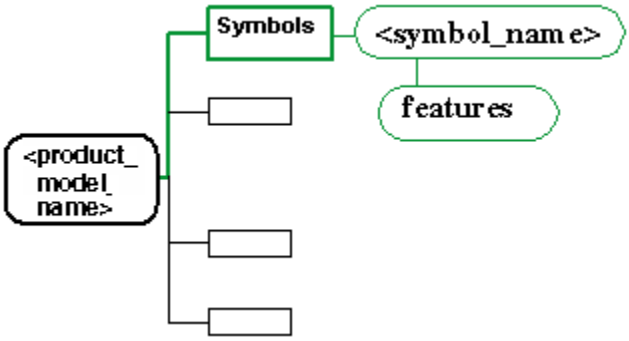
The features file contains definitions of graphic features.

features	
Type	Line Record Text
Compression	Yes
Path	*<product_model_name>/steps/<step_name>/layers/<layer_name>/features*
UNITS	The UNITS directive is required.

The *features* file contains most of the graphic information representing the physical board layers, NC drills and rout layers, and miscellaneous drawings. The *features* file should be present only for layers of these types.



User-defined symbols have a *features* file to describe their shape. See "symbols (User-Defined Symbols)" on page 41.



The files defining a step *profile* or a layer *profile* have the same format as a *features* file. The *profile* defines a single surface feature:

- "<step_name>/profile (Outline Shape of Step)" on page 92
- "<layer_name>/profile (Outline Shape of Layer)" on page 136

These topics are discussed:

Example of a features file

```
#
UNITS=MM
#
#Feature symbol names
#
$0 r120
$1 rect20x60 M
$2 rect3x5 I
$3 punch_target
#
#Feature attribute names
#
```

```
@0 .smd
@1 .nomenclature
@2 .test_point
@3 .geometry
@4 .pad_usage
#
#Feature attribute text strings
#
&0 9796334
&1 fid_0_0_0
&2 moire
&3 p115_115_115_095
#
#Layer features
#
P -0.198 1.62 16 P 0 3;3=2,4=0;ID=123456
P 0.118 1.62 16 P 0 3;3=25,4=0;ID=5678
L 3.834 -1.16 3.86728 -1.16 2 P 0 ;1,3=0
L 3.86728 -1.16 3.8782 -1.16485 2 P 0 ;1,3=0

...

S P 0;ID=123456
OB -0.013 2.427 I
OS -0.013 2.218
OS -0.263 2.218
OS -0.263 2.427
OS -0.219 2.427
OS -0.219 2.262
OS -0.057 2.262
OS -0.057 2.427
OS -0.013 2.427
OE
SE

...
```

Features File Sections

The *features* files have these sections:

Section	Explanation	Information
Header	Contains the UNITS directive, the unique ID (optional) and the number of features record (optional).	"Header Section" on page 142
Symbol Names	Contains the names of all the symbols used by the features in the file and corresponding serial numbers for reference by the feature records.	"Feature Symbol Names Section" on page 144
Attribute Lookup Tables	Contains a table of feature attribute names and a table of feature attribute string values. The feature definition records can assign attributes by referencing the index of an attribute and, for attributes of type TEXT, the index of the text string that is the value assigned to the attribute.	"Feature Attribute Lookup Tables" on page 145
Features	This is the main section of the features file. It describes all the features on the layer. Most features are represented by a single line in the file. Surface features might require multiple lines.	"L - Line Records" on page 146 "P - Pad Records" on page 147 "A - Arc Records" on page 149 "T - Text Records" on page 150 "B - Barcode Records" on page 154 "S - Surface Records" on page 156

Header Section

The header section of the *features* file contains the UNITS directive, the unique ID (optional) and the number of features record (optional).

Example of a features file header

```
UNITS=INCH
ID=216
#
#Num Features
#
F 1406

...
```

UNITS Directive

Features and coordinates are saved in the units in which they were created to avoid loss of precision due to rounding.

In every *features* file, there must be a UNITS directive that indicates the units of measurement to be applied to the features in the file. See "Units of Measurement" on page 18.

This is the format of the line: UNITS=<MM_or_INCH>.

- **UNITS=MM** — Indicates that metric units of measurement are used in the file.
- **UNITS=INCH** — Indicates that imperial units of measurement are used in the file.

If the UNITS directive does not exist in the *features* file, the units defined for the product model are assumed. If the *misc/info* file of the product model does not contain a UNITS directive, imperial units are assumed.

All coordinate values will be interpreted as inches or millimeters. Resize factors for user-defined symbols are interpreted as mils or microns. Exceptions are discussed where relevant.

This also applies to semi-standard symbols described in the *features* file. See "Feature Symbol Names Section" on page 144.

The UNITS directive in the *features* file sets the units to be used for symbols that do not specify units. In the example, units of measurement are set to metric.

```
#
UNITS=MM
#
#Feature symbol names
#
$0 r120
$1 rect20x60 M
$2 rect3x5 I
$3 punch_target

...
```

The dimensions of the symbols are interpreted accordingly:

Symbol	Units Used for Symbol Dimensions
r120	Standard round symbol with units in microns.
rect20x60 M	Standard rectangle symbol with units in microns. X is 20 microns and Y is 60 microns.
rect3x5 I	Standard rectangle with units in mils. X is 3 mils and Y is 5 mils.
punch_target	User-defined symbol. This must exist in the symbol directory of the product model.

ID Record Indicating a Unique ID (Optional)

This file can contain a unique ID.

This is the format of the line: ID=<id>.

See “Unique ID” on page 25.

F Record Indicating the Number of Features (Optional)

An optional line of the *features* file indicates the actual number of feature records that there are in the file. If available, this information could be used to determine what actions to take when reading the *features* file content.

This is the format of the line: F <number_of_features>.

Feature Symbol Names Section

This section lists the symbols used to construct lines, pads, or arcs defined in the file.

This is the format of the line: $<serial_num> <symbol_name> [<I|M>]

The parameter I or M can be added at the end of the line to indicate that the dimensions in this symbol name are expressed in imperial units or metric units.

Each line, pad, or arc feature definition record lists the index of the symbol used to construct that feature.

For a list of standard symbol names, see “Standard ODB++ Symbols” on page 169.

Example of the Feature Symbol Names Section

```
#
#Feature symbol names
#
$0 r120
$1 rect20x60 M
$2 rect3x5 I
$3 punch_target
```

Feature Attribute Lookup Tables

This section of the file contains a table of feature attribute names and a table of all feature attribute values that are strings.

- **Feature attribute names** — Each record contains an at sign (@), and index, and an attribute name. For example, @1 .nomenclature.
- **Feature attribute text strings** — Each record contains an ampersand (&), an index, and a text string that can be used as the value of an attribute of type TEXT. For example, &2 moire.

System attributes that can be placed in this table are listed with Entity = Feature in the attribute description. See “System Attributes” on page 189

This section must be placed before the layer features section of the file.

The feature definition records in the file can assign feature attributes by referencing the index of a feature attribute and, for attributes of type TEXT, the index of the text string that is the value assigned to the attribute.

This is an example of a feature attribute lookup table, and a feature definition record that assigns attributes.

```
#
#Feature attribute names
#
@0 .smd
@1 .nomenclature
@2 .test_point
@3 .geometry
@4 .pad_usage
#
#Feature attribute text strings
#
&0 9796334
&1 fid_0_0_0
&2 moire
&3 p115_115_115_095
#
#Layer features
#
```

```
P -0.198 1.62 16 P 0 3;3=2,4=0;ID=123456
```

The string 3=2,4=0 in the feature definition record in the example should be interpreted as assigning these attributes to the feature:

Assignment in the Feature Definition Record	Line in the Table of Attribute Names	Attribute Type	Line in the Table of Attribute Text Strings	Result
3=2	@3 .geometry	TEXT	&2 moire	.geometry = moire
4=0	@4 .pad_usage	OPTION	-	.pad_usage = toeprint Options for the .pad_usage attribute are: toeprint; via; g_fiducial; l_fiducial; tooling_hole, so 0 indicates toeprint.

L - Line Records

For line (L) records:

L <xs> <ys> <xe> <ye> <sym_num> <polarity> <dcode>;<atr>=<value>,...;ID=<id>

Line Parameter	Explanation
type	L
xs, ys	Start point.
xe, ye	End point.
sym_num	The index of the feature used to construct the line, in the feature symbol names section. See “Feature Symbol Names Section” on page 144.
polarity	**P** for positive, **N** for negative.
dcode	Gerber dcode number or Excellon tool number (0 if not defined).
atr	An attribute number, referencing an attribute from the feature attribute names section. See “Feature Attribute Lookup Tables” on page 145.

Line Parameter	Explanation
value	An attribute value that depends on the type of attribute: • **BOOLEAN** — No value is necessary. If the index of the attribute is listed, the attribute is set to TRUE. • **FLOAT** or **INTEGER** — A number. • **OPTION** — An option number. • **TEXT** — A number referencing the feature attribute text strings section. See “Feature Attribute Lookup Tables” on page 145.
ID=<id>	Assigns a unique identifier to the feature. See “Unique ID” on page 25.

Example of Line records in a features file

```
L 5.15 0.35 5.125 0.325 0 P 0
```

P - Pad Records

For pad (P) records:

P <x> <y> <apt_def> <polarity> <dcode> <orient_def>;<atr>=<value>,...;ID=<id>

Pad Parameter	Explanation
x, y	Center point.
apt_def	The index, in the feature symbol names section, of the symbol to be used to draw the pad. This value can be expressed in one of these ways: • **<sym_num>** — The symbol with index **<sym_num>** is used as is. • **-1 <sym_num> <resize_factor>** — The symbol with index <sym_num> is enlarged or shrunk by factor <resize_factor>. The resize factor is expressed in thousandths of the units being used (mils or microns). See “Feature Symbol Names Section” on page 144.
polarity	**P** for positive, **N** for negative
dcode	Gerber dcode number or Excellon tool number (0 if not defined)

Pad Parameter	Explanation
orient_def	Pad orientation. This value is expressed as: **0|1|2|3|4|5|6|7|8<rotation>|9<rotation>** If the first number of orientation definition is an integer from **0** through **7**, it is legacy data and will be handled accordingly. These values are read from legacy data to maintain backward compatibility. **0** — 0 degrees rotation, no mirror. **1** — 90 degrees rotation, no mirror. **2** — 180 degrees rotation, no mirror. **3** — 270 degrees rotation, no mirror. **4** — 0 degrees rotation, mirror in x-axis. **5** — 90 degrees rotation, mirror in x-axis. **6** — 180 degrees rotation, mirror in x-axis. **7** — 270 degrees rotation, mirror in x-axis. If the first number is 8 or 9, it is a two number definition, with the following number representing rotation. **8** — Any angle rotation, no mirror. **9** — Any angle rotation, mirror in x-axis. When pads are both rotated and mirrored, they are rotated first, and then mirrored along the x-axis.
atr	An attribute number, referencing an attribute from the feature attribute names section. See “Feature Attribute Lookup Tables” on page 145.
value	An attribute value that depends on the type of attribute: • **BOOLEAN** — No value is necessary. If the index of the attribute is listed, the attribute is set to TRUE. • **FLOAT** or **INTEGER** — A number. • **OPTION** — An option number. • **TEXT** — A number referencing the feature attribute text strings section. See “Feature Attribute Lookup Tables” on page 145.
ID=<id>	Assigns a unique identifier to the feature. See “Unique ID” on page 25.

Example of Pad records in a features file

Special pad const_1 at location x=1.0, y=2.0 positive, with dcode 4, is used as an example for different transformations:

```
#
#Feature symbol names
#
$0 const_1
#
#Feature attribute names
#
@0 .smd
1 .nomenclature
```

```
2 .test_point
3 .geometry
4 .pad_usage
.....
#

P 1.0 2.0 0 P 4 1;;ID=3948
P 1.0 2.0 0 P 4 8 30.0;;ID=7205
P 1.0 2.0 -1 0 0.02 P 4 1;;ID=7421
P 1.0 2.0 -1 0 0.02 P 4 8 30.0;;ID=9630
```

In the example,

P 1.0 2.0 0 P 4 1 is rotated by 90 degrees (orient_def = 1).

P 1.0 2.0 0 P 4 8 30.0 is rotated by 30 degrees (orient_def = 8 30.0).

P 1.0 2.0 -1 0 0.02 P 4 1 is rotated by 90 degrees, and resized by 0.02 mil (apt_def = -1 0 0.02 and orient_def = 1).

P 1.0 2.0 -1 0 0.02 P 4 8 30.0 is rotated by 30, and resized by 0.02 mil (apt_def = -1 0 0.02 and orient_def = 8 30.0).

A - Arc Records

For arc (A) records:

A <xs> <ys> <xe> <ye> <xc> <yc> <sym_num> <polarity> <dcode> <cw>;<atr>=<value>,...;ID=<id>

Arc Parameter	Explanation
xs, ys	Start point. When the start and end-point of an arc coincide, it is considered a 360 degree arc. There are no single-point arcs in ODB++ format.
xe, ye	End point. When the start and end-point of an arc coincide, it is considered a 360 degree arc. There are no single-point arcs in ODB++ format.
xc, yc	Center point.
sym_num	The index, in the feature symbol names section, of the symbol to be used to draw the arc. See “Feature Symbol Names Section” on page 144.
polarity	**P** for positive, **N** for negative.
dcode	Gerber dcode number or Excellon tool number (0 if not defined).
cw	**Y** for clockwise, **N** for counter clockwise.
atr	An attribute number, referencing an attribute from the feature attribute names section. See “Feature Attribute Lookup Tables” on page 145.

Arc Parameter	Explanation
value	An attribute value that depends on the type of attribute: • **BOOLEAN** — No value is necessary. If the index of the attribute is listed, the attribute is set to TRUE. • **FLOAT** or **INTEGER** — A number. • **OPTION** — An option number. • **TEXT** — A number referencing the feature attribute text strings section. See “Feature Attribute Lookup Tables” on page 145.
ID=<id>	Assigns a unique identifier to the feature. See “Unique ID” on page 25.

Example of Arc records in a features file

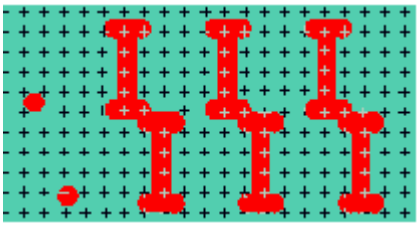
```
A 1.49401033 2.53683297 0.8389936 2.48563819 1.17858858 2.35659203 2 P 0 Y
```

T - Text Records

For text (T) records:

T <x> <y> <font> <polarity> <orient_def> <xsize> <ysize> <width_factor> <text> <version>;<atr>=<value>,...;ID=<id>

Text Parameter	Explanation
x, y	Text location (bottom left of first character for 0 orientation).
font	Font name.
polarity	**P** for positive, **N** for negative.

Text Parameter	**Explanation**
orient_def	Text orientation. This value is expressed as: **0|1|2|3|4|5|6|7|8<rotation>|9<rotation>** If the first number of orientation definition is an integer from **0** through **7**, it is legacy data and will be handled accordingly. These values are read from legacy data to maintain backward compatibility. **0** — 0 degrees rotation, no mirror. **1** — 90 degrees rotation, no mirror. **2** — 180 degrees rotation, no mirror. **3** — 270 degrees rotation, no mirror. **4** — 0 degrees rotation, mirror in x-axis. **5** — 90 degrees rotation, mirror in x-axis. **6** — 180 degrees rotation, mirror in x-axis. **7** — 270 degrees rotation, mirror in x-axis. If the first number is 8 or 9, it is a two number definition, with the following number representing rotation. **8** — Any angle rotation, no mirror. **9** — Any angle rotation, mirror in x-axis. When pads are both rotated and mirrored, they are rotated first, and then mirrored along the x-axis.
xsize, ysize	character size—**xsize**=character width including the following space; **ysize**=height of capital character
width_factor	width of character segment (in units of 12 mils); that is, 1 = 12 mils, 0.5 = 6 mils
text	Text string enclosed in single quotes. The text string can contain dynamic values listed in “Dynamic Text Variables” on page 153.
version	Defines how the first character of a text string is to be placed relative to the insertion point. **0** — Lower left corner of the text limit box coincides with the insertion point. **1** — Lower left corner of the widest character coincides with the insertion point. (The widest character is usually an M or a W, by which smaller letters are centered.)  The first row of text has this parameter set to **0**. The second row has it set to **1**.
atr	An attribute number, referencing an attribute from the feature attribute names section. See “Feature Attribute Lookup Tables” on page 145.

Text Parameter	Explanation
value	An attribute value that depends on the type of attribute: • **BOOLEAN** — No value is necessary. If the index of the attribute is listed, the attribute is set to TRUE. • **FLOAT** or **INTEGER** — A number. • **OPTION** — An option number. • **TEXT** — A number referencing the feature attribute text strings section. See “Feature Attribute Lookup Tables” on page 145.
ID=<id>	Assigns a unique identifier to the feature. See “Unique ID” on page 25.

Example of Text records in features file

```
#Feature attribute names
#
#Layer features
#
T 4.033375 6.377506 standard P 8 0 0.2 0.2 2.00000 '$$DATE-DDMMYY - Coupon
STD500CV' 1
```

The fields of the record indicate this information:

Parameter	Value in Example	Explanation
Record type	T	The record is text.
Location	4.033375 6.377506	The x, y coordinates at the bottom left of the first character of the text string.
Font	standard	The font is standard
Polarity	P	The polarity is positive.
Orientation	8 0	The text is rotated but not mirrored (8) with 0 degrees of rotation.
Size	0.2 0.2	The x, y size of text string (by layer units).
Width factor	2.00000	The width of character segment is 24 mils.
Text string	‘$$DATE-DDMMYY - Coupon STD500CV’	This text string contains a variable, and a string that contains a space. See “Dynamic Text Variables” on page 153.
Version	1	The lower left corner of the widest character coincides with the insertion point.

Dynamic Text Variables

A text string can include dynamic text variables that change textual value according to status or condition. For example, a variable for the date always displays the current date, as set by the system. Dynamic text variables can be located anywhere within a text string. They are distinguished by the prefix $$, as in $$DATE for the current date.

An example of a dynamic text record would be:

```
T 11.890963 13.697185 standard P 8 0 5.08 5.08 2.00000 '$$DATE-MMDDYY' 1
```

Where the variable $$DATE-MMDDYY displays the date as 05/30/14.

The ODB++ format supports these dynamic text variables:

Table 6-6. Dynamic Text Variables

Variable	Descriptions
DATE-MMDDYY	Date expressed as month/day/year (short).
DATE-DDMMYY	Date expressed as day/month/year (short).
DATE-MMDDYYYY	Date expressed as month/day/year.
DATE-DDMMYYYY	Date expressed as day/month/year.
DD	Day of month (01-31).
WEEK-DAY	Day of week (Sunday-Saturday).
MM	Month of year (01-14).
YY	Year (14).
YYYY	Year (2014).
WW	Week of year (01-52).
TIME	Current time.
JOB	Name of current product module.
STEP	Name of step where text is placed.
LAYER	Name of layer where text is placed.
X	Bottom left x-coordinate of where text is placed (inch).
Y	Bottom left y-coordinate of where text is placed (inch).
X_MM	Bottom left x-coordinate of where text is placed (mm).
Y_MM	Bottom left y-coordinate of where text is placed (mm).

B - Barcode Records

For barcode (B) records:

B <x> <y> <barcode> <font> <polarity> <orient_def> E <w> <h> <fasc> <cs> <bg> <astr> <astr_pos> <text>;<atr>=<value>,...;ID=<id>

Barcode Parameter	**Explanation**
x, y	Text location (bottom left of first character for 0 orientation.
barcode	Barcode name (currently must be UPC39).
font	Font name.
polarity	**P** for positive, **N** for negative.
orient_def	Text orientation. This value is expressed as: **0|1|2|3|4|5|6|7|8<rotation>|9<rotation>** If the first number of orientation definition is an integer from **0** through **7**, it is legacy data and will be handled accordingly. These values are read from legacy data to maintain backward compatibility. **0** — 0 degrees rotation, no mirror. **1** — 90 degrees rotation, no mirror. **2** — 180 degrees rotation, no mirror. **3** — 270 degrees rotation, no mirror. **4** — 0 degrees rotation, mirror in x-axis. **5** — 90 degrees rotation, mirror in x-axis. **6** — 180 degrees rotation, mirror in x-axis. **7** — 270 degrees rotation, mirror in x-axis. If the first number is 8 or 9, it is a two number definition, with the following number representing rotation. **8** — Any angle rotation, no mirror. **9** — Any angle rotation, mirror in x-axis. When pads are both rotated and mirrored, they are rotated first, and then mirrored along the x-axis.
E	A constant value (reserved for future use).
w	Element width (inches or mm).
h	Barcode height (inches or mm).
fasc	**Y** for full ASCII, **N** for partial ASCII.
cs	**Y** for checksum, **N** for no checksum.
bg	**Y** for inverted background, **N** for no background.

Barcode Parameter	Explanation
astr	**Y** for an addition of a text string; **N** for only the barcode  The first barcode with Iver=0 text has this parameter set to Y; the second has this parameter set to N.
astr_pos	**T** for adding the string on top, **B** for adding the string on the bottom.
text	Text string, enclosed in single quotes, centered on top or bottom of the barcode. The text string can contain dynamic values listed in “Dynamic Text Variables” on page 153.
atr	An attribute number, referencing an attribute from the feature attribute names section. See “Feature Attribute Lookup Tables” on page 145.
value	An attribute value that depends on the type of attribute: • **BOOLEAN** — No value is necessary. If the index of the attribute is listed, the attribute is set to TRUE. • **FLOAT** or **INTEGER** — A number. • **OPTION** — An option number. • **TEXT** — A number referencing the feature attribute text strings section. See “Feature Attribute Lookup Tables” on page 145.
ID=<id>	Assigns a unique identifier to the feature. See “Unique ID” on page 25.

Example of Barcode record in features file

```
UNITS=INCH
#
#Feature attribute names
#
@0 .nomenclature
#
#Layer features
#
B -1.73493967 0.25060236 UPC39 standard P 8 0 E 0.008 0.2 Y N Y Y T
'Barcode 1234567890';0
```

S - Surface Records

A surface definition must have at least one polygon defined. The surface usually consists of multiple records.

```
S <polarity> <dcode>;<atr>=<value>,...;ID=<id>
<polygon 1>
<polygon n>
SE
```

These are the parameters for the S line:

Surface Parameter	Explanation
polarity	**P** for positive, **N** for negative
dcode	Gerber dcode number or Excellon tool number (0 if not defined)
atr	An attribute number, referencing an attribute from the feature attribute names section. See “Feature Attribute Lookup Tables” on page 145.
value	An attribute value that depends on the type of attribute: • **BOOLEAN** — No value is necessary. If the index of the attribute is listed, the attribute is set to TRUE. • **FLOAT** or **INTEGER** — A number. • **OPTION** — An option number. • **TEXT** — A number referencing the feature attribute text strings section. See “Feature Attribute Lookup Tables” on page 145.
ID=<id>	Assigns a unique identifier to the feature. See “Unique ID” on page 25.

The S line is followed by a list of polygon definitions. Each polygon is a collection of segments (lines without width) and curves (arcs without width). The polygons must meet the restrictions described in “Surfaces” on page 32.

It is recommended that each polygon be represented as a single island, because a multi-island polygon is electrically disconnected. As a single feature, it should be connected to a single net.

A polygon begins with an **OB** command, contains **OS** (segment) or **OC** (arc) commands, and ends with an **OE** command:

OB <xbs> <ybs> <poly_type>
OS <x> <y>
...
OC <xe> <ye> <xc> <yc> <cw>
...
OE

The last OS or OC coordinate should be the same as the OB coordinate.

These are the parameters of the OB (begin) line:

OB Parameter	Explanation
xbs, ybs	Beginning point of the polygon.
poly_type	**I** for island, **H** for hole.

These are the parameters of the OS (segment) lines:

OS Parameter	Explanation
x, y	Segment end point. The previous polygon point is the start point.
cw	**Y** for clockwise, **N** for counter-clockwise.

These are the parameters of the OC (arc) lines:

OC Parameter	Explanation
xe, ye	Curve end point. The previous polygon point is the start point.
xc, yc	Curve center point.
cw	**Y** for clockwise, **N** for counter-clockwise.

Example of Surface records in the features file

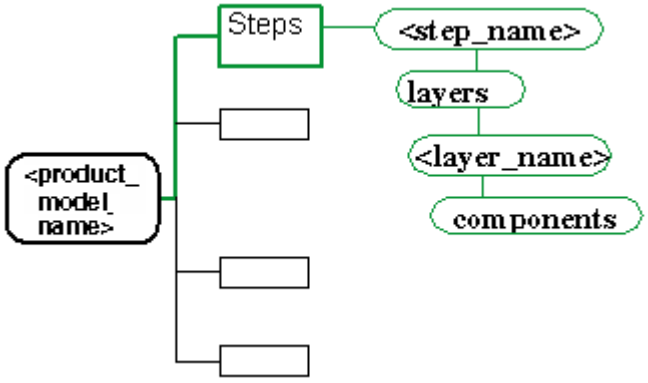
```
S P 0
OB 5.36 3.525 I
OS 5.36 3.809
OS 5.501 3.809
OS 5.501 3.735
OS 5.431 3.735
OS 5.431 3.525
OS 5.36 3.525
OE
OB 5.593 3.525 I
OS 5.593 3.73
OS 5.75 3.73
OS 5.75 3.525
OS 5.593 3.525
OE
SE
```

<layer_name>/components (Components)

Each layer of type component has a unique name: comp_+_top or comp_+_bot. There can be only one of each layer name within a product model—one for the top components and one for the bottom components.

components	
Type	Line Record Text
Compression	Yes
Path	*<product_model_name>/steps/<step_name>/layers/<layer_name>/components*
UNITS	The UNITS directive is required.

Each component layer has a *components* file that contains information about components placed on the layer. The components file contains references to the file *<step_name>/eda/data*. See "<step_name>/eda (Electronic Design Automation)" on page 105.



The components file contains these sections:

Example

```
#
UNITS=INCH
#
#Component attribute names
#
@0 .no_pop
@1 .comp_ignore
@2 .desc1
@3 .comp_mount_type
```

```
@4 .comp_height
#
#Component attribute text strings
#
&0 Internally created connector
&1 Socket
&2 ball
&3 gw

# CMP 0
CMP 0 1.55 8.9 270.0 N J10 2300046 ;0,2=0,3=1
PRP PART_NUMBER '1000045'
PRP VALUE 'FAN1654'
PRP PART_NUMBER '1000045'
TOP 0 6.568 5.2296 0.0 N 8 163 1
TOP 1 6.568 5.204 0.0 N 8 164 2
TOP 2 6.568 5.1784 0.0 N 13 66 3
TOP 3 6.568 5.1528 0.0 N 2 779 4
TOP 4 6.568 5.1272 0.0 N 2 780 5
TOP 5 6.568 5.1016 0.0 N 13 67 6
TOP 6 6.568 5.076 0.0 N 8 165 7
TOP 7 6.568 5.0504 0.0 N 2 781 8
TOP 8 6.798 5.0504 0.0 N 1494 0 9
TOP 9 6.798 5.076 0.0 N 13 68 10
TOP 10 6.798 5.1016 0.0 N 12 1 11
TOP 11 6.798 5.1272 0.0 N 1143 1 12
TOP 12 6.798 5.1528 0.0 N 2 782 13
TOP 13 6.798 5.1784 0.0 N 1144 1 14
TOP 14 6.798 5.204 0.0 N 8 166 15
TOP 15 6.798 5.2296 0.0 N 1495 0 16
TOP 16 6.683 5.14 0.0 N 2 783 17
#
# BOM DATA
CPN M62X971EL7
PKG
IPN
DSC  16-ETSSOP FAN1654
VPL_VND MOTOROLA
VPL_MPN MC74HC259ADT
#COMP_EXT_PTY 1
VND MOTOROLA INC.
MPN 0 Y MC74HC259ADT
# CMP 1
CMP 3 5.017 1.791 180.0 N C400 ??? ;3=1
PRP VOLTAGE '6.3V'
PRP VALUE '22UF'
TOP 0 5.072 1.791 180.0 N 87 2 1
TOP 1 4.962 1.791 180.0 N 2 245 2
#
# BOM DATA
CPN IZZ5FH7FX7
PKG
IPN
DSC  B3258 22UF 6.3V
VPL_VND
VPL_MPN
#COMP_EXT_PTY 1
VND KOA
```

```
MPN 0 Y TMC1AABA106K
```

Component Attribute Lookup Tables

This section of the file contains a table of component attribute names and a table of all component attribute values that are strings.

- **Component attribute names** — Each record contains an at sign (@), and index, and an attribute name. For example, @1 .comp_ignore.
- **Feature attribute text strings** — Each record contains an ampersand (&), an index, and a text string that can be used as the value of an attribute of type TEXT. For example, &3 gw.

System attributes that can be placed in this table are listed with Entity = Comp. in the attribute description. See “System Attributes” on page 189.

This section must be placed before the first CMP record of the file.

The CMP records in the file can assign component attributes by referencing the index of a component attribute and, for attributes of type TEXT, the index of the text string that is the value assigned to the attribute.

This is an example of a component attribute lookup table, and a CMP record that assigns attributes.

```
#Component attribute names
#
@0 .no_pop
@1 .comp_ignore
@2 .desc1
@3 .comp_mount_type
@4 .comp_height
#
#Component attribute text strings
#
&0 Internally created connector
&1 Socket
&2 ball
&3 gw

# CMP 0
CMP 0 1.55 8.9 270.0 N J10 2300046 ;0,2=0,3=1
```

The string 0,2=0,3=1 in the CMP record in the example should be interpreted as assigning these attributes to the component:

Assignment in the CMP Record	Line in the Table of Attribute Names	Attribute Type	Line in the Table of Attribute Text Strings	Result
0	@0 .no_pop	BOOLEAN	-	.no_pop = TRUE
2=0	@2 .desc1	TEXT	&0 Internally created connector	.desc1 = Internally created connector
3=1	@3 .comp_mount_type	OPTION	-	.comp_mount_type = SMT Options for the .comp_mount_type attributes are: Other, SMT, THMT, PressFit, so 1 indicates SMT.

Component Description Record

After the attribute header, components are listed in order, using these types of records:

Record Type	Explanation
CMP	Definition of a component. See “CMP—Component Record” on page 161.
PRP	Property of a component. See “PRP—Property Record” on page 162.
TOP	Toeprint of a component. See “TOP—Toeprint Record” on page 163.

CMP—Component Record

This record contains a definition of a component. Each CMP line is followed by 0 or more property (PRP) records, and 0 or more TOP records. The pkg_ref field of the CMP record references the sequential order of the PKG records in the *eda/data* file. PKG order is referenced using numbers starting from 0 and up. See “PKG—Package Record” on page 112.

The CMP component record takes the form of:

CMP <pkg_ref> <x> <y> <rot> <mirror> <comp_name> <part_name>; <attributes>;ID=<id>

Field	Explanation
pkg_ref	Reference number of the package in the *eda/data* file. See "<step_name>/eda (Electronic Design Automation)" on page 105.
x, y	Board location of the component in inches or mm.
rot	Rotation of the component, in degrees, clockwise.
mirror	**N** for not mirrored, **M** for mirrored.
comp_name	Unique reference designator (component name); a single string of ASCII characters without spaces. Upper case and lower case characters are not equivalent.
part_name	Part identification is a single string of ASCII characters without spaces.
attributes	A comma separated list of attribute assignments. The attribute is indicated by its index in the attribute name lookup table. • **BOOLEAN** — n indicating that attribute n is set. • **OPTION** — n=m indicating that attribute n has value m. • **INTEGER** — n=i indicating that attribute n has value i. • **FLOAT** — n=f indicating that attribute n has value f. • **TEXT** — n=s indicating that attribute n has the value associated with index s in the attribute text string lookup table. See "Component Attribute Lookup Tables" on page 160.
ID=<id>	Assigns a unique identifier to the component. See "Unique ID" on page 25.

PRP—Property Record

This record represents a property of the component. A property consists of a name, a string value, and 0 or more floating numbers. It takes the form of:

PRP <name> '<value>' n1 n2 ...,

Field	Explanation
name	Name of the property.
value	String of the property (between quotes; no restrictions).
n1,n2,...	Floating numbers to be kept in the property.

TOP—Toeprint Record

This record contains a definition of a toeprint of a component.

TOP <pin_num> <x> <y> <rot> <mirror> <net_num> <subnet_num> <toeprint_name>,

Field	Explanation
pin_num	Pin number inside the package of the component.
x, y	Board location of the pin in inches or mm.
rot	Rotation of the component, in degrees, clockwise.
mirror	N for not mirrored, M for mirrored.
net_num	Number of net in the *eda/data* file. The net_num used in the TOP record corresponds to the sequence of the NET records in the *eda/data* file. The first NET record is net_num 0, the second is net_num 1 and so on.
subnet_num	Number of subnet (SNT record) in the referenced net.
toeprint_name	Name of the toeprint.

BOM Description Records

The component BOM DATA section contains BOM information on components.

Parameter	Description
CPN	Customer part number.
PKG	Package name.
IPN	Internal part number.
DSC	Unlimited number of descriptions.
VPL_VND	Manufacturer from an external vendor part library corresponding to original vendor (as determined in BOM Validation).
VPL_MPN	MPN from an external vendor part library corresponding to original MPN (as determined in BOM Validation).
VND	Manufacturer (vendor) name.
MPN	Manufacturer part number.
Qualify	Whether the part (vendor+mpn) is qualified for production: **-1** — Not qualified. **0** — Unknown. **1** — Qualified.

Parameter	Description
Chosen	Whether the part is chosen from among the alternate parts for the CPN. Only one part can be a chosen part. **0**=not chosen, **1**=chosen.

The MPN lines contain the following parameters separated by spaces: **qualify chosen MPN**. For example, **MPN O Y 4N35S** for a CPN component whose qualification is unknown (0), that is the 'chosen' component (**1**), with a manufacturer part number of **4N35S**. These sections repeat for all the alternate parts of the CPN: **VPL_VND**, **VPL_MPN**, **VND**, and **MPN**.

Example of Alternate Parts

This is an example of alternate parts defined for the same CPN. The first is chosen, and the third is not qualified.

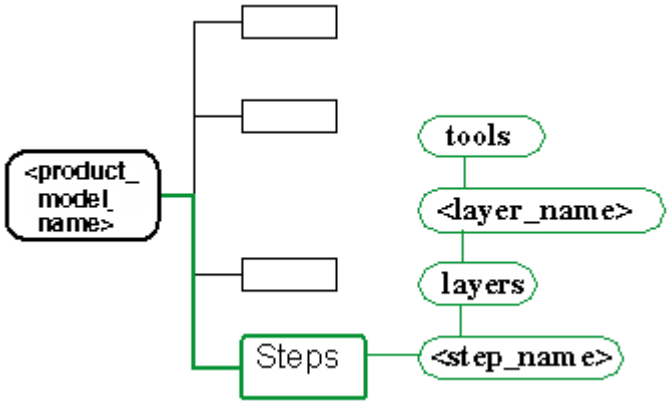
```
CPN 69K9KMH12B
LNFILE 51 name_of_bom_file.txt
VPL_MPN MAX5156ACEE
VPL_VND MAXIM
MPN MAX5156ACEE
LNFILE 51 name_of_bom_file.txt
VND MAXIM
LNFILE 51 name_of_bom_file.txt
QLF 0
CHS 1
PRIORITY 1

CPN 69K9KMH12B
LNFILE 51 name_of_bom_file.txt
VPL_MPN CY22150FC
VPL_VND CYPRESS
MPN CY22150FC
LNFILE 52 name_of_bom_file.txt
VND CYPRESS
LNFILE 52 name_of_bom_file.txt
QLF 0
CHS 0
PRIORITY 2

CPN 69K9KMH12B
LNFILE 51 name_of_bom_file.txt
VPL_MPN AT93C66-10SC
VPL_VND ATMEL
MPN AT93C66-10SC
LNFILE 53 name_of_bom_file.txt
VND ATMEL
LNFILE 53 name_of_bom_file.txt
QLF -1
CHS 0
PRIORITY 0
```

<layer_name>/tools (Drill Tools)

This file contains the tools table of a drill layer, initially created during input.

tools (Drill Tools)	
Type	Structured Text
Compression	None
Path	*<product_model_name>/steps/<step_name>/layers/<layer_name>/tools*
UNITS	The UNITS directive is required.

The file contains global parameters and a **TOOLS** array structure.



Global Parameters for Drill Tools

These are the global parameters:

Table 6-7. Global Parameters for Drill Tools

Field	Explanation
THICKNESS	Board thickness (mils or mm) (Obsolete). Where it is necessary to describe board thickness, it is derived from the step attribute .board_thickness. See “ODB++ System Attributes - Alphabetical List” on page 189.
USER_PARAMS	Free text.

Tools array structure

This is the tools array structure:

Table 6-8. TOOLS Array Structure

Field	Explanation
NUM	Tool number.
TYPE	One of these values: PLATED, NON_PLATED, VIA.
TYPE2	Legal values depend on the value of TYPE: • **TYPE=PLATED** — STANDARD or PRESS_FIT • **TYPE=NON_PLATED** — STANDARD • **TYPE=VIA** — STANDARD, PHOTO, or LASER (Default=STANDARD).
MIN_TOL, MAX_ TOL	Allowed tolerances (mils or microns).
BIT	Drill bit string.
FINISH_SIZE	Required drill size in the finished board. (mils or microns). If this value was not set, the value should be -1.
DRILL_SIZE	Drill tool size used by the drilling machine to achieve the required finished hole size after production of the board (mils or microns) (optional).

Example of a tools file

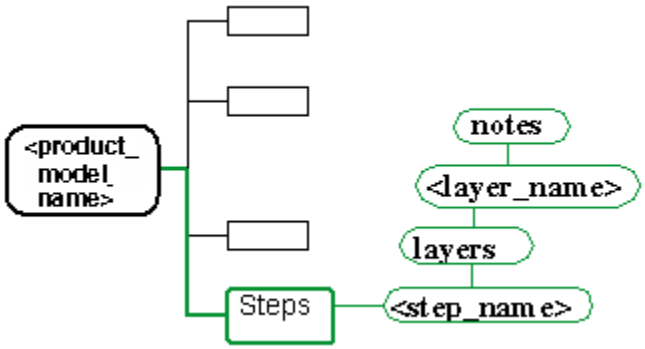
```
UNITS=INCH
THICKNESS=62.5
USER_PARAMS=method25
TOOLS {
   NUM=1
   TYPE=VIA
   TYPE2=STANDARD
   MIN_TOL=0
   MAX_TOL=0
   BIT=
   FINISH_SIZE=11.5
   DRILL_SIZE=13.5
}
TOOLS
{
   NUM=2
   TYPE=PLATED
   TYPE2=STANDARD
   MIN_TOL=0
   MAX_TOL=0
   BIT=
   FINISH_SIZE=15
   DRILL_SIZE=19
}
```

<layer_name>/notes (Electronic Notes)

The *notes* file contains notes added by the user.

notes (Electronic Notes)	
Type	Line Record Text
Compression	None
Path	*<product_model_name>/steps/<step_name>/layers/<layer_name>/notes*
UNITS	The UNITS directive is not relevant for the *notes* file.

This file contains all the notes added by the user to the graphic layer.



Example of a notes file

866467418,moshik,2.03807,-1.22818,,,,,,First line\nSecond line

Each line in the notes file has this format:

<time>,<user>,<x>,<y>,,,,,,<note>, where:

Field	**Explanation**
time	Last date updated. Uses a standard UNIX time code method that is the number of seconds that have passed since January 1st, 1970.
user	User who most recently updated the note.
x, y	Graphic location expressed in accordance with UNITS=MM|INCH.
note	Up to four lines of text where the \n character describes the line break

Appendix 1
Standard ODB++ Symbols

A symbol is one of the geometric primitives supported by ODB++ described here, or a combination of them. They are defined by the parameters: width, height, radius, diameter, spokes, gap, angle, size and corner. The corners of rounded or chamfered rectangles or thermals are specified in ascending order—moving counter-clockwise—starting from the top-right corner. For an example of a symbol with chamfered corners, see "Rounded/Chamfered Rectangles" on page 185.

All symbols described in this section are standard (system) symbols. User defined symbols are defined as described in "symbols (User-Defined Symbols)" on page 41.

Basic Standard Symbols

These are the basic standard symbols.

Table 1-1. Basic Standard Symbols

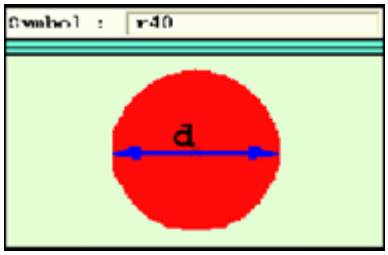
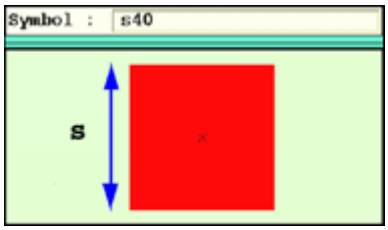
Symbol Name	Example	Parameters
Round		**r<d>** **d** — Circle diameter
Square		**s<s>** **s** — Square side

Table 1-1. Basic Standard Symbols

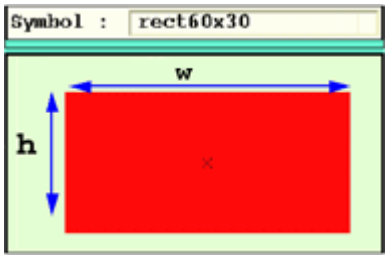
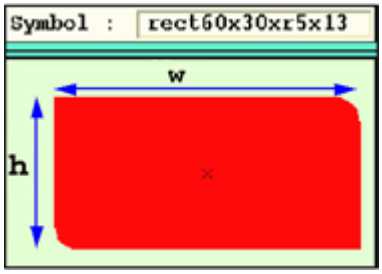
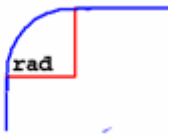
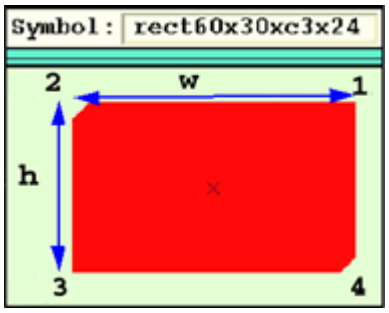
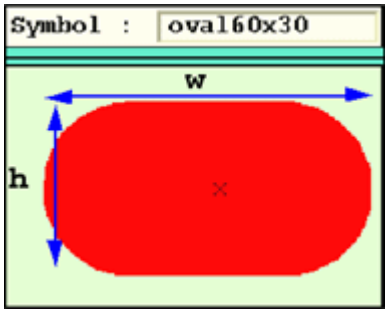
Symbol Name	Example	Parameters
Rectangle		**rect<w>x<h>** **w** — Rectangle width **h** — Rectangle height
Rounded Rectangle		**rect<w>x<h>xr<rad>x<corners>** **w** — Rectangle width **h** — Rectangle height **rad** — Corner radius **corners** — Indicates which corners are rounded. x<corners> is omitted if all corners are rounded.   See “Rounded/Chamfered Rectangles” on page 185.
Chamfered Rectangle		**rect<w>x<h>xc<rad>x<corners>** **w** — Rectangle width **h** — Rectangle height **rad** — Corner radius **corners** — Indicates which corners are chamfered. x<corners> is omitted if all corners are chamfered.   See “Rounded/Chamfered Rectangles” on page 185.
Oval		**oval<w>x<h>** **w** — Oval width **h** — Oval height

Table 1-1. Basic Standard Symbols

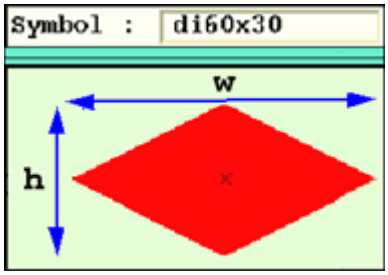
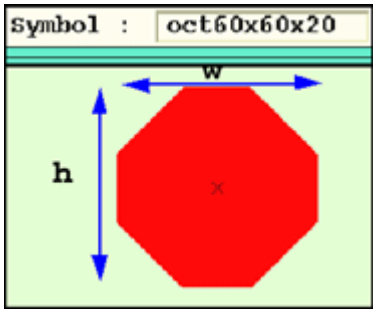
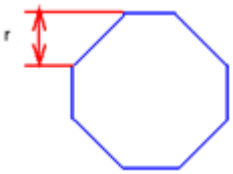
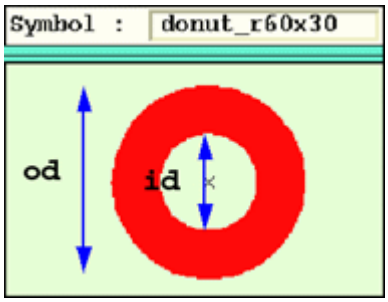
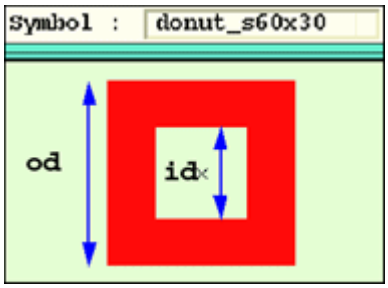
Symbol Name	Example	Parameters
Diamond		**di<w>x<h>** **w** — Diamond width **h** — Diamond height
Octagon		**oct<w>x<h>x<r>** **w** — Octagon width **h** — Octagon height **r** — Corner size 
Round Donut		**donut_r<od>x<id>** **od** — Outer diameter **id** — Inner diameter
Square Donut		**donut_s<od>x<id>** **od** — Outer diameter **id** — Inner diameter

Table 1-1. Basic Standard Symbols

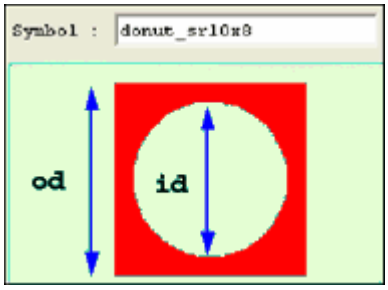
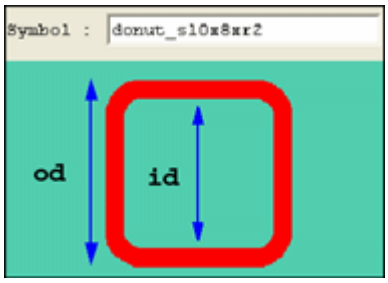
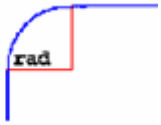
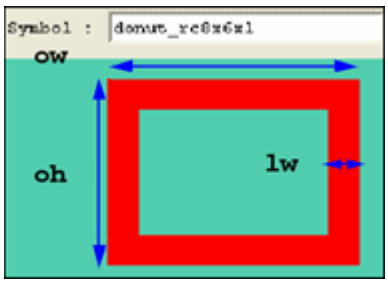
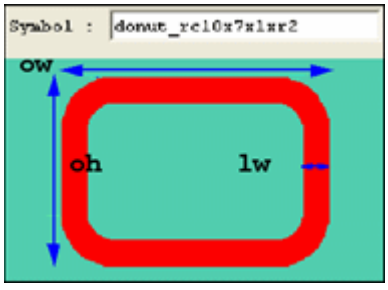
Symbol Name	Example	Parameters
Square / Round Donut		**donut_sr<od>x<id>** **od** — Outer diameter **id** — Inner diameter
Rounded Square Donut		**donut_s<od>x<id>xr<rad>** **x<corners>** **od** — Outer diameter **id** — Inner diameter **rad** — Corner radius **corners** — Indicates which corners are rounded. x<corners> is omitted if all corners are rounded.  
Rectangle Donut		**donut_rc<ow>x<oh>x<lw>** **ow** — Outer width **oh** — Outer height **lw** — Line width
Rounded Rectangle Donut		**donut_rc<ow>x<oh>x<lw>** **xr<rad>x<corners>** **ow** — Outer width **oh** — Outer height **lw** — Line width **rad** — Corner radius **corners** —Indicates which corners are rounded. x<corners> is omitted if all corners are rounded.  

Table 1-1. Basic Standard Symbols

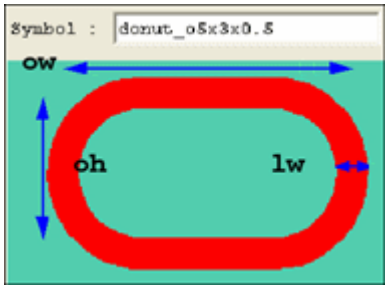
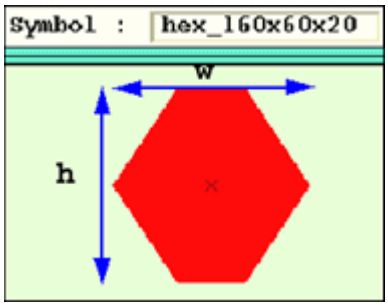
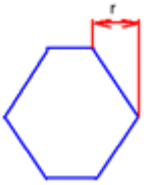
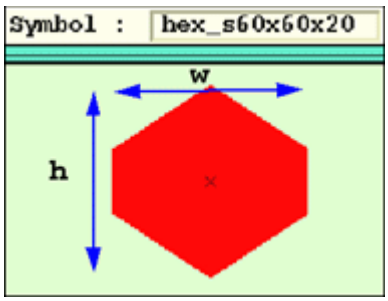
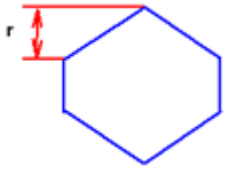
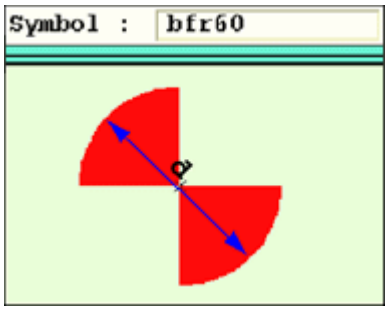
Symbol Name	Example	Parameters
Oval Donut		**donut_o<ow>x<oh>x<lw>** **ow** — Outer width **oh** — Outer height **lw** — Line width
Horizontal Hexagon		**hex_l<w>x<h>x<r>** **w** — Hexagon width **h** — Hexagon height **r** — Corner size 
Vertical Hexagon		**hex_s<w>x<h>x<r>** **w** — Hexagon width **h** — Hexagon height **r** — Corner size 
Butterfly		**bfr<d>** **d** — Diameter

Table 1-1. Basic Standard Symbols

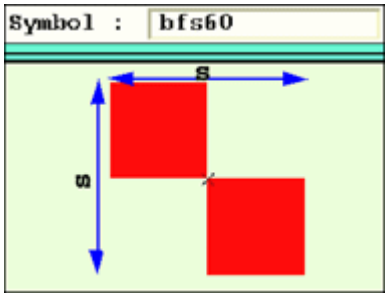
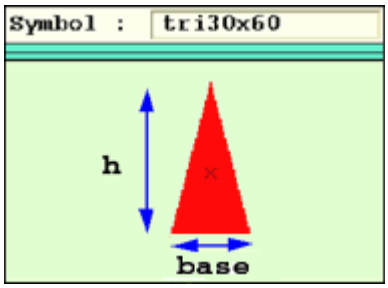
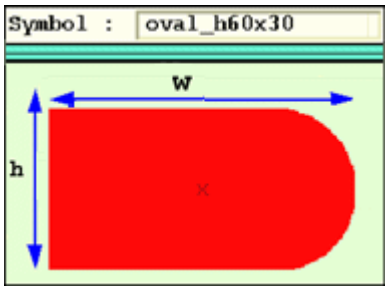
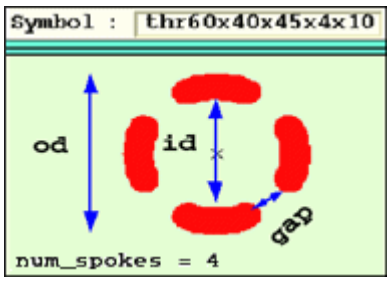
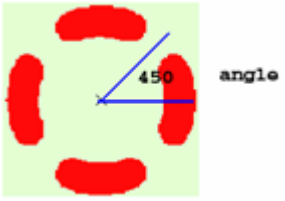
Symbol Name	Example	Parameters
Square Butterfly		**bfs<s>** **s** — Size
Triangle		**tri<base>x<h>** **base** — Triangle base **h** — Triangle height
Half Oval		**oval_h<w>x<h>** **w** — Width **h** — Height
Round Thermal (Rounded)		**thr<od>x<id>x<angle>** **x<num_spokes>x<gap>** **od** — Outer diameter **id** — Inner diameter **angle** — Gap angle from 0 degrees  **num_spokes** — Number of spokes **gap** — Size of spoke gap od and id control the air gap (size of laminate separation).

Table 1-1. Basic Standard Symbols

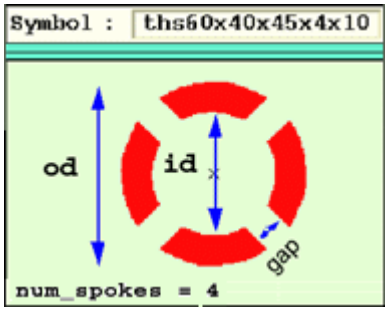
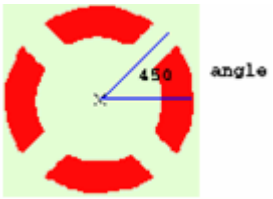
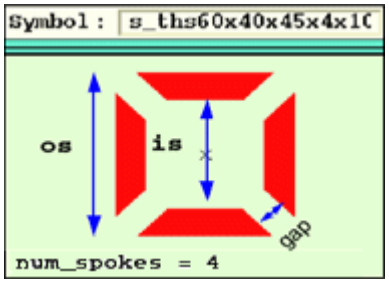
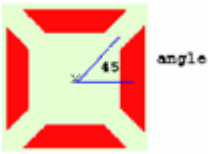
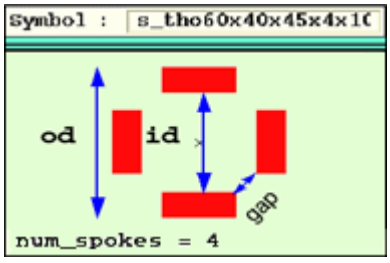
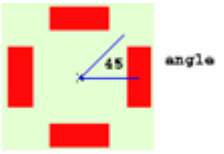
Symbol Name	Example	Parameters
Round Thermal (Squared)		**ths<od>x<id>x<angle> x<num_spokes>x<gap>** **od** — Outer diameter **id** — Inner diameter **angle** — Gap angle from 0 degrees  **num_spokes** — Number of spokes **gap** — Size of spoke gap od and id control the air gap (size of laminate separation).
Square Thermal		**s_ths<os>x<is>x<angle> x<num_spokes>x<gap>** **os** — Outer size **is** — Inner size **angle** — Gap angle from 0 degrees  **num_spokes** — Number of spokes **gap** — Size of spoke gap os and is control the air gap (size of laminate separation).
Square Thermal (Open Corners)		**s_tho<od>x<id>x<angle> x<num_spokes>x<gap>** **od** — Outer diameter **id** — Inner diameter **angle** — Gap angle from 0 degrees in increments of 45 degrees  **num_spokes** — Number of spokes: 1, 2, or 4 **gap** — Size of spoke gap od and id control the air gap (size of laminate separation).

Table 1-1. Basic Standard Symbols

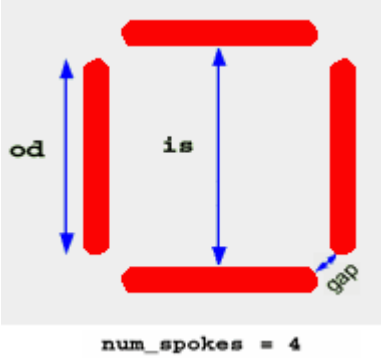
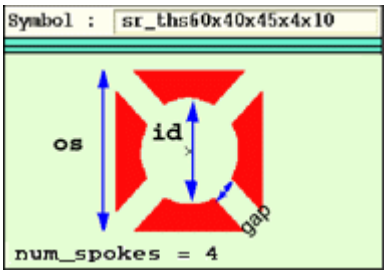
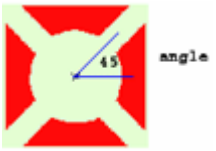
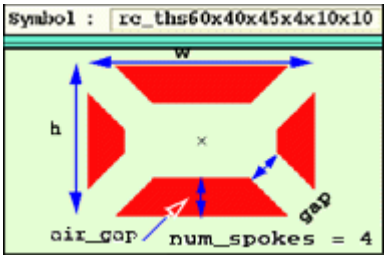
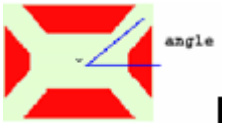
Symbol Name	Example	Parameters
Line Thermal		**s_thr<os>x<is>x<angle> x<num_spokes>x<gap>** **os** — Outer size **is** — Inner size **angle** — Gap angle always 45 degrees **num_spoke** — number of spokes always 4 **gap** — Size of spoke gap Ends of lines are rounded with diameter (os-is)/2.
Square-Round Thermal		**sr_ths<os>x<id>x<angle> x<num_spokes>x<gap>** **os** — Outer size **id** — Inner diameter **angle** — Gap angle from 0 degrees  **num_spokes** — Number of spokes **gap** — Size of spoke gap **os** and id control the air gap (size of laminate separation).
Rectangular Thermal		**rc_ths<w>x<h>x<angle> x<num_spokes>x<gap> x<air_gap>** **w** — Outer width **h** — Outer height **angle** — Gap angle from 0 degrees; in multiples of 45 degrees  **num_spokes** — Number of spokes **gap** — Size of spoke gap **air_gap** — Size of laminate separation

Table 1-1. Basic Standard Symbols

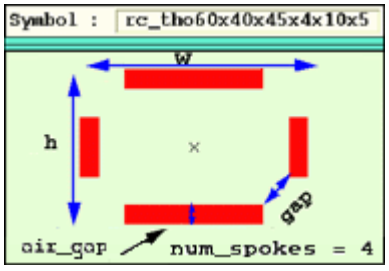
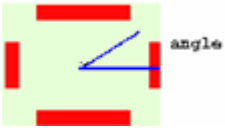
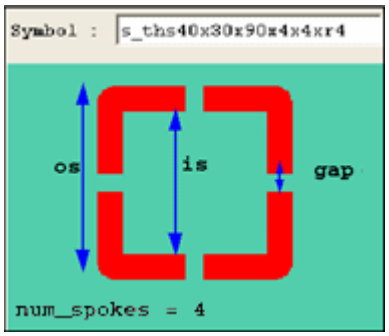
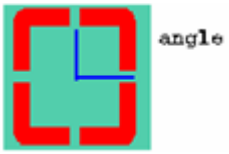
Symbol Name	Example	Parameters
Rectangular Thermal (Open Corners)		**rc_tho<w>x<h>x<angle> x<num_spokes>x<gap> x<air_gap>** **w** — Outer width **h** — Outer height **angle** — Gap angle from 0 degrees in multiples of 45 degrees.  **num_spokes** — Number of spokes **gap** — Size of spoke gap **air gap** — Size of laminate separation
Rounded Square Thermal		**s_ths<os>x<is>x<angle> x<num_spokes>x<gap>xr<rad> x<corners>** **os** — Outer size **is** — Inner size **angle** — Gap angle from 0 degrees  **num_spokes** — Number of spokes **gap** — Size of spoke gap **rad** — Corner radius **corners** — Indicates which corners are rounded. x<corners> is omitted if all corners are rounded. 

Table 1-1. Basic Standard Symbols

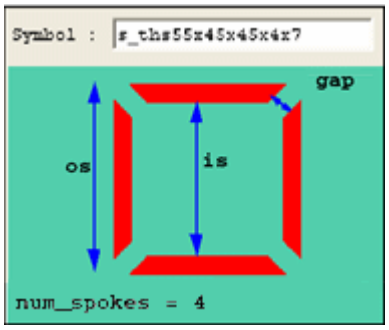
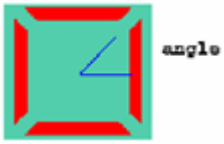
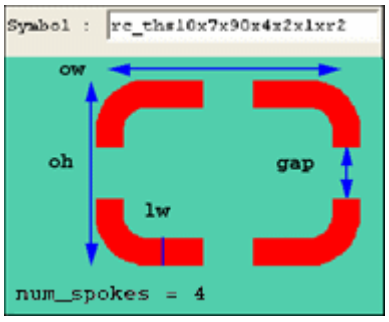
Symbol Name	Example	Parameters
Rounded Square Thermal (Open Corners)		**s_ths<os>x<is>x<angle>x<num_spokes>x<gap>xr<rad>x<corners>** **os** — Outer size **is** — Inner size **angle** — Gap angle from 45 degrees  **num_spokes** — Number of spokes **gap** — Size of spoke gap **rad** — Corner radius **corners** — Indicates which corners are rounded. x<corners> is omitted if all corners are rounded. 
Rounded Rectangle Thermal		**rc_ths<ow>x<oh>x<angle>x<num_spokes>x<gap>x<lw>xr<rad>x<corners>** **ow** — Outer width **oh** — Outer height **lw** — Line width **angle** — Gap angle from 0 degrees  **num_spokes** — Number of spokes **gap** — Size of spoke gap **rad** — Corner radius **corners** — Indicates which corners are rounded. x<corners> is omitted if all corners are rounded. 

Table 1-1. Basic Standard Symbols

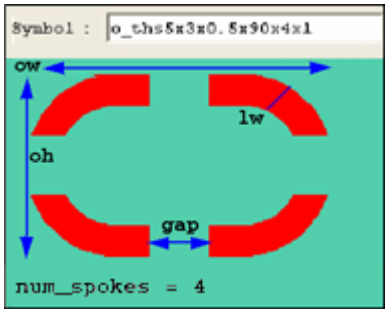
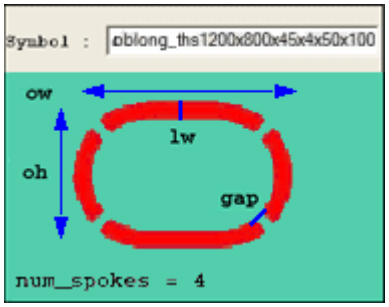
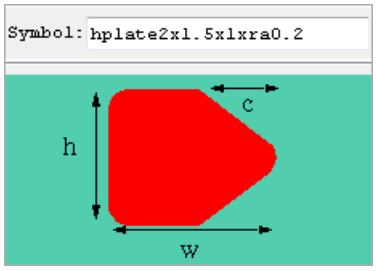
Symbol Name	Example	Parameters
Oval Thermal		**o_ths<ow>x<oh>x<angle>x<num_spokes>x<gap>x<lw>** **ow** — Outer width **oh** — Outer height **angle** — Gap angle from 0 degrees  **num_spokes** — Number of spokes **gap** — Size of spoke gap **lw** — Line width
Oblong Thermal		**oblong_ths<ow>x<oh>x<angle>x<num_spokes>x<gap>x<lw>x<r|s>** **ow** — Outer width **oh** — Outer height **angle** — For 2 spokes, angle can be 0 or 90 degrees; for 4 spokes, angle can be 0, 45, or 90 degrees. **num_spokes** — Number of spokes **gap** — Size of spoke gap **lw** — Line width **r|s** — Support for rounded or straight corners
Home Plate (hplate)		**hplate<w>x<h>x<c>** **hplate<w>x<h>x<c>x<ra>x<ro>** **w** — Horizontal side **h** — Vertical side **c** — Cut size (c<=w) **ra** — Corner radius (acute angle) (optional) **ro** — Corner radius (obtuse angle) (optional) ra and ro must meet the restrictions in “Symbols Suitable for Solder Stencil Design” on page 184.

Table 1-1. Basic Standard Symbols

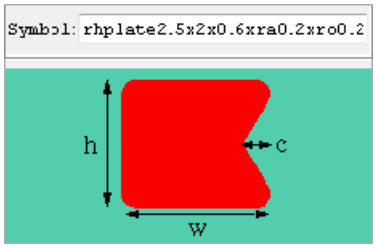
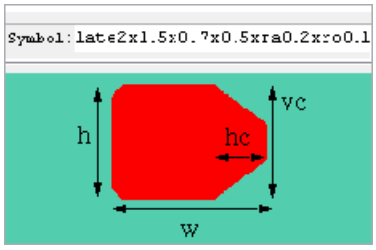
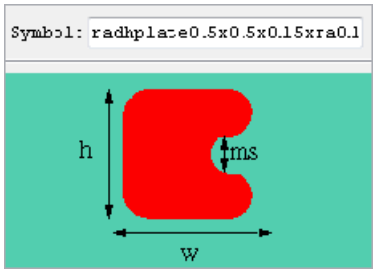
Symbol Name	Example	Parameters
Inverted Home Plate (rhplate)		**rhplate<w>x<h>x<c>** **rhplate<w>x<h>x<c>x<ra>x<ro>** **w** — Horizontal side **h** — Vertical side **c** — Cut size (c<w) **ra** — Corner radius (acute angle) (optional) **ro** — Corner radius (obtuse angle) (optional) ra and ro must meet the restrictions in "Symbols Suitable for Solder Stencil Design" on page 184.
Flat Home Plate (fhplate)		**fhplate<w>x<h>x<vc>x<hc>** **fhplate<w>x<h>x<vc>x<hc>x<ra>x<ro>** **w** — Horizontal side **h** — Vertical side **hc** — Horizontal cut size (hc<w) **vc** — Vertical cut size (vc<h/2) **ra** — Corner radius (acute angle) (optional) **ro** — Corner radius (obtuse angle) (optional) ra and ro must meet the restrictions in "Symbols Suitable for Solder Stencil Design" on page 184.
Radiused Inverted Home Plate (radhplate)		**radhplate<w>x<h>x<ms>** **radhplate<w>x<h>x<ms>x<ra>** **w** — Horizontal side **h** — Vertical side (h>(ms + (w-ms)/2)/2) **ms** — Middle curve size (ms <w) **ra** — Corner radius (optional) ra and ro must meet the restrictions in "Symbols Suitable for Solder Stencil Design" on page 184.

Table 1-1. Basic Standard Symbols

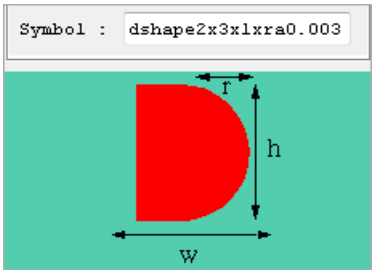
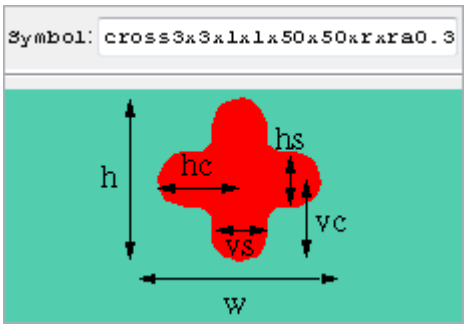
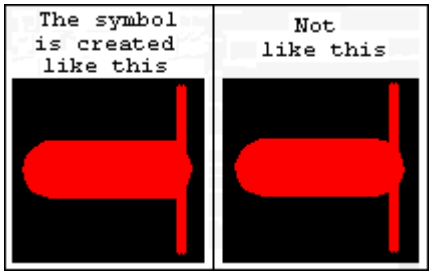
Symbol Name	Example	Parameters
Radiused Home Plate (dshape)		**dshape<w>x<h>x<r>** **dshape<w>x<h>x<r>x<ra>** **w** — Horizontal side **h** — Vertical side **r** — Relief (r<=w or (r=w and h=2*w)) **ra** — Corner radius (optional) ra and ro must meet the restrictions in "Symbols Suitable for Solder Stencil Design" on page 184.
Cross (crossx[r|s])	 Variations:  If the cross point is located outside the skeleton of the segment, the resulting indentation is ignored. 	The cross symbol consists of two intersecting orthogonal line segments. The cross point is at the intersection of their respective skeletons. **cross<w>x<h>x<hs>x<vs>x<hc>x<vc>x[r|s]** **cross<w>x<h>x<hs>x<vs>x<hc>x<vc>x[r|s]<ra>** **w** — Horizontal side **h** — Vertical side **hs** — Horizontal line width (hs<h) **vs** — Vertical line width (vs<w) **hc** — Horizontal cross point **vc** — Vertical cross point **r|s** — Line style (round or square) **ra** — Inside corner radius (limited by the lengths of the line segments)

Table 1-1. Basic Standard Symbols

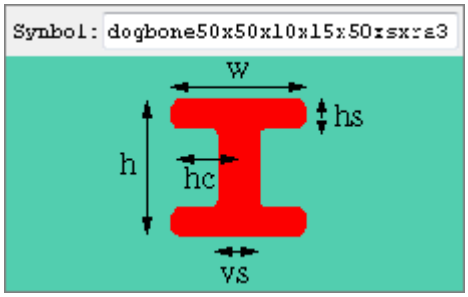
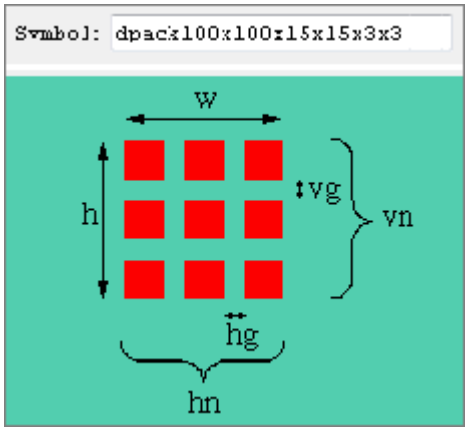
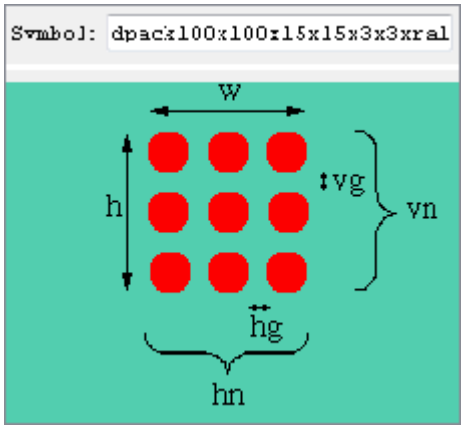
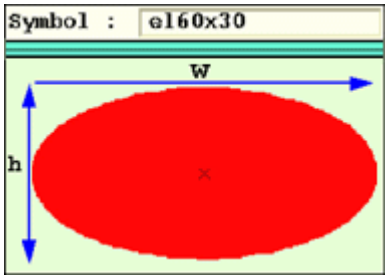
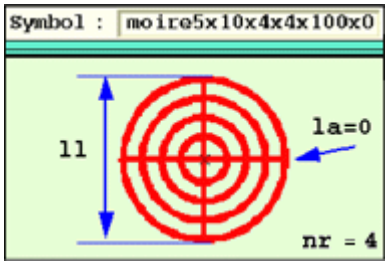
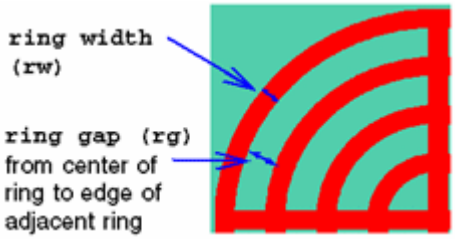
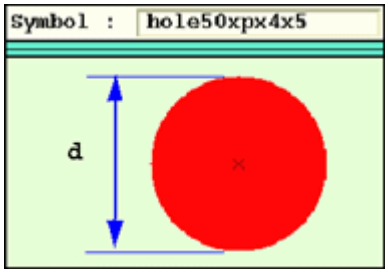
Symbol Name	Example	Parameters
Dogbone (dogbonex[r|s])	 Variation: 	**dogbone<w>x<h>x<hs>x<vs> x<hc>x[r|s]** **dogbone<w>x<h>x<hs>x<vs> x<hc>x[r|s]x<ra>** **w** — Horizontal side **h** — Vertical side **hs** — Horizontal line width (hs<h/2) **vs** — Vertical line width (vs<w) **hc** — Horizontal cross point **r|s** — Line style (round/square) **ra** — Corner radius (must be less than the whichever value is smaller: hs or vs/2)
D-Pack (dpack)	 	The individual pads of dpack can be straight or rounded, depending on the corner radius (ra) parameter. **dpack<w>x<h>x<hg>x<vg>x<hn> x<vn>** **dpack<w>x<h>x<hg>x<vg>x<hn> x<<vn>x<ra>** **w** — Horizontal side **h** — Vertical side **hg** — Horizontal gap (hg<h) **vg** — Vertical gap (vg<v) **vn** — Columns number **hn** — Rows number **ra** — Corner radius (ra<=[the smaller of hs and vs]/2) The horizontal and vertical sides of an individual pad within the D-Pack are calculated. They are not user-defined. They must be greater than zero. • hs=w-hg·(hn-1)/hn • vs=h-(vg·(vn-1))/vn

Table 1-1. Basic Standard Symbols

Symbol Name	Example	Parameters
Ellipse		**el<w>x<h>** **w** — Width **h** — Height
Moire		**moire<rw>x<rg>x<nr>x<lw>x<ll>x<la>** **rw** — Ring width **rg** — Ring gap  **nr** — Number of rings **lw** — Line width **ll** — Line length **la** — Line angle
Hole		**hole<d>x<p>x<tp>x<tm>** **d** — Hole diameter **p** — Plating status (p(lated), n(on-plated) or v(ia)) **tp** — Positive tolerance **tm** — Negative tolerance Intended for wheels created for drill files.
Null		**ext** — Extension number Used as a place holder for non-graphic features.

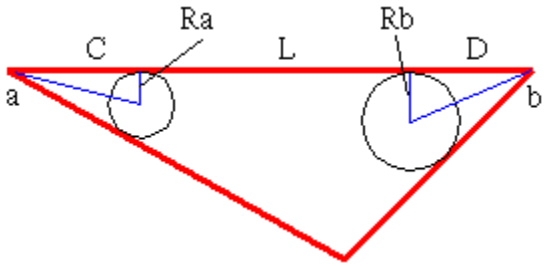
Symbols Suitable for Solder Stencil Design

Several symbols were added to fulfill the requirements for effective stencil design manufacture.

These symbols are suitable for solder stencil design:

- Home Plate (hplate)
- Inverted Home Plate (rhplate)
- Flat Home Plate (fhplate)
- Radiused Inverted Home Plate (radhplate)
- Radiused Home Plate (dshape)
- Cross (cross...[r|s])
- Dogbone (dogbone...r|s)
- D-Pack (dpack)

Most of these symbols can be created with either straight or rounded corners. If you specify a corner radius to be used for rounding the symbol, corners can be rounded only if the rounded part of one corner would not meet the rounded part of an adjacent corner.



The diagram shows how two corners of a symbol are to be rounded:

Ra — Perpendicular radius of corner a.

Rb — Perpendicular radius of corner b.

C — Distance from perpendicular radius to corner a.

D — Distance from perpendicular radius to corner b.

L — Length of the side connecting the two corners.

Aa, Ab — Angles at corners a and b.

If L > C + D, the symbol can be created with rounded corners.

This restriction can be expressed as: L > Ra/tan(Aa/2) + Rb/tan(Ab/2).

Other Symbol Information

These topics are discussed:

Rotated Standard Symbols

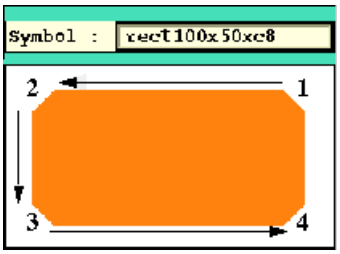
Prior to ODB++ version 7.0, pads and text could be rotated only in increments of 90 degrees. Angles other than 0, 90, 180 or 270 were considered user-defined symbols; and that increased the amount of feature data stored in the product model.

Rotation of pads and text at any angle is now allowed.

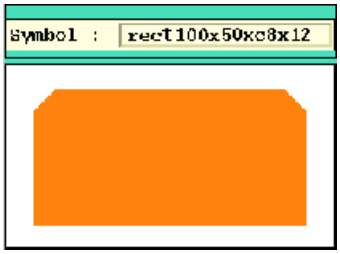
Note
Rotation is clockwise.

Rounded/Chamfered Rectangles

The corners of rounded/chamfered rectangles are specified in ascending order counter-clockwise, starting from the top-right corner. If all corners are to be rounded/chamfered, there is no need to specify corners after the **corner radius** parameter. A rectangle with all corners chamfered is indicated by: **rect100x50xc8**—100=width, 50=height, 8=corner radius.



A rectangle with only the top right corner chamfered is indicated by: **rect100x50xc8x12** (**12** indicates top right and top left corners).



These are the standard symbols whose corners can be rounded or chamfered:

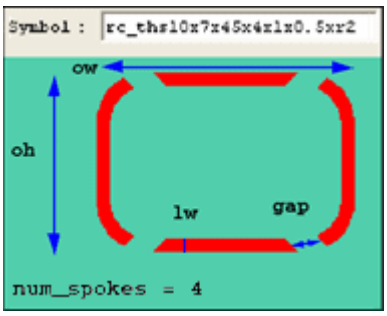
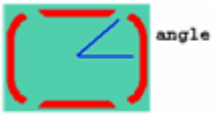
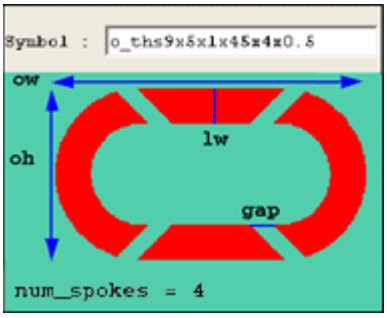
- Rounded Rectangle
- Chamfered Rectangle
- Rounded Square Donut
- Rounded Rectangle Donut
- Rounded Square Thermal
- Rounded Square Thermal (Open Corners)
- Rounded Rectangle Thermal

Corner Size

The standard Octagon symbol corner size is the distance between the bounding box corner and the vertex.

Obsolete Symbols

These symbols are obsolete.

Symbol Name	Example	Parameters
Rounded Rectangle Thermal (Open Corners)		**rc_ths<ow>x<oh>x<angle> x<num_spokes>x<gap>x<lw>xr<rad> x<corners>** **ow** — Outer width **oh** — Outer height **lw** — Line width **angle** — Gap angle of 45 degrees  **num_spokes** — Number of spokes **gap** — Size of spoke gap **rad** — Corner radius **corners** — Indicates which corners are rounded. x<corners> is omitted if all corners are rounded. 
Oval Thermal (Open Corners)		**o_ths<ow>x<oh>x<angle> x<num_spokes>x<gap>x<lw>** **ow** — Outer width **oh** — Outer height **angle** — Gap angle of 45 degrees  **num_spokes** — Number of spokes **gap** — Size of spoke gap **lw** — Line width

Appendix 2
System Attributes

Attributes are user-defined properties that add intelligence to the ODB++ data and facilitate automation.

See “ODB++ Attributes” on page 22.

All measurable values are interpreted according to the UNITS directive in the corresponding attribute definition file. See “Units of Measurement” on page 18.

For information on the *sysattr* file to which each attribute belongs, see “ODB++ System Attributes by Class” on page 238.

For an attribute that requires a units of measurement definition, this is indicated in the description. Attributes without a units of measurement definition are expressed using the units of measurement of the product model.

ODB++ System Attributes - Alphabetical List

These are the system attributes:

Table 2-1. System Attributes Supported by ODB++

Attribute	Type	Entity	Description	Display Name
.all_eda_layers	Text	Step	(0 to 1000 characters) This attribute consists of a list of all the layers in the current EDA design (not ODB++).	All EDA Layers
.aoi_cpbm	Integer	Feature	(0 to 255) (Obsolete)	.aoi_cpbm
.aoi_cpcu	Integer	Feature	(0 to 255) (Obsolete)	.aoi_cpcu
.aoi_drbm	Integer	Feature	(0 to 255) (Obsolete)	.aoi_drbm
.aoi_drcu	Integer	Feature	(0 to 255) (Obsolete)	.aoi_drcu

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.aoi_value	Integer	Feature	(0 to 255) (Obsolete)	.aoi_value
.ar_pad_drill_bottom_max	Float	Feature	(-10 to 1000) Assigned to a drill to define the maximum annular ring size between the drill and the copper of the bottom layer of the drill span. (UNITS=MIL_MICRON)	Max Pad AR Drill Bottom
.ar_pad_drill_bottom_min	Float	Feature	(-10 to 1000) Assigned to a drill to define the minimum annular ring size between the drill and the copper of the bottom layer of the drill span. (UNITS=MIL_MICRON)	Min Pad AR Drill Bottom
.ar_pad_drill_inner_max	Float	Feature	(-10 to 1000) Assigned to a drill to define the maximum annular ring size between the drill and the copper of an inner layer in the drill span. (UNITS=MIL_MICRON)	Max Pad AR Drill Inner
.ar_pad_drill_inner_min	Float	Feature	(-10 to 1000) Assigned to a drill to define the minimum annular ring size between the drill and the copper of an inner layer in the drill span. (UNITS=MIL_MICRON)	Min Pad AR Drill Inner
.ar_pad_drill_top_max	Float	Feature	(-10 to 1000) Assigned to a drill to define the maximum annular ring size between the drill and the copper of the top layer of the drill span. (UNITS=MIL_MICRON)	Max Pad AR Drill Top

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.ar_pad_drill_top_min	Float	Feature	(-10 to 1000) Assigned to a drill to define the minimum annular ring size between the drill and the copper of the top layer of the drill span. (UNITS=MIL_MICRON)	Min Pad AR Drill Top
.ar_sm_drill_bottom_max	Float	Feature	(-10 to 1000) Assigned to a drill piercing the bottom layer, to define the maximum annular ring size between the drill and the soldermask on the bottom layer. (UNITS=MIL_MICRON)	Max SM AR Drill Bottom
.ar_sm_drill_bottom_min	Float	Feature	(-10 to 1000) Assigned to a drill piercing the bottom layer, to define the minimum annular ring size between the drill and the soldermask on the bottom layer. (UNITS=MIL_MICRON)	Min SM AR Drill Bottom
.ar_sm_drill_top_max	Float	Feature	(-10 to 1000) Assigned to a drill piercing the top layer, to define the maximum annular ring size between the drill and the soldermask on the top layer. (UNITS=MIL_MICRON)	Max SM AR Drill Top
.ar_sm_drill_top_min	Float	Feature	(-10 to 1000) Assigned to a drill piercing the top layer, to define the minimum annular ring size between the drill and the soldermask on the top layer. (UNITS=MIL_MICRON)	Min SM AR Drill Top

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.ar_sm_pad_ bottom_max	Float	Feature	(-10 to 1000) Assigned to a drill piercing the bottom layer, to define the maximum annular ring size between the drilled pad of the bottom layer and the soldermask above. (UNITS=MIL_MICRON)	Max SM AR Bottom
.ar_sm_pad_ bottom_min	Float	Feature	(-10 to 1000) Assigned to a drill piercing the bottom layer, to define the minimum annular ring size between the drilled pad of the bottom layer and the soldermask above. (UNITS=MIL_MICRON)	Min SM AR Bottom
.ar_sm_pad_top_ max	Float	Feature	(-10 to 1000) Assigned to a drill piercing the top layer, to define the maximum annular ring size in between the drilled pad of the top layer and the soldermask above. (UNITS=MIL_MICRON)	Max SM AR Top
.ar_sm_pad_top_ min	Float	Feature	(-10 to 1000) Assigned to a drill piercing the top layer, to define the minimum annular ring size in between the drilled pad of the top layer and the solder mask above. (UNITS=MIL_ MICRON)	Min SM AR Top
.area_name	Text	Feature	(0 to 64) Assigned to surface features that are drawn in a process map layer. A process map layer is used in assembly analysis for determining the process type used in the location at which a measurement is found.	DFx Area Name

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.array_with_rotation	Boolean	Step	(YES, NO) (default=NO) If YES, this step is a multi-panel array, with the same panel possibly appearing in 180-degree rotation to itself	Array with Rotation
.artwork	Text	Feature	(0-1000) Indicating to which entity the feature belongs (component, package, net, board)	Feature Source
.assembly_proc_ bottom	Text	Step	(0 to 20) Default assembly process for the bottom side, to be used when there is no specific area defined in the process map layer (or no process map layer at all)	Assembly Process (Bottom)
.assembly_proc_top	Text	Step	(0 to 20) Default assembly process for the top side, to be used when there is no specific area defined in the process map layer (or no process map layer at all).	Assembly Process (Top)
.axi_direction	Option	Step	(Left2Right; Top2Bottom; Right2Left; Bottom2Top) Defines the angle at which a board is inserted into a 5DX machine. Values are translated as 0, 90, 180, 270 degrees.	Assembly X-Ray Inspection Direction
.bit	Text	Feature	(0-64) Contains the drill designator that is to be used for each tool.	Drill Designator

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.board_mark	Option	Feature	(bbm, gpm) In Vi-Technology output, this controls whether a step or a panel needs to be inspected: **bbm**—this feature is a bad board mark. Skip inspection of the step. **gpm**—this feature is a good panel mark. The panel can be accepted for printing without scanning its steps for bad board marks.	VI Tech. Board Mark
.board_thickness	Float	Product Model	(0.0 to 10.0; default=0.0) Total thickness of the board. (UNITS=MIL_MICRON)	Board Thickness
.break_away	Boolean	Symbol	(Default=NO) Assigned to a symbol that represents a break-away that can be inserted into any line or arc of the rout path. When adding a **break_away** symbol through dimensions, it automatically adjusts to the line or arc angle, breaks that feature (in the breaking points defined in that symbol with the **.brk_pnt** attribute), and adds all the necessary connections and dimensions.	Break Away

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.brk_point	Boolean	Feature	(YES, NO) Assigned to a pad or a dpoint in a break-away symbol (that was given the attribute **.break_away**). When adding the break-away to the line/arc in the layer, through dimensions, the line/arc is broken at the connection point with the dpoint that has the **.brk_ point** attribute. In each break-away symbol there should be two points with this attribute.	Rout Break Point
.bulk_resistivity	Float	Layer	(0-10000; Default-0) The nano-ohmic resistance of the semiconductor material. (nΩ•m) (UNITS=NANO_OHM)	Bulk Resistivity (nano-ohm-m)
.cad_local_ footprint_change	Boolean	Comp.; Pkg.	(Default=NO) Indicates whether there has been a local change to a pad code in the local design.	CADStar Pad Change
.cad_package_name	Text	Comp.	(0-10000) Contains the full CAD package name of a Cadstar component. This name can be longer than the Valor package name which is limited to 64 characters.	CADStar Package Name
.cad_part_override	Text	Comp.; Pkg.	(0-64) Assigns component properties in accordance with data received from the **ASSY_PN_OVERRIDE** property.	CAD Part Varient Support
.center_fiducial	Boolean	Comp.; Pkg.	(Default=NO) Specifies component is expected to have a fiducial at its center.	Center Fiducial Required

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.centroid_ correction_x	float	Comp; Pkg	(-100 to 100) Specifies the x-offset from the CAD centroid to the calculated package centroid.	Centroid Correction X
.centroid_ correction_y	float	Comp; Pkg	(-100 to 100) Specifies the y-offset from the CAD centroid to the calculated package centroid.	Centroid Correction Y
.color	Text	Feature; Comp.	(0 to 1000 for each color) Can be attached to any feature or component to define the color to be used in plotting a layer in HPGL-1 or 2. The format is **rrggbb** (where **r**=red, **g**=green, **b**=blue). White**—.color="999999"** Black**—.color="000000"** Red**—.color="990000"** Green**—.color="009900"** Yellow**—.color="009999"** Blue**—.color="000099"** Magenta**—.color="990099"** Cyan**—.color="999900"**	HPGL Output Color
.comment	Text	Product Model; Step; Layer; Wheel; Symbol; Buildup	(0 to 500) Used for general textual comments.	Comment
.comp	Option	Feature	(none; right; left) For a chained feature, this attribute sets the offset of the cutting tool from the rout path. Three options: • None—in center of the rout path • Left—to the left of the rout path in the direction of cutting • Right—to the right of the path	Rout Compensate

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.comp_height	Float	Comp.; Pkg.	(0.0 to 10.0) Stores the height of the component above the board surface. (UNITS=INCH_ MM)	Height
.comp_height_area	Integer	Feature; Comp.	(0.0 to 1000000.0; default=0) This attribute is used to assign the same ID to a component under which there is an area with space for a shorter component, and to the feature that defines the area.	Allegro Height ID
.comp_height_max	Float	Comp.; Pkg.	(0.0-10.0) The height of the tallest package (in the CPN package of alternate MPNs) above the board surface. (UNITS=INCH_MM)	Maximal Height
.comp_htol_minus	Float	Comp.; Pkg.	(0.0-10.0) Contains the minus tolerance for component height, used for calculation of plug-in boards. (UNITS=INCH_MM)	Height Tolerance Minus
.comp_htol_plus	Float	Comp.; Pkg.	(0.0-10.0) Contains the plus tolerance for component height, used for calculation of plug-in boards. (UNITS=INCH_ MM)	Height Tolerance Plus
.comp_ign_spacing	Boolean	Comp.; Pkg.	(Default=NO) This attribute, when set, disables spacing checks on a component during assembly analysis. It is used for printed components that have no actual body	Ignore during Spacing Analysis

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.comp_ignore	Boolean	Comp.; Pkg.	(Default=NO) Determines whether the component is to be ignored when calculating statistics, or during certain operations, such as analysis.	Ignore
.comp_mount_type	Option	Comp.; Pkg.	(Other; SMT; THMT; PressFit) Indicates whether the component is a surface mount, through-hole mount, press-fit or other. (SMT;THMT;PRESSFIT)	Mount Type
.comp_polarity	Option	Comp.; Pkg.	• POLARIZED, has a specific pin designated as pin #1. • NON_POLARIZED has no specific pin #1.	Polarity
.comp_type	Option	Comp.; Pkg.	This attribute represents the type of the component. Options are: axial; bga; cbga; cob; dip; discrete; discrete402; discrete603; label; pga; pihconn; pihmisc; plcc; pqfp; printed; qfp; radial; sip; smtconn; smtmisc; socket; soic; soj; sop; sot; tab; tqfp; tsoic; tsop	Type I

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.comp_type2	Option	Comp.; Pkg.	Options are: axial; axial-large; bga; cbga; cob; csp; dip; dip300; dip600; discrete; discrete201; discrete402; discrete603; electro-mech; flipchip; label; lcc; lqfp; pfconn; pga; pihconn-inline; pihconn-rt-angle; pihmisc; pih-polar; plcc; pqfp; printed; qfp; radial; radial-tall; sip; smtconn; smtelect-mech; smtmisc; smtmixedconn; smtpolar; socket; soic; soj; solderable-mech; sop-ssop; sot; tab; tqfp; tsoic; tsop; tsop-tssop	Type II
.comp_variant_list	Text	Comp.; Pkg.	(0-1000) Consists of a list of variants where a component is used. The list contains variant names separated by a colon (:). See “Component Variant Implementation” on page 35.	Variant List
.comp_weight	Float	Comp.; Pkg.	(0.0-1000.0) Stores the weight of the component (in accordance to its Units of Measurement) for the purpose of the total weight calculation.	Weight
.copper_weight	Float	Layer	(0.0 to 1000.0; default=1.0) The weight of copper according to its Units of Measurement.	Copper Weight
.critical_net	Boolean	Feature; Net	(YES, NO) Specifies critical nets.	SQA Critical Net

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.critical_tp	Boolean	Feature	(YES, NO) Assigned to the mid-point of a netlist to force it to become a testpoint. If both **.non_tp** and **.critical_tp** are assigned to the same point, **.critical_tp** takes precedence and the mid point is tested. In case of a drilled feature the attribute must be added to the drill hole.	Netlist Critical Midpoint Output
.cu_base	Boolean	Layer	(Default=NO) This attribute indicates that the specific via layer is built in such a way that it necessitates a copper pad on each layer of the buildup, since the vias are drilled and filled (rather than plated), and the pads are an essential element in ensuring connectivity.	Vias Need Cu
.current_variant	Text	Step	(0-100) Consists of the name of the current variant for a step. See “Component Variant Implementation” on page 35.	Current Variant
.customer	Text	Product Model	(0-100) This attribute can store the name of the customer for whom the product model was created.	Customer

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.cut_line	Integer	Feature	(0-100000; default=0) Assigned to lines added in the creation of film layers by the film optimization algorithm. The attribute is given to three kinds of lines: • frame of the film • cutting lines inside the film • frame of each layer inside the film.	Film Optimization Cut Line
.data_source	Text	Product Model; Step	(0-100) The source of the data. For example, Cadence, Mentor.	Data Source
.desc1....10	Text	Comp.; Pkg.	(0-1000) Storage for ten BOM description fields into corresponding ten description attributes.	General Description 1....10
.design_center	Text	Product Model; Step	(0-100) The design center from which the product model originated.	Design Center
.design_origin_x	Integer	Product Model	(-254000000 to 254000000) Defines the design origin x coordinate.	Design X Origin
.design_origin_y	Integer	Product Model	(-254000000 to 254000000) Defines the design origin y coordinate.	Design Y Origin
.device_style	Text	Comp.; Pkg.	(0-64) This attribute is an enhancement of .device_ type and is used to store the style of the component as defined in GenCAD (such as, NPN, PNP, NFET, PFET, NJFET, PJFET, TTL, CMOS and ECL).	Device Style

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.device_type	Text	Comp.; Pkg.	(0-64) Stores the type of the component as defined in the device (such as, RES, VRES, DIODE, ZENER, LOGIC, SWITCH, CONN, etc.).	Device Type
.device_value	Text	Comp.; Pkg.	(0-64) Stores the electrical value of a component.	Device Electrical Value
.device_value_ntol	Float	Comp.; Pkg.	0-1000000; default=0.0) This is a real value expressing the percent of the value to use as a tolerance (negative tolerance). This is used for all devices. Range of characters: all floating point numbers.	Device Value Negative Tolerance
.device_value_ptol	Float	Comp.; Pkg.	(0-1000000; default=0.0) This is a real value expressing the percent of the value to use as a tolerance (positive tolerance). This is used for all devices.	Device Value Positive Tolerance
.dielectric_constant	Float	Layer	(0.0 to 1000.0; default=0.0) The ratio of the field without dielectric (Eo) to the net field (E) with dielectric. It is unitless and has a range value of 1 for metals. Typical values are 4 and 5.	Dielectric Constant (ER)
.diff_pair	Text	Net	(0 to 64) Differential pair name associating two nets that must be routed together.	Differential Pair
.dpair_gap	Float	Net	(0.0 to 10.0) Spacing gap value specifying the spacing between differential pair nets. (UNITS=INCH_MM)	Differential Pair Spacing
.drc_add_rad	Integer	Mania_ AOI	(0 to 100; default=2) (Obsolete)	Enlarge By

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.drc_assembly_lyrs	Option	Feature	(Top; Bottom; Both) Specifies whether the keepout/keepin area applies to Top, Bottom, or Both component layers.	Assigned Area to Component Side
.drc_bend_keepout	Boolean	Feature	NOT USED	FLEX Bend Keepout Area
.drc_board	Boolean	Feature	(YES, NO) Assigned to a DRC area defined for the whole board.	Analysis Board Area
.drc_comp_height	Boolean	Feature	(YES, NO) Assigns component height restriction to a keepin/keepout area.	Component Height for Area
.drc_comp_height_lyr	Text	Product Model	(0 to 64) Stores name of document layer in which all component height restriction keepin/keepout areas are stored.	Comp. Height Restriction Layer
.drc_comp_keepin	Boolean	Feature	(YES, NO) Defines an area as the board's component placement keepin boundary.	Component Keep In
.drc_comp_keepin_lyr	Text	Product Model	(0 to 64) Stores name of document layer in which all component keepin areas are stored.	Component Keep In Layer
.drc_comp_keepout	Boolean	Feature	(YES, NO) Defines an area as the board's component placement keepout boundary.	Component Keep Out
.drc_comp_keepout_lyr	Text	Product Model	(0 to 64) Stores name of document layer in which all component keepout areas are stored.	Component Keep Out Layer

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.drc_etch_lyrs	Text	Feature	(0 to 1000) Value=layer names separated by semi-colons (;). User-defined attribute for user to specify name of layers in which to activate keepin/keepout areas. For example, when **.drc_etch_lyrs=pg1;pg2**, this enables you to select/highlight (on the relevant document layer) keepin/keepout areas that are active in layers **pg1**, **pg2**. The attributes: **.drc_etch_lyrs_bit** and **.drc_etch_lyrs** must both specify the same layers. If there is a discrepancy between the two, then **.drc_etch_lyrs_bit** is the determining attribute.	DFx Area Layers by Name
.drc_etch_lyrs_all	Boolean	Feature	(YES, NO) Defines a keepin/keepout area as effective on all layers.	DFx Area All Layers
.drc_etch_lyrs_bit	Text	Feature	(0 to 64) Values=string consisting of ‘0’ and ‘1’ characters. Allows the keepin/keepout area to apply only to specified board layers. The attribute’s length is equal to the number of board layers. 0=ignore layer, 1=activate areas in that layer	DFx Area Selected Layers
.drc_max_height	Float	Feature	(0.0 to 10.0) Stores the maximum height of components to be allowed in a height restriction area (area with .drc_comp_height attribute). (UNITS=INCH_MM)	Maximum Height for Component

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.drc_mech	Boolean	Feature	(Obsolete)	Analysis Mechanical Area
.drc_min_height	Float	Feature	(0.0 to 10.0) Stores the minimum height of components to be allowed in a height restriction area (area with .drc_comp_height attribute). (UNITS=INCH_MM)	Minimum Height for Component
.drc_min_space	Integer	Mania_AOI	(1 to 100; default=5) (Obsolete)	Min Spacing
.drc_min_width	Integer	Mania_AOI	(1 to 100; default=7) (Obsolete)	Min Track Width
.drc_pad_keepout	Boolean	Feature	(YES, NO) Specifies area to be used as pads keepout boundary.	Pad Keep Out
.drc_pad_keepout_lyr	Text	Product Model	(0 to 64) Stores name of document layer in which all pad keepout areas are stored. Default as defined in the **drc_pad_keepout** configuration parameter.	Pad Keep Out Layer
.drc_plane_keepout	Boolean	Feature	(YES, NO) Specifies area to be used as planes keepout boundary	Plane Keep Out
.drc_plane_keepout_lyr	Text	Product Model	(0 to 64) Stores name of document layer in which all plane keepout areas are stored.	Plane Keep Out Layer
.drc_ref_des	Text	Feature	(0 to 100) Assigned to DRC areas defined for components.	Reference Designator
.drc_route_keepin	Boolean	Feature	(YES, NO) Specifies areas to be used as the rout keepin boundary (rout=lines, arcs, vias, pads and surfaces on signal and/or power and ground layers).	Route Keep In

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.drc_route_keepin_lyr	Text	Product Model	(0 to 64) Stores name of document layer in which all rout keepin areas are stored.	Route Keep In Layer
.drc_route_keepout	Boolean	Feature	Specifies areas to be used as the rout keepout boundary (rout=lines, arcs, vias, pads and surfaces on signal and/or power and ground layers).	Route Keep Out
.drc_route_keepout_lyr	Text	Product Model	(0 to 64) Stores name of document layer in which all rout keepout areas are stored.	Route Keep Out Layer
.drc_tp_keepin	Boolean	Feature	Defines areas to be used as testpoint keepin area boundaries.	Testpoint Keep In
.drc_tp_keepin_lyr	Text	Product Model	(0 to 64) Stores name of document layer in which all testpoint keepin areas are stored.	Testpoint Keep In Layer
.drc_tp_keepout	Boolean	Feature	Specifies areas to be used as the testpoint keepout boundary.	Testpoint Keep Out
.drc_tp_keepout_lyr	Text	Product Model	(0 to 64) Stores name of document layer in which all testpoint keepout areas are stored.	Testpoint Keep Out Layer
.drc_trace_keepout	Boolean	Feature	(YES, NO) Defines areas to be used as trace keepout boundaries (traces=lines and arcs on signal and/or power and ground layers).	Trace Keep Out
.drc_trace_keepout_lyr	Text	Product Model	(0 to 64) Stores name of document layer in which all traces keepout areas are stored.	Trace Keep Out Layer

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.drc_via_keepout	Boolean	Feature	(YES, NO) Defines areas to be used as vias keepout boundaries.	Via Keep Out
.drc_via_keepout_lyr	Text	Product Model	(0 to 64) Stores name of document layer in which all vias keepout areas are stored.	Via Keep Out Layer
.drill	Option	Feature	(plated; non_plated; via) Assigned to hole features in drill layers. It defines the type of the drill and is used extensively during fabrication analysis.	Drill Type
.drill_flag	Integer	Feature	(0 to 100000; default=0) (Obsolete)	Auto Drill Mgr Flag
.drill_layer_direction	Option	Layer	(top2bottom, bottom2top, any; default=top2bottom) Specifies the drill direction for all drills on a given layer. This information impacts on analysis and potentially on the generation of production tooling information. See START_NAME and END_NAME in “Fields in the LAYER Array of a Matrix File” on page 46.	Drill Layer Direction
.drill_noopt	Boolean	Feature	(Obsolete)	Auto Drill No Opt.
.drill_sr_zero	Option	Feature	(1; 2; 3) (Obsolete)	Auto Drill S and R Offset
.drill_stage	Option	Feature	(1; 2; 3) (Obsolete)	Auto Drill Mgr Stage
.dxf_dimension	Boolean	Feature	(YES, NO) Assigned during DXF file input to mark its features as part of a DXF dimension entity.	DXF Dimension

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.eclass_ accumulative_ parallel_dist_list	Text	Net	(0-255) List of electrical class rules (blank-separated) defining the maximum distance between the two traces of nets considered parallel.	Accum. Parallel Dist. List
.eclass_ accumulative_ parallel_max_ length_list	Text	Net	(0-255) List of electrical class rules (blank-separated) defining the maximum distance between the two traces of nets considered parallel.	Accum. Parallel Max. Len. List
.eclass_impedance	Float	Net	(0.0 to 1000.0) Electrical class rule	Impedance
.eclass_individual_ parallel_dist_list	Text	Net	(0-255) List of blank-separated electrical class rules. Defines the separation distance within which two traces are considered parallel.	Individ. Parallel Dist. List
.eclass_individual_ parallel_max_ length_list	Text	Net	(0-255) List of electrical class rules (blank-separated). Defines the maximum length that two nets can run parallel to each other.	Individ. Parallel Max. Len. List
.eclass_individual_ parallel_min_jog_ list	Text	Net	(0-255) List of electrical class rules (blank-separated). Defines the distance parallel traces that deviate must maintain the deviation before it is considered a break in parallelism.	Individ. Parallel Min. Jog List
.eclass_max_stub_ length	Float	Net	(0.0 to 100.0) Electrical class rule—high limit of the stub length. (UNITS=MIL_MICRON)	Max. Stub Length
.eclass_max_via_ count	Integer	Net	(0 to 1000) Maximal number of vias on the nets.	Max. Via Count

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.eclass_min_stub_ length	Float	Net	(0.0 to 100.0) Electrical class rule—low limit of the stub length. (UNITS=MIL_MICRON)	Min. Stub Length
.eclass_rise_time	Float	Net	(0.0 to 100.0) Electrical class rule specifying the interval of a rising signal transition (low to high)	Rise Time
.eclass_voltage_ swing	Float	Net	(0.0 to 100.0) Electrical class rule	Voltage Swing
.ecmp_layer_tech	Option	Layer	(none, additive, subtractive) Assigns a technology type attribute to an embedded components layer.	Embedded Passive Technology
.ecmp_max_value	Float	Feature	(0.0 to 1000000.0) Maximum nominal value received at input (its value plus a tolerance).	Embedded Passive Max. Value
.ecmp_min_value	Float	Feature	(0.0 to 1000000.0) Minimum value received at input (its value minus a tolerance).	Embedded Passive Min. Value
.ecmp_name	Text	Feature	(0 to 64 characters) Name assigned to an embedded passive feature.	Embedded Passive Name
.ecmp_value	Float	Feature	(0.0 to 1000000.0) Embedded passive nominal value. For resistors it is the resistance in ohms.	Embedded Passive Nominal Value
.eda_dimension_id	Integer	Feature	(0 to 100000; default=0) Assigns system-generated ID to dimensions	Embedded Passive Nominal Value

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.eda_layers	Text	Layer	(0 to 1000) Contains the EDA system layer names that compose a physical layer. It is loaded during the direct EDA translation and is used for graphic synchronization with the EDA system.	EDA Layers
.electrical_class	Text	Net	(0 to 64) Electrical class name associating a net with a set of electrical call rules. Electrical class rules include physical and electrical limitations required to assure and analyze the signal integrity of a high speed net.	Electrical Class
.et_adjacency	Float	Layer	(0.0 to 1000.0; default=20.0) A distance value (per layer) to use in netlist adjacency calculation for moving probe testers (currently BSL and PROBOT). (UNITS=MIL_MICRON)	Adjacency Distance (ET)
.et_align	Boolean	Feature	(YES, NO) Determines that a feature will be used as an alignment target for PROBOT output	Probot Alignment Targets

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.extended	Integer	Feature	(-1 to 100000; default=0) Assigned to construction features (lines and pads) added to assist in the generation of a rout path. These features have zero width and are not output to the rout machine as regular features. They are used, for example, as source elements from which to create actual features by dimensions. If the attribute value is not zero then the feature is an extended feature and the decimal value is its serial value in the layer (to be referenced in dimension creation).	Dimension Feature
.fab_drc	Text	Step	(0 to 20) Stores the default DRC area name. This name is applied when no specific area is defined in the DRC map layer, or no such map layer at all.	Fab DRC
.feature_ignore	Boolean	Feature	(YES, NO) Copper features with this attribute are ignored in analysis actions. (Currently implemented for rout tests only.)	Copper Feature Ignore
.feed	Integer	Feature	(0 to 100000; default=0) For a chained feature, this attribute sets the table feed rate when routing.	Rout Feed Rate
.fiducial_name	Text	Feature	(0 to 64) (Obsolete)	Etec Fiducial Output

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.fiducial_rdlist	Text	Feature	(0-1000) This attribute is assigned **local** fiducial features. It can consist of a list of REFDES (separated by semicolons); a list of the component/s using this local fiducial.	Local Fiducial Ref.Des. List
.fill_dx	Float	Symbol	(0.000001 to 50.0; default=0.1) This attribute is used as the default horizontal distance between symbols when the symbol is used for pattern filling. (UNITS=INCH_ MM)	Fill Dx
.fill_dy	Float	Symbol	(0.000001 to 50.0; default=0.1) This attribute is used as the default vertical distance between symbols when the symbol is used for pattern filling. (UNITS=INCH_ MM)	Fill Dy
.foot_down	Boolean	Feature	(Default=NO) (Obsolete)	Auto Rout Footdown
.fs_direction_ bottom	Option	Step	(Right2Left; Left2Right; Top2Bottom; Bottom2Top). Indicates the manufacturing direction the PCB will be going through during the flow solder process.	Flow Solder Direction (Bottom)
.fs_direction_top	Option	Step	(Left2Right; Right2Left; Top2Bottom; Bottom2Top) Indicates the manufacturing direction the PCB will be going through during the flow solder process.	Flow Solder Direction (Top)
.full_plane	Boolean	Feature	NOT USED	

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.gencad_device_ntol	Float	Comp.; Pkg.	0-1000000; default=0.0) This is a real value expressing the percent of the value to use as a tolerance (negative tolerance). This is used for all devices: Range of characters: all floating point numbers.	GENCAD Negative Tolerance
.gencad_device_ptol	Float	Comp.; Pkg.	(0-1000000; default=0.0) This is a real value expressing the percent of the value to use as a tolerance (positive tolerance). This is used for all devices:	GENCAD Positive Tolerance
.gencad_device_ style	Text	Comp.; Pkg.	(0-64) This attribute is an enhancement of **.gencad_ device_type** and is used to store the style of the component as defined in GenCAD (such as, NPN, PNP, NFET, PFET, NJFET, PJFET, TTL, CMOS and ECL)	GENCAD Device Style
.gencad_device_ type	Text	Comp.; Pkg.	(0-64) Stores the type of the component as defined in GenCAD (such as, RES, VRES, DIODE, ZENER, LOGIC, SWITCH, CONN, etc.).	GENCAD Device Type
.gencad_device_ value	Text	Comp.; Pkg.	(0-64) Stores the electrical value of a component.	GENCAD Device Value
.geometry	Text	Feature	(0 to 500) Contains the name of the padstack that created this feature.	Geometry
.global_camtek_ aoiset	Text	Product Model	(0-80) (Obsolete)	Global CAMTEK aoi-set

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.gold_plating	Boolean	Feature	Used during auto-panelization to orient the gold plated area toward the extreme side of the panel.	Gold Plating
.guard_comp	Boolean	Comp.; Pkg.	(Default=NO) Assigned to a component that 'guards' other components. If TRUE, this component is considered a 'guard component' (that is, not likely to be knocked off the board accidentally. To be used in future actions.)	Component Guard
.hatch	Boolean	Feature	(YES, NO) Assigned to hatched planes (filled with lines—hatches—or cross lines—cross-hatch—instead of solid copper). The lines that make up the border and fill the surface are hatches.	Hatch
.hatch_border	Boolean	Feature	(YES, NO) The lines making up the border of a surface.	Hatch Border
.hatch_serrated_ border	Boolean	Feature	(YES, NO) Assigned to features that are added for partial hatch. The difference between regular hatch and partial hatch is that in partial hatch the cells along the border that intersect the border line are filled; the feature(s) that fill these cells are assigned this attribute.	Hatch Serrated Border
.hdi_drc	Text	Step	(0 to 20) Default area name applied to all HDI actions.	HDI Assembly Technology
.hp3070_comment	Text	Comp.; Pkg.	(0-64) (Obsolete)	HP 3070 Comment

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.hp3070_common_pin	Text	Comp.; Pkg.	(0 to16). (Obsolete)	HP 3070 Common Pin
.hp3070_contact_pin	Text	Comp.; Pkg.	(0 to16). (Obsolete)	HP 3070 Contact Pin
.hp3070_device	Text	Comp.; Pkg.	(0 to 16) (Obsolete)	HP 3070 Contact Pin
.hp3070_fail_msg	Text	Comp.; Pkg.	(0-64) (Obsolete)	HP 3070 Fail Message
.hp3070_hi_value	Float	Comp.; Pkg.	(0-100000.0; default=0.0) (Obsolete)	HP 3070 High Value
.hp3070_lo_value	Float	Comp.; Pkg.	(0-100000.0; default=0.0) (Obsolete)	HP 3070 Low Value
.hp3070_probe_access	Text	Feature; Comp.	(0-64) (Obsolete)	HP 3070 Contact Pin
.hp3070_seriesr	Float	Comp.; Pkg.	(0-100000.0; default=0.0) (Obsolete)	HP 3070 Series
.hp3070_test	Boolean	Comp.; Pkg.	(Default=NO) (Obsolete)	HP 3070 Test
.hp3070_tol_neg	Float	Comp.; Pkg.	(0-100; default=0.0) (Obsolete)	HP 3070 Negative Tolerance
.hp3070_tol_pos	Float	Comp.; Pkg.	(0-100; default=0.0) (Obsolete)	HP 3070 Positive Tolerance
.hp3070_type	Text	Comp.; Pkg.	(0-8) (Obsolete)	HP 3070 Type
.hp3070_value	Text	Comp.; Pkg.	(0-16) (Obsolete)	HP 3070 Value
.ignore_net	Boolean	Net	(YES, NO) When this attribute is assigned to a net, it is ignored during Testpoint Allocation Analysis. No potential testpoints are assigned, they are not reported in the Nets without Potential TPs category, the Testpoints Allocation Report, or in Total Number of Nets.	Ignore Net During Allocation

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.image_dx .image_dy	Float	Symbol	(-1.0 to 50.0; default=-1.0) These values are set when inputting Image files into the system. They contain the datum point of an Image user-defined symbol entity used to set the datum when performing output back into Image format. These values should not be changed by the user as this can cause data corruption. (UNITS=INCH_MM)	Image Dx Image Dy
.imp_line	Boolean	Feature	Assigned to lines that are impedance-controlled. When set, it prevents the lines from being rerouted or thinned during signal layer optimization.	Impedance Line
.ind_orient_req	Boolean	Comp.; Pkg.	(Default=NO) Indicates that the component requires silkscreen orientation indication. (To be used in future actions.)	Orientation Indication Required
.inp_file	Text	Layer	(0 to 500) Contains the name of the file (Gerber, Drill) from which the data was input into the layer.	Input File
.is_buried	Boolean	Comp.; Pkg.	(Default=NO) Assigned to buried components.	Buried Component
.is_capped	Boolean	Feature	Used on via pads on top and bottom signal layers to indicate that the via is capped on this side.	Capped Via
.is_shadowed	Boolean	Comp.; Pkg.	(Default=NO) Components with this attribute are considered vunerable to component shadowing.	Check for Shadowing

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.is_wirebonded	Boolean	Comp.; Pkg.	(Default=NO) Defines a component to be wire-bonded.	Wirebonded Component
.label_clearance	Boolean	Comp.; Pkg.	(Default=NO) Assigned to components that are not allowed to be too close to a glued label (e.g. fine pitch SOIC components).	Label Clearance
.layer_class	Text	Layer	(0 to 1000 characters) Used to differentiate between layers.	Layer Class
.layer_dielectric	Float	Layer	(0.0) Applies to layers of type DIELECTRIC or SOLDERMASK in the layer matrix that contains the thickness of material according to its specified Units of Measurement. (UNITS=INCH_MM)	Thickness of Material
.layer_hdi_type	Option	Layer	(Buildup; Core) An attribute added to a copper layer—not a dielectric layer—to distinguish buildup layers from core layers in HDI product models.	HDI Layer Type
.local_fiducial_dist	Float	Comp.; Pkg.	(0.0 to 100.0) Defines the allowed distance of fiducials from the outline of the components that require local fiducials (See the attribute .num_local_ fiducial.) If set to 0, the fiducials must be included INSIDE the outline. (UNITS=INCH_ MM)	Search Distance for Local Fiducial

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.loss_tangent	Float	Layer	(0-100; Default=0) The parameter of a dielectric material that quantifies its inherent dissipation of electromagnetic energy.	Loss Tangent
.lpol_done	Boolean	Layer	(Default=NO) Reflects the state of the product model in whether or not the polarity sort according to a format has already been performed.	Polarity Order was Done
.lpol_surf	Boolean	Feature	(Default=NO) Indicates surface modified by layer polarity reduction algorithm.	Polarity Optimization
.lyr_prf_ref	Integer	Feature	(0-100000; Default=0)	Layer Profile Reference Number
.machine_pkg	Text	Comp.; Pkg.	(0-100) Assigned to a component to indicate the name of a corresponding package in the assembly machine libraries.	Package Name for ALE
.mechanical	Boolean	Comp	(YES, NO) Components with this attribute are considered part of the **MECHNICAL** section of a GenCAD file.	GENCAD Mechanical
.merge_processes	Text	Step	(0-64) A list of the last three merge actions in the order in which they were run. The list is updated each time a merge (bom_merge, library_ merge, board_merge) is run. It is for informational purposes and does not have to be changed by the user.	Assembly Merge Processes

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.min_line_width	Float	Net.	(0.0-100.0) Assigned to nets that should have a minimum trace width, i.e. each line of the net should have a width of at least this value. If such nets have a split (the net traces split and then meet again) the sum of the split traces should be at least this value. (UNITS=MIL_MICRON)	Min. Line Width
.mount_hole	Boolean	Feature	(YES, NO) Used on drill features to indicate that they are mounting holes.	Mounting Hole
.mount_stage	Integer	Comp.; Pkg.	(0-255; default=0) User-defined integer used to assign machine number in the assembly line where component is to be placed.	Placement Machine Number
.n_electric	Boolean	Feature	(YES, NO) Assigned to a feature, defines it as non-electric (it is not considered for the current netlist for the step).	Non Electrical Feature
.needs_guarding	Boolean	Comp.; Pkg.	(YES, NO) (Default=NO) YES—this component needs to be protected by guard components (see the attribute .guard_comp) else it is likely to be knocked off the board accidentally.	Require Guards
.net_length_max	Float	Net	(0.0 to 100.0) High limit of net length. (UNITS=INCH_MM)	Net Max. Length
.net_length_min	Float	Net	(0.0 to 100.0) Lower limit of net length. (UNITS=INCH_MM)	Net Min. Length
.net_name	Text	Feature; Layer	(0 to 10000) Set by the netlist layer. Contains the net name.	Net Name

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.net_physical_type	Text	Feature	(0 to 64) Physical type of constraint area used for search in table that contains physical parameters of nets.	Net Physical Type
.net_point	Boolean	Feature	(YES, NO) When assigned to a pad in an inner layer, defines the pad as an internal test point.	Internal Net Point
.net_spacing_type	Text	Feature	(0 to 64) SQA area name of an SQA area map.	Net Spacing Type
.net_test_current	Float	Net	(0 to 100000) Nominal current associated with the corresponding net.	
.net_test_type	Option	Net	(signal; power; ground) Classification type of an electrical net: Signal, Power, or Ground.	
.net_test_voltage	Float	Net	(-100000 to 100000) Nominal voltage associated with the corresponding net.	
.net_type	Text	Net	(0 to 64) A name for the type of net. The **.net_type** attribute can reference the set of routing rules for a net.	Net Type
.neutralization_ angle	Float	Comp.; Pkg.	(0.0 to 360.0) An attribute attached to each component stating the angle of rotation counter-clockwise from ODB++ standard orientation. Consideration of this value provides a rotation neutralized component.	Rotation Neutralization

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.neutralization_info	Text	Step	(0 to 200) An attribute attached to a step indicating component rotation neutralization has occurred. This attribute contains the information <CPL|CAD>;<DataCenter>;Site. where Site is a name representing the location at which the correction was performed.	Neutralization Information
.neutralization_ reviewed	Boolean	Comp.; Pkg.	(YES, NO) Attached to each component in a package reviewed for confirming component orientation correction conforms to ODB++ standard orientation.	Rotation Reviewed
.neutralization_ss_ layers	Text	Step	(0 to 200) Attached to the step where component orientations have been validated to ODB++ standard rotations through the use of the .neutralization_angle attribute. This attribute designates which layers are to be considered the silkscreen layers.	Neutralization Silkscreen
.no_copper_shape_ under	Boolean	Comp.; Pkg.	(Default=NO) This attribute indicates that the component should not have copper pads or surfaces underneath it. See also the attribute .no_trace_under.	No Shape Under
.no_fiducial_check	Boolean	Comp.; Pkg.	(Default=NO) Components with this attribute do not require fiducials.	No Fiducial Check
.no_hole_under	Boolean	Comp.; Pkg.	(Default=NO) Indicates that no drill holes are allowed under the component.	No Hole Under

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.no_pop	Boolean	Comp.; Pkg.	(YES, NO) Designates a component as being not populated for the current version of the BOM.	Not Populated per BOM
.no_protrude_board	Boolean	Comp	(YES, NO) Indicates that toeprint length (as defined by attribute .pin_ length) should be less than the board thickness so that the pins do not protrude from the other side of the board.	Pins Cannot Protrude
.no_text_under	Boolean	Comp.; Pkg.	(Default=NO) Assigned to a component, does not allow silk screen text to be placed under the component outline. Printed components (e.g. edge connectors) may not have this attribute.	No Silkscreen Under
.no_tp_under	Boolean	Comp.; Pkg.	(Default=NO) Assigned to a component, does not allow testpoints to be placed under the component outline. Printed components (e.g. edge connectors) may not have this attribute.	No Test Point Under
.no_trace_under	Boolean	Comp.; Pkg.	(YES, NO) (Default=NO) YES—traces are NOT allowed under the component except for those that touch the component's toeprint pads and exit the component on that toeprint's side.	No Trace Under
.no_uncap_via_ under	Boolean	Comp.; Pkg.	(YES, NO) (Default=NO) YES—uncapped vias are NOT allowed under this component.	No Uncapped Vias Under

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.nomenclature	Boolean	Feature	(YES, NO) Defines a feature as a nomenclature (legend) feature.	Copper Text
.non_tp	Boolean	Feature	(YES, NO) Assigned to a feature causes it NOT to be considered as a net testpoint. It is used for connectivity calculation but is not used as a test point (bare board testing).	Non Test Point
.num_local_ fiducials	Integer	Comp.; Pkg.	(0 to 20; default=0) Defines how many local fiducials are expected to be inside or near a component.	Number Local Fiducials Required
.orbotech_plot_ stamp	Boolean	Feature	NOT USED	Orbotech Plot Stamp
.orig_surf	Integer	Feature	(0-2147483647; default=0) Identifies original surface that will be rebuilt.	Original Contour Index
.otherside_keepout	Option	Comp.; Pkg.	(full_area; pins_only; pads_ only) Defines for components whether the other side of the board may also contain components in the same area.	No Components on Other Side
.out_angle	Option	Layer	(0.0; 90.0; 180.0; 270.0; default=0.0) Contains the output angle to be used when generating manufacturing data.	Output Angle
.out_break	Boolean	Feature; Symbol	(Default=NO) Indicates that upon output generation standard symbols and user-defined symbols should be replaced with basic primitive round and square symbols representing the intended shapes.	Output Break

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.out_comp	Float	Layer	(-100.0 to 100.0; default=0.0) Layer entity attributes with default values that are used by the output translator.	Output Compensation
.out_drill_full	Boolean	Step	(Default=NO) This attribute can be used for a drill coupon that needs to be fully drilled before continuing to the next step and repeat entity.	Output Drill Full
.out_drill_optional	Boolean	Feature; Step	(Default=NO) If a drill feature is set with this attribute it will have the '**/**' command prefix in the final output file, indicating that the drill is optional. If a step entity attribute is set, then all the commands that are part of that step will have the '**/**' command prefixed. Thus, the whole step is optional.	Output Drill Optional
.out_drill_order	Integer	Step	(-10000 to 10000; default=0) The attribute controls the order in which the steps are drilled. That is, who is first, second, etc. The attribute has the following valid values: 0—no special order for that step 1—first 2—second 3—and above, order from the beginning -1—last -2—one before last -3—and on (drill order from the end)	Output Drill Order

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.out_flag	Integer	Feature	(-1 to 1000000; default=-1) (Obsolete) Used for Excellon translation.	Excellon Unknown
.out_mirror	Boolean	Layer	(Default=NO) Indicates the output could be mirrored along the X- axis when generating manufacturing data.	Output Mirror
.out_name	Text	Step	(Obsolete)	Output Name
.out_orig	Boolean	Feature	(Obsolete)	Rout Output Origin
.out_polarity	Option	Layer	(Positive; Negative) Contains the output polarity to be used when generating manufacturing data.	Output Polarity
.out_rout_optional	Boolean	Feature; Step	(Default=NO) If drill feature is set with this attribute it will have the '/' command in front of it in the final output file. This means that the drill is optional. If a step entity attribute is set then all the commands that are part of that step will have the '/' command at the beginning. Thus, the whole step is optional.	Output Rout Optional

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.out_rout_order	Integer	Step	(-10000 to 10000; default=0) The attribute controls the order in which the steps will be drilled. Thus, who is first, second, etc. The attribute has the following valid values: 0 —no special order for that step 1—first 2—second 3 and above—order from the beginning -1—last -3 and on—drill order from the end	Output Rout Order
.out_scale	Boolean	Feature; Symbol	(Default=NO) During the generation of manufacturing data, it may be required to specify whether certain features can be scaled or not. This is important in cases where special registration targets would not be scaled together with all the other features. This special output option applies only to features that have this attribute set. In the case of a user-defined symbol, setting the attribute enables you to control the scaling of all features using the symbol.	Output Scale
.out_x_scale	Float	Layer	(0.000001 to 5.0; default=1.0) Contains the output scale along the X-axis to be used when generating manufacturing data.	Output X Scale

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.out_y_scale	Float	Layer	(0.000001 to 5.0; default=1.0) Contains the output scale along the Y-axis to be used when generating manufacturing data.	Output Y Scale
.output_dcode	Integer	Feature	(0 -1000000) Assigned to features to provide action codes for an assembly machine, such as the GSI Lumonics laser cutter.	GSI Lumonics Output Dcode
.package_version	Text	Comp.; Pkg.	(0 to 50) Used for Zuken Board Designer translation.	Zuken Package Info
.pad_usage	Option	Feature	(toeprint; via; g_fiducial; l_ fiducial; tooling_hole) This attribute defines the specific usage of a pad.	Pad Usage
.part_desc1...10	Text	Comp.; Pkg.	(0 to 1000) Ten BOM description fields.	Part Description 1 ... Part Description 10
.patch	Boolean	Feature	(YES, NO) Assigned to features added to eliminate pin holes.	Copper Patch
.pattern_fill	Boolean	Feature	(YES, NO) Assigned to features that represent a pattern fill, typically used for copper balancing. Best practice in copper balancing is to use zero length lines where X/Y start is the same as X/Y end (rather than pads) to support manufacturing that requires data preparation.	Copper Balancing
.pf_optimized	Boolean	Feature	(YES, NO) This attribute describes the state of the product model. It is attached to a padstack that has had non-functional pads removed.	Optimized Pad Stack

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.physical_type	Text	Net	(0 to 64) Physical type of net.	Physical Type
.pilot_hole	Integer	Feature	(0 to 100000; default=0) Attribute assigned to pads that are pilot holes in a chain (holes that are drilled in each tool down in the chain rout path before routing the chain). The value of the attribute is the serial chain number to which the pilot hole belongs. When merging or inserting chains, the pilot holes are updated automatically. (UNITS=MIL_MICRON)	Rout Pilot Hole
.pitch	Float	Feature	NOT USED	
.plated_type	Option	Feature	(Standard, Press_fit) Defines plated hole type for drill tools.	Through Hole Type
.polarity_marker	Integer	Comp.; Pkg.	(1-10000; default=1) An attribute indicating which pin of the component is Pin 1.	Polarity Indicator Pin
.primary_side	Option	Product Model	(Top; Bottom) Indicates the primary side for this product model.	Primary Side
.rot_correction	Integer	Comp.; Pkg.	(0-359) Component machine rotation correction to apply.	Rotation Correction
.rout_chain	Integer	Feature	(0 to 100000; default=0) Contains the serial number of the chain to which the feature belongs. Features belonging to that chain are rearranged in the features file according to their order inside the chain. Additional attributes that are added to a chained feature: .feed, .speed, .rout_flag, .comp.	Rout Chain

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.rout_flag	Integer	Feature	(0 to 100000; default=0). For each chained feature this attribute represents a numeric value supplied to a chain.	Rout Flag
.shave	Boolean	Feature	(Default=NO) Assigned to negative features used to optimize layer content for manufacturing purposes.	Silkscreen Shaved
.sip	Option	Feature	(Detected / Repaired). Indicates the presence of a SIP (self-intersecting polygon).	Self Intersecting Polygon
.sliver_fill	Boolean	Feature	(YES, NO) Assigned to all features added to eliminate slivers.	Sliver Fill Patch
.smd	Boolean	Feature	(YES, NO) Assigned to outer layer pads designated as toeprints that are lands for SMD components.	SMD
.smt_direction_ bottom	Option	Step	(Left2Right; Top2Bottom; Right2Left; Bottom2Top) Defines the direction of the SMT process flow on the bottom side.	SMT Direction (Bottom)
.smt_direction_top	Option	Step	(Left2Right; Top2Bottom; Right2Left; Bottom2Top) Defines the direction of the SMT process flow on the top side.	SMT Direction (Top)
.source_llayer	Text	Feature	(0-64) This attribute identifies the **Source Logical Layer** of features (traces) appearing on signal or mixed layers.	Source Layer
.spacing_req	Float	Feature	(0.0 to 100.0) (UNITS=MILS_ MICRONS)	Spacing Required

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.speed	Integer	Feature	(0 to 100000; default=0) For a chained feature this attribute sets the spindle speed (in revolutions per minute) when routing.	Rout Spindle Speed
.spo_h_fact	Float	Feature; Comp.	(0.3 <-> 2.0; default=0.8) When .spo_h_mode = Factor, .spo_h_fact specifies the factor by which paste pad heights are sized relative to their SMD pads. For example, 0.9 means height is 90% of SMD pad.	Solderpaste Height Factor Value
.spo_h_mode	Option	Feature; Comp.	(Distance, Factor, Value) Defines how heights of paste pads are sized: by distance, factor or value.	Create Solderpaste Height Mode
.spo_h_val	Float	Feature; Comp.	(-500 to +500; default=5) When .spo_h_ mode=Distance, .spo_h_val is the reduction/expansion in mils or microns of the paste pad width relative to the SMD pad width. For example, .sp_h_val=5.0 mils shrinks paste pad by 5.0 mils (2.5 mils on each side) relative to SMD pad width. Positive number results in smaller paste pad, negative number in larger paste pad. When .spo_h_mode=Value, .spo_h_val becomes the absolute width of the paste pad (for example, 5.0 mils becomes the actual width of the paste pad). (UNITS=MIL_ MICRON)	Solderpaste Height Relative Value

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.spo_move_center	Float	Feature; Comp.	(range: -500 to 500, default=0) To move the paste pad from the SMD pad center. A positive value will move the paste from the component center out. A negative value will move the paste towards the component center. (UNITS=MIL_MICRON)	Shift Solderpaste Symbol
.spo_p_fact	Float	Feature; Comp.	(0.3 <-> 2.0; default=0.8) When .spo_p_mode=Factor, .spo_p_fact specifies the factor by which paste pad heights are sized relative to their SMD pads. For example, 0.9 means area is 90% of SMD pad.	Solderpaste Nonstandard Factor Value
.spo_p_mode	Option	Feature; Comp.	(Distance, Area) Defines how paste pads for non-standard symbol SMD pads are sized: by distance, or area.	Create Solderpaste Nonstandard Mode
.spo_p_val	Float	Feature; Comp.	(-500 to 500; default=5) When .spo_p_ mode=Distance, .spo_p_val is the reduction/expansion of the paste pad width relative to the SMD pad width expressed in mils or microns. For example, .sp_ p_val=5.0 mils shrinks paste pad by 5.0 mils (2.5 mils on each side) relative to SMD pad width. Positive number results in smaller paste pad, negative number in larger paste pad. (UNITS=MIL_ MICRON)	Create Solderpaste Nonstandard Mode

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.spo_s_fact	Float	Feature; Comp.	(0.3 <-> 2.0; default=0.8) When .spo_s_mode=Factor, .spo_s_fact specifies the factor by which paste pad heights are sized relative to their non-standard SMD pads. For example, 0.9 means height is 90% of SMD pad.	Solderpaste Symmetric Factor Value
.spo_s_mode	Option	Feature; Comp.	(Distance, Factor, Value, Area) Defines how heights of paste pads for symmetric SMD pads are sized: by distance, factor, value, area.	Create Solderpaste Symmetric Mode
.spo_s_val	Float	Feature; Comp.	(-500 to 500; default=5) When .spo_s_ mode=Distance, .spo_s_val is the reduction/expansion expressed in mils or microns of the paste pad width relative to their non-standard SMD pad width. When .spo_h_mode=Value, .spo_h_val becomes the absolute size of the paste pad. (UNITS=MIL_ MICRON)	Solderpaste Symmetric Relative Value
.spo_shape	Text	Feature; Comp.	Specifies the symbol to be used as the solder paste pad applied to a feature (smd pad) or to the toeprints of a component. The initial orientation of the symbol is also affected by the .spo_ shape_rotate attribute when defined.	Apply Solderpaste Symbol

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.spo_shape_rotate	Float	Feature; Comp.	(0 to 360; default=0) Specifies the initial rotation of the symbol defined in the .spo_shape attribute. Both .spo_shape_rotate and .spo_shape should appear as a pair at the level at which they are activated (component or feature). This means, for example, that an .spo_shape_rotate defined without a corresponding .spo_shape in the feature level is ignored.	Apply Solderpaste Rotation
.spo_shape_stretch	Boolean	Feature; Comp.	(Default=NO) Specifies that the symbol defined in .spo_shape is to be stretched to fit the copper pad dimensions. The stretch limits are determined by applying the SPO width and height parameters (pp_w_*, pp_h_*) or attributes (.spo_w_*, .spo_h_*) on the copper pad bounding box. Both .spo_shape_stretch and .spo_shape should appear as a pair at the level at which they are activated (component or feature). This means, for example, that an .spo_shape_stretch defined without a corresponding .spo_shape in the feature level is ignored.	Extend Solderpaste Symbol

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.spo_w_fact	Float	Feature, Comp.	(0.3 <-> 2.0; default=0.8) When .spo_w_ mode=Factor, .spo_w_fact specifies the factor by which paste pad widths are sized relative to their SMD pads. For example, .spo_w_ fact=0.9 width of paste pad is 90% of width of SMD pad.	Solderpaste Width Factor Value
.spo_w_mode	Option	Feature, Comp.	(Distance; Factor; Value; Area) Defines how widths of paste pads are sized: by distance, factor or value.	Create Solderpaste Wide Mode
.spo_w_val	Float	Feature, Comp.	(-500 to 500; default=5) When .spo_w_ mode=Distance, .spo_w_val is the reduction/expansion expressed in mils or microns of the paste pad width relative to the SMD pad width. For example, .sp_w_ val=5.0 mils shrinks paste pad by 5.0 mils (2.5 mils on each side) relative to SMD pad width. Positive number results in smaller paste pad, negative number in larger paste pad. When .spo_w_ mode=Value, .spo_w_val becomes the absolute width of the paste pad (for example, 5.0 mils becomes the actual width of the paste pad). (UNITS=MIL_ MICRON)	Solderpaste Width Relative Value
.station	Text	Comp.; Pkg.	NOT USED (Min_len= 0, Max_len=255)	ALE Station

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.string	Text	Feature	(0 to 1000) For nomenclature features, the value of this attribute is the original text string of which the feature is a part. See “TEXT Attribute Definition” on page 70.	Copper Text String
.string_angle	Float	Feature	(0 to 360) For nomenclature features, the value of this attribute is the original (in the input file) text rotation angle of which the feature is a part.	String Angle
.string_justification	Option	Feature	String justification: tl, tc, tr, cl, cc, cr, bl, bc, br **tl, tc, tr** — top-left, top-center, top-right **cl, cc, cr** — center-left, center-center, center-right **bl, bc, br** — bottom-left, bottom-center, bottom-right (Default=**bl**)	String Justification
.string_mirrored	Boolean	Feature	(YES, NO) Assigned to mirrored strings.	String Mirror
.tear_drop	Boolean	Feature	(YES, NO) Assigned to features representing a tear drop.	Tear Drop Feature
.technology	Text	Product Model	(0-100) Defines the technology used in creating the product model.	Technology
.test_point	Boolean	Feature	(YES, NO) Assigned to features that are used for In-Circuit Testing operations.	ICT Test Point

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.test_potential	Option	Feature	An attribute attached to features being considered as testpoints (potential testpoints) for In-Circuit Testing operations. potential_tp_by_analysis—a feature meeting all criteria of the Testpoint Allocation Action. potential_tp_manually—a feature to be used as a testpoint though it does not meet all criteria. not_potential_tp_ manually—a feature not to be used as a testpoint even though it meets all criteria.	Test Point Potential
.testpoint_count	Integer	Net	(-1 to 10000) Specifies the number of testpoints expected on this net. If this variable is not defined, or its value is **-1**, the number of expected testpoints in unlimited. A value of **-1** is given to a net that does not require a specific number of test points.	Expected Test Point Count
.testpoint_name	Text	Feature	(0 to 64) Name of the testpoint.	Testpoint Name
.testprobe_diameter	Float	Feature	(10 to 10000; default=30) Used to provide information on the size of test probes. (UNITS=MIL_MICRON)	Test Probe Diameter
.thickness_over_cu	Float	Step	(1.0 to 1000.0; default=1.0) (UNITS=MIL_MICRON)	Total Thickness over Copper
.thickness_over_sm	Float	Step	(1.0 to 1000.0; default=1.0) (UNITS=MIL_MICRON)	Total Thickness over Soldermask
.thvpad_required	Boolean	Comp.; Pkg.	(Default=NO) Assigned to components that require thieving pads (such as fine pitch SOIC).	Thieving Pad Required

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.toep_nochk_o_side	Boolean	Comp.; Pkg.	(Default=NO) Assigned to components so that their toeprints on the opposite side will not be taken into account.	Toeprint Spacing on Placement Side
.toep_spacing_req	Float	Comp.; Pkg.	(1 to 500; default=5) Defines for components the maximum pad to pad spacing measurements. (UNITS=MIL_MICRON)	Required Toeprint Spacing
.tooling_hole	Boolean	Feature	(YES, NO) Used on drill features to indicate that they are tooling holes.	Tooling Hole
.user_bom_rev	Text	Comp.; Pkg.	(0-1000) Used to describe user modified component extensions.	User Define Extension
.variant_list	Text	Product Model	(0-1000) Consists of a list of all possible variants of a product model. The list contains variant names separated by a colon (:). See “Component Variant Implementation” on page 35.	Product Model Variant List
.vcut	Boolean	Feature	(YES, NO) Assigned to rout features that are cut in a V-shape.	V-Score Edge
.via_type	Option	Feature	(Drilled; Laser; Photo) Assigned to via drills for further classification.	Via Type
.viacap_layer	Option	Step	(Top; Bottom; Both; None; default=None) Defines on which layer via capping can occur, if any.	Via Capping on Layer

Table 2-1. System Attributes Supported by ODB++ (cont.)

Attribute	Type	Entity	Description	Display Name
.wheel_type	Option	Wheel	(Gerber; Tools) Specifies for a wheel whether it is used for Gerber files translation or for drill file translation.	Wheel Type
.z0impedance	Float	Layer	(0.0 to 10000.0; default=0.0) The typical characteristic impedance (in ohms) required for a layer.	Z0/Impedance

ODB++ System Attributes by Class

ODB++ system attributes are listed here by class. For each class, the name of the sysattr file is indicated, in which attribute definitions are stored for the class.

For a description of attributes, see “ODB++ System Attributes - Alphabetical List” on page 189.

See “Attribute Classes” on page 22.

Attributes are listed for these classes:

Product (sysattr)

These attributes support the product:

Table 2-2. ODB++ Product Attributes

ODB++ Product Attributes
.all_eda_layers
.artwork
.board_thickness
.break_away
.brk_point

Table 2-2. ODB++ Product Attributes

ODB++ Product Attributes
.bulk_resistivity
.cad_local_footprint_change
.cad_package_name
.cad_part_override
.centroid_correction_x
.centroid_correction_y
.comment
.comp_height
.comp_height_area
.comp_height_max
.comp_htol_minus
.comp_htol_plus
.comp_polarity
.comp_variant_list
.comp_weight
.copper_weight
.critical_net
.critical_tp
.cu_base
.current_variant
.customer
.data_source
.desc1....10
.design_center
.design_origin_x
.design_origin_y
.dielectric_constant
.diff_pair
.drill
.drill_layer_direction

Table 2-2. ODB++ Product Attributes

ODB++ Product Attributes
.ecmp_layer_tech
.ecmp_max_value
.ecmp_min_value
.ecmp_name
.ecmp_value
.eda_dimension_id
.eda_layers
.fiducial_rdlist
.geometry
.gold_plating
.hatch
.hatch_border
.hatch_serrated_border
.ignore_net
.imp_line
.inp_file
.is_buried
.is_capped
.is_wirebonded
.layer_dielectric
.layer_hdi_type
.loss_tangent
.lyr_prf_ref
.mechanical
.merge_processes
.mount_hole
.n_electric
.net_name
.net_physical_type
.net_point

Table 2-2. ODB++ Product Attributes

ODB++ Product Attributes
.net_spacing_type
.net_type
.neutralization_angle
.neutralization_info
.neutralization_reviewed
.neutralization_ss_layers
.no_pop
.nomenclature
.non_tp
.package_version
.pad_usage
.part_desc1...10
.patch
.pattern_fill
.pf_optimized
.pitch
.plated_type
.polarity_marker
.primary_side
.shave
.sip
.sliver_fill
.smd
.source_llayer
.string
.string_angle
.string_justification
.string_mirrored
.tear_drop
.technology

Table 2-2. ODB++ Product Attributes

ODB++ Product Attributes
.test_point
.test_potential
.testpoint_name
.testprobe_diameter
.thickness_over_cu
.thickness_over_sm
.tooling_hole
.user_bom_rev
.variant_list
.vcut
.via_type
.viacap_layer
.wheel_type
.z0impedance

DFM (sysattr.dfm)

These attributes support Design For Manufacture (DFM):

Table 2-3. ODB++ DFM Attributes

ODB++ DFM Attributes
.ar_pad_drill_bottom_max
.ar_pad_drill_bottom_min
.ar_pad_drill_inner_max
.ar_pad_drill_inner_min
.ar_pad_drill_top_max
.ar_pad_drill_top_min
.ar_sm_drill_bottom_max
.ar_sm_drill_bottom_min
.ar_sm_drill_top_max
.ar_sm_drill_top_min

Table 2-3. ODB++ DFM Attributes

ODB++ DFM Attributes
.ar_sm_pad_bottom_max
.ar_sm_pad_bottom_min
.ar_sm_pad_top_max
.ar_sm_pad_top_min
.area_name
.center_fiducial
.comp_ign_spacing
.comp_ignore
.comp_mount_type
.comp_type
.comp_type2
.dpair_gap
.drc_assembly_lyrs
.drc_bend_keepout
.drc_board
.drc_comp_height
.drc_comp_height_lyr
.drc_comp_keepin
.drc_comp_keepin_lyr
.drc_comp_keepout
.drc_comp_keepout_lyr
.drc_etch_lyrs
.drc_etch_lyrs_all
.drc_etch_lyrs_bit
.drc_max_height
.drc_mech
.drc_min_height
.drc_pad_keepout
.drc_pad_keepout_lyr
.drc_plane_keepout

Table 2-3. ODB++ DFM Attributes

ODB++ DFM Attributes
.drc_plane_keepout_lyr
.drc_ref_des
.drc_route_keepin
.drc_route_keepin_lyr
.drc_route_keepout
.drc_route_keepout_lyr
.drc_tp_keepin
.drc_tp_keepin_lyr
.drc_tp_keepout
.drc_tp_keepout_lyr
.drc_trace_keepout
.drc_trace_keepout_lyr
.drc_via_keepout
.drc_via_keepout_lyr
.eclass_accumulative_parallel_dist_list
.eclass_accumulative_parallel_max_length_list
.eclass_impedance
.eclass_individual_parallel_dist_list
.eclass_individual_parallel_max_length_list
.eclass_individual_parallel_min_jog_list
.eclass_max_stub_length
.eclass_max_via_count
.eclass_min_stub_length
.eclass_rise_time
.eclass_voltage_swing
.electrical_class
.fab_drc
.feature_ignore
.guard_comp
.hdi_drc

Table 2-3. ODB++ DFM Attributes

ODB++ DFM Attributes
.ind_orient_req
.is_shadowed
.label_clearance
.layer_class
.local_fiducial_dist
.min_line_width
.needs_guarding
.net_length_max
.net_length_min
.no_copper_shape_under
.no_fiducial_check
.no_hole_under
.no_protrude_board
.no_text_under
.no_tp_under
.no_trace_under
.no_uncap_via_under
.num_local_fiducials
.otherside_keepout
.physical_type
.spacing_req
.testpoint_count
.thvpad_required
.toep_nochk_o_side
.toep_spacing_req

Fabrication (sysattr.fab)

These attributes support fabrication:

Table 2-4. ODB++ Fabrication Attributes

ODB++ Fabrication Attributes
.aoi_cpbm
.aoi_cpcu
.aoi_drbm
.aoi_drcu
.aoi_value
.array_with_rotation
.bit
.comp
.cut_line
.drc_add_rad
.drc_min_space
.drc_min_width
.drill_flag
.drill_noopt
.drill_sr_zero
.drill_stage
.et_adjacency
.et_align
.extended
.feed
.fiducial_name
.fill_dx
.fill_dy
.foot_down
.full_plane
.global_camtek_aoiset
.image_dx

Table 2-4. ODB++ Fabrication Attributes

ODB++ Fabrication Attributes
.image_dy
.lpol_done
.lpol_surf
.orbotech_plot_stamp
.out_angle
.out_break
.out_comp
.out_drill_full
.out_drill_optional
.out_drill_order
.out_flag
.out_mirror
.out_name
.out_orig
.out_polarity
.out_rout_optional
.out_rout_order
.out_scale
.out_x_scale
.out_y_scale
.pilot_hole
.rout_chain
.rout_flag
.speed

Assembly (sysattr.assy)

These attributes support assembly.

Table 2-5. ODB++ Assembly Attributes

ODB++ Assembly Attributes
.assembly_proc_bottom
.assembly_proc_top
.axi_direction
.device_style
.device_type
.device_value
.device_value_ntol
.device_value_ptol
.fs_direction_bottom
.fs_direction_top
.gencad_device_ntol
.gencad_device_ptol
.gencad_device_style
.gencad_device_type
.gencad_device_value
.machine_pkg
.mount_stage
.rot_correction
.smt_direction_bottom
.smt_direction_top
.spo_h_fact
.spo_h_mode
.spo_h_val
.spo_move_center
.spo_p_fact
.spo_p_mode
.spo_p_val

Table 2-5. ODB++ Assembly Attributes

ODB++ Assembly Attributes
.spo_s_fact
.spo_s_mode
.spo_s_val
.spo_shape
.spo_shape_rotate
.spo_shape_stretch
.spo_w_fact
.spo_w_mode
.spo_w_val
.station

Test (sysattr.test)

These attributes support testing:

Table 2-6. ODB++ Test Attributes

ODB++ Test Attributes
.board_mark
.hp3070_comment
.hp3070_common_pin
.hp3070_contact_pin
.hp3070_device
.hp3070_fail_msg
.hp3070_hi_value
.hp3070_lo_value
.hp3070_probe_access
.hp3070_seriesr
.hp3070_test
.hp3070_tol_neg
.hp3070_tol_pos
.hp3070_type

Table 2-6. ODB++ Test Attributes

ODB++ Test Attributes
.hp3070_value
.net_test_current
.net_test_type
.net_test_voltage

Generic (sysattr.gen)

These are generic attributes:

Table 2-7. ODB++ Generic Attributes

ODB++ Generic Attributes
.color
.dxf_dimension
.orig_surf
.output_dcode

Appendix 3
Standard Font

This section contains the font definition for the font Standard used for ODB++ text. The file is not part of the ODB++ format, but the geometry of character and letter representations is.

See “fonts (Fonts Used in Product Model)” on page 76.

These topics are discussed:

Font Format Specific Information

The font file contains these types of line:

- **XSIZE** and **YSIZE** — Defines the bounding box size for letters.
- **OFFSET** — Fixed offset of letters along the line.
- **CHAR <symbol>** — Begins the character description section. **<symbol>** is the ASCII representation for the character or letter.
- **CHAR** — Begins the representation of a character or letter, followed by records that are formatted using LINE records. The collection of lines representing the character are written around (0, 0) of the coordinate axis.
- **ECHAR** — Ends a character or letter representation.

Standard Font Description

```
XSIZE  0.302000
YSIZE  0.302000
OFFSET 0.000000


CHAR !
LINE 0.000000 0.000000 0.000000 0.200000 P R 0.012000
LINE 0.000000 -0.100000 0.000000 -0.100000 P R 0.012000
ECHAR

CHAR "
LINE -0.050000 0.100000 -0.050000 0.200000 P R 0.012000
LINE 0.050000 0.100000 0.050000 0.200000 P R 0.012000
```

```
ECHAR

CHAR #
LINE -0.050000 -0.100000 -0.050000 0.200000 P R 0.012000
LINE 0.050000 -0.100000 0.050000 0.200000 P R 0.012000
LINE -0.100000 0.000000 0.100000 0.000000 P R 0.012000
LINE -0.100000 0.100000 0.100000 0.100000 P R 0.012000
ECHAR

CHAR $
LINE 0.000000 -0.100000 0.000000 0.200000 P R 0.012000
LINE -0.100000 -0.050000 0.050000 -0.050000 P R 0.012000
LINE 0.050000 -0.050000 0.100000 0.000000 P R 0.012000
LINE 0.100000 0.000000 0.050000 0.050000 P R 0.012000
LINE 0.050000 0.050000 -0.050000 0.050000 P R 0.012000
LINE -0.050000 0.050000 -0.100000 0.100000 P R 0.012000
LINE -0.100000 0.100000 -0.050000 0.150000 P R 0.012000
LINE -0.050000 0.150000 0.100000 0.150000 P R 0.012000
ECHAR

CHAR %
LINE -0.100000 -0.100000 0.100000 0.200000 P R 0.012000
LINE -0.075000 0.175000 -0.075000 0.175000 P R 0.012000
LINE 0.075000 -0.075000 0.075000 -0.075000 P R 0.012000
ECHAR

CHAR &
LINE 0.100000 -0.100000 -0.100000 0.100000 P R 0.012000
LINE -0.100000 0.100000 -0.100000 0.150000 P R 0.012000
LINE -0.100000 0.150000 -0.050000 0.200000 P R 0.012000
LINE -0.050000 0.200000 0.000000 0.150000 P R 0.012000
LINE 0.000000 0.150000 0.000000 0.100000 P R 0.012000
LINE 0.000000 0.100000 -0.100000 0.000000 P R 0.012000
LINE -0.100000 0.000000 -0.100000 -0.050000 P R 0.012000
LINE -0.100000 -0.050000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 0.000000 -0.100000 P R 0.012000
LINE 0.000000 -0.100000 0.100000 0.000000 P R 0.012000
ECHAR

CHAR '
LINE 0.000000 0.050000 0.050000 0.150000 P R 0.012000
LINE 0.025000 0.175000 0.025000 0.175000 P R 0.012000
ECHAR

CHAR (
LINE 0.050000 -0.100000 -0.050000 0.000000 P R 0.012000
LINE -0.050000 0.000000 -0.050000 0.100000 P R 0.012000
LINE -0.050000 0.100000 0.050000 0.200000 P R 0.012000
ECHAR

CHAR )
LINE -0.050000 -0.100000 0.050000 0.000000 P R 0.012000
LINE 0.050000 0.000000 0.050000 0.100000 P R 0.012000
LINE 0.050000 0.100000 -0.050000 0.200000 P R 0.012000
ECHAR

CHAR *
LINE -0.100000 -0.050000 0.100000 0.150000 P R 0.012000
```

```
LINE -0.100000 0.150000 0.100000 -0.050000 P R 0.012000
LINE 0.000000 -0.100000 0.000000 0.200000 P R 0.012000
ECHAR

CHAR +
LINE -0.100000 0.050000 0.100000 0.050000 P R 0.012000
LINE 0.000000 -0.050000 0.000000 0.150000 P R 0.012000
ECHAR

CHAR ,
LINE 0.000000 -0.100000 0.050000 0.000000 P R 0.012000
LINE 0.025000 0.025000 0.025000 0.025000 P R 0.012000
ECHAR

CHAR -
LINE -0.100000 0.050000 0.100000 0.050000 P R 0.012000
ECHAR

CHAR .
LINE 0.025000 -0.075000 0.025000 -0.075000 P R 0.012000
ECHAR

CHAR /
LINE -0.100000 -0.050000 0.100000 0.150000 P R 0.012000
ECHAR

CHAR 0
LINE -0.100000 -0.050000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 0.100000 -0.050000 P R 0.012000
LINE 0.100000 -0.050000 0.100000 0.150000 P R 0.012000
LINE 0.100000 0.150000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 -0.050000 0.200000 P R 0.012000
LINE -0.050000 0.200000 -0.100000 0.150000 P R 0.012000
LINE -0.100000 0.150000 -0.100000 -0.050000 P R 0.012000
LINE -0.100000 -0.050000 0.100000 0.150000 P R 0.012000
ECHAR

CHAR 1
LINE -0.050000 0.150000 0.000000 0.200000 P R 0.012000
LINE 0.000000 0.200000 0.000000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 0.050000 -0.100000 P R 0.012000
ECHAR

CHAR 2
LINE -0.100000 0.150000 -0.050000 0.200000 P R 0.012000
LINE -0.050000 0.200000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 0.100000 0.150000 P R 0.012000
LINE 0.100000 0.150000 0.100000 0.100000 P R 0.012000
LINE 0.100000 0.100000 0.050000 0.050000 P R 0.012000
LINE 0.050000 0.050000 0.000000 0.050000 P R 0.012000
LINE 0.000000 0.050000 -0.100000 -0.100000 P R 0.012000
LINE -0.100000 -0.100000 0.100000 -0.100000 P R 0.012000
ECHAR

CHAR 3
LINE -0.100000 0.200000 0.100000 0.200000 P R 0.012000
LINE 0.100000 0.200000 0.100000 0.100000 P R 0.012000
```

```
LINE 0.100000 0.100000 0.050000 0.050000 P R 0.012000
LINE 0.050000 0.050000 0.000000 0.050000 P R 0.012000
LINE 0.050000 0.050000 0.100000 0.000000 P R 0.012000
LINE 0.100000 0.000000 0.100000 -0.050000 P R 0.012000
LINE 0.100000 -0.050000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 -0.100000 -0.050000 P R 0.012000
ECHAR

CHAR 4
LINE 0.050000 -0.100000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 -0.100000 0.000000 P R 0.012000
LINE -0.100000 0.000000 0.100000 0.000000 P R 0.012000
ECHAR

CHAR 5
LINE 0.100000 0.200000 -0.100000 0.200000 P R 0.012000
LINE -0.100000 0.200000 -0.100000 0.100000 P R 0.012000
LINE -0.100000 0.100000 0.050000 0.100000 P R 0.012000
LINE 0.050000 0.100000 0.100000 0.050000 P R 0.012000
LINE 0.100000 0.050000 0.100000 -0.050000 P R 0.012000
LINE 0.100000 -0.050000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 -0.100000 -0.050000 P R 0.012000
ECHAR

CHAR 6
LINE 0.100000 0.200000 0.000000 0.200000 P R 0.012000
LINE 0.000000 0.200000 -0.100000 0.100000 P R 0.012000
LINE -0.100000 0.100000 -0.100000 -0.050000 P R 0.012000
LINE -0.100000 -0.050000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 0.100000 -0.050000 P R 0.012000
LINE 0.100000 -0.050000 0.100000 0.000000 P R 0.012000
LINE 0.100000 0.000000 0.050000 0.050000 P R 0.012000
LINE 0.050000 0.050000 -0.100000 0.050000 P R 0.012000
ECHAR

CHAR 7
LINE -0.100000 0.200000 0.100000 0.200000 P R 0.012000
LINE 0.100000 0.200000 -0.050000 -0.100000 P R 0.012000
ECHAR

CHAR 8
LINE -0.050000 0.200000 0.050000 0.200000 P R 0.012000
LINE -0.050000 0.050000 0.050000 0.050000 P R 0.012000
LINE -0.050000 -0.100000 0.050000 -0.100000 P R 0.012000
LINE -0.100000 0.150000 -0.100000 0.100000 P R 0.012000
LINE 0.100000 0.150000 0.100000 0.100000 P R 0.012000
LINE -0.100000 0.000000 -0.100000 -0.050000 P R 0.012000
LINE 0.100000 0.000000 0.100000 -0.050000 P R 0.012000
LINE 0.050000 0.200000 0.100000 0.150000 P R 0.012000
LINE -0.050000 0.200000 -0.100000 0.150000 P R 0.012000
LINE 0.050000 0.050000 0.100000 0.100000 P R 0.012000
LINE -0.050000 0.050000 -0.100000 0.100000 P R 0.012000
LINE 0.050000 0.050000 0.100000 0.000000 P R 0.012000
LINE -0.050000 0.050000 -0.100000 0.000000 P R 0.012000
LINE 0.050000 -0.100000 0.100000 -0.050000 P R 0.012000
```

```
LINE -0.050000 -0.100000 -0.100000 -0.050000 P R 0.012000
ECHAR

CHAR 9
LINE -0.100000 -0.100000 0.000000 -0.100000 P R 0.012000
LINE 0.000000 -0.100000 0.100000 0.000000 P R 0.012000
LINE 0.100000 0.000000 0.100000 0.150000 P R 0.012000
LINE 0.100000 0.150000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 -0.050000 0.200000 P R 0.012000
LINE -0.050000 0.200000 -0.100000 0.150000 P R 0.012000
LINE -0.100000 0.150000 -0.100000 0.100000 P R 0.012000
LINE -0.100000 0.100000 -0.050000 0.050000 P R 0.012000
LINE -0.050000 0.050000 0.100000 0.050000 P R 0.012000
ECHAR

CHAR :
LINE 0.000000 -0.05000 0.000000 -0.05000 P R 0.012000
LINE 0.000000  0.15000 0.000000  0.15000 P R 0.012000
ECHAR

CHAR ;
LINE 0.000000 0.025000 -0.050000 -0.100000 P R 0.012000
LINE 0.000000 0.025000 0.000000 0.025000 P R 0.012000
LINE 0.000000 0.175000 0.000000 0.175000 P R 0.012000
ECHAR

CHAR <
LINE 0.100000 0.200000 -0.050000 0.050000 P R 0.012000
LINE -0.050000 0.050000 0.100000 -0.100000 P R 0.012000
ECHAR

CHAR =
LINE -0.100000 0.100000 0.100000 0.100000 P R 0.012000
LINE -0.100000 0.000000 0.100000 0.000000 P R 0.012000
ECHAR

CHAR >
LINE -0.100000 0.200000 0.050000 0.050000 P R 0.012000
LINE 0.050000 0.050000 -0.100000 -0.100000 P R 0.012000
ECHAR

CHAR ?
LINE -0.100000 0.150000 -0.050000 0.200000 P R 0.012000
LINE -0.050000 0.200000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 0.100000 0.150000 P R 0.012000
LINE 0.100000 0.150000 0.000000 0.050000 P R 0.012000
LINE 0.000000 0.050000 0.000000 0.000000 P R 0.012000
LINE 0.000000 -0.100000 0.000000 -0.100000 P R 0.012000
ECHAR

CHAR @
LINE 0.050000 -0.100000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 -0.100000 -0.050000 P R 0.012000
LINE -0.100000 -0.050000 -0.100000 0.150000 P R 0.012000
LINE -0.100000 0.150000 -0.050000 0.200000 P R 0.012000
LINE -0.050000 0.200000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 0.100000 0.150000 P R 0.012000
LINE 0.100000 0.150000 0.100000 0.000000 P R 0.012000
```

```
LINE 0.100000 0.000000 0.000000 0.000000 P R 0.012000
LINE 0.000000 0.000000 0.000000 0.100000 P R 0.012000
LINE 0.000000 0.100000 0.100000 0.100000 P R 0.012000
ECHAR

CHAR A
LINE -0.100000 -0.100000 -0.100000 0.150000 P R 0.012000
LINE -0.100000 0.150000 -0.050000 0.200000 P R 0.012000
LINE -0.050000 0.200000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 0.100000 0.150000 P R 0.012000
LINE 0.100000 0.150000 0.100000 -0.100000 P R 0.012000
LINE -0.100000 0.050000 0.100000 0.050000 P R 0.012000
ECHAR

CHAR B
LINE -0.100000 0.200000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 0.100000 0.150000 P R 0.012000
LINE 0.100000 0.150000 0.100000 0.100000 P R 0.012000
LINE 0.100000 0.100000 0.050000 0.050000 P R 0.012000
LINE 0.050000 0.050000 0.100000 0.000000 P R 0.012000
LINE 0.100000 0.000000 0.100000 -0.050000 P R 0.012000
LINE 0.100000 -0.050000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 -0.100000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 -0.050000 0.200000 P R 0.012000
LINE -0.050000 0.050000 0.050000 0.050000 P R 0.012000
ECHAR

CHAR C
LINE 0.100000 -0.050000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 -0.100000 -0.050000 P R 0.012000
LINE -0.100000 -0.050000 -0.100000 0.150000 P R 0.012000
LINE -0.100000 0.150000 -0.050000 0.200000 P R 0.012000
LINE -0.050000 0.200000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 0.100000 0.150000 P R 0.012000
ECHAR

CHAR D
LINE -0.100000 0.200000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 0.100000 0.150000 P R 0.012000
LINE 0.100000 0.150000 0.100000 -0.050000 P R 0.012000
LINE 0.100000 -0.050000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 -0.100000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 -0.050000 0.200000 P R 0.012000
ECHAR

CHAR E
LINE -0.100000 0.200000 0.100000 0.200000 P R 0.012000
LINE -0.100000 -0.100000 0.100000 -0.100000 P R 0.012000
LINE -0.100000 0.050000 0.050000 0.050000 P R 0.012000
LINE -0.100000 0.200000 -0.100000 -0.100000 P R 0.012000
ECHAR

CHAR F
LINE -0.100000 0.200000 0.100000 0.200000 P R 0.012000
LINE -0.100000 0.050000 0.050000 0.050000 P R 0.012000
LINE -0.100000 0.200000 -0.100000 -0.100000 P R 0.012000
ECHAR
```

```
CHAR G
LINE -0.050000 0.200000 0.100000 0.200000 P R 0.012000
LINE -0.050000 0.200000 -0.100000 0.150000 P R 0.012000
LINE -0.100000 0.150000 -0.100000 -0.050000 P R 0.012000
LINE -0.100000 -0.050000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 0.100000 -0.100000 P R 0.012000
LINE 0.100000 -0.100000 0.100000 0.050000 P R 0.012000
LINE 0.100000 0.050000 0.050000 0.050000 P R 0.012000
ECHAR

CHAR H
LINE -0.100000 0.200000 -0.100000 -0.100000 P R 0.012000
LINE 0.100000 0.200000 0.100000 -0.100000 P R 0.012000
LINE -0.100000 0.050000 0.100000 0.050000 P R 0.012000
ECHAR

CHAR I
LINE -0.050000 0.200000 0.050000 0.200000 P R 0.012000
LINE -0.050000 -0.100000 0.050000 -0.100000 P R 0.012000
LINE 0.000000 0.200000 0.000000 -0.100000 P R 0.012000
ECHAR

CHAR J
LINE 0.100000 0.200000 0.100000 -0.050000 P R 0.012000
LINE 0.100000 -0.050000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 -0.100000 -0.050000 P R 0.012000
ECHAR

CHAR K
LINE -0.100000 0.200000 -0.100000 -0.100000 P R 0.012000
LINE 0.100000 0.200000 -0.100000 0.050000 P R 0.012000
LINE -0.100000 0.050000 0.100000 -0.100000 P R 0.012000
ECHAR

CHAR L
LINE -0.100000 0.200000 -0.100000 -0.100000 P R 0.012000
LINE -0.100000 -0.100000 0.100000 -0.100000 P R 0.012000
ECHAR

CHAR M
LINE -0.100000 -0.100000 -0.100000 0.200000 P R 0.012000
LINE -0.100000 0.200000 0.000000 0.050000 P R 0.012000
LINE 0.000000 0.050000 0.100000 0.200000 P R 0.012000
LINE 0.100000 0.200000 0.100000 -0.100000 P R 0.012000
ECHAR

CHAR N
LINE -0.100000 -0.100000 -0.100000 0.200000 P R 0.012000
LINE -0.100000 0.200000 0.100000 -0.100000 P R 0.012000
LINE 0.100000 -0.100000 0.100000 0.200000 P R 0.012000
ECHAR

CHAR O
LINE -0.100000 -0.050000 -0.100000 0.150000 P R 0.012000
LINE -0.100000 0.150000 -0.050000 0.200000 P R 0.012000
LINE -0.050000 0.200000 0.050000 0.200000 P R 0.012000
```

```
LINE 0.050000 0.200000 0.100000 0.150000 P R 0.012000
LINE 0.100000 0.150000 0.100000 -0.050000 P R 0.012000
LINE 0.100000 -0.050000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 -0.100000 -0.050000 P R 0.012000
ECHAR

CHAR P
LINE -0.100000 -0.100000 -0.100000 0.200000 P R 0.012000
LINE -0.100000 0.200000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 0.100000 0.150000 P R 0.012000
LINE 0.100000 0.150000 0.100000 0.100000 P R 0.012000
LINE 0.100000 0.100000 0.050000 0.050000 P R 0.012000
LINE 0.050000 0.050000 -0.100000 0.050000 P R 0.012000
ECHAR

CHAR Q
LINE -0.100000 -0.050000 -0.100000 0.150000 P R 0.012000
LINE -0.100000 0.150000 -0.050000 0.200000 P R 0.012000
LINE -0.050000 0.200000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 0.100000 0.150000 P R 0.012000
LINE 0.100000 0.150000 0.100000 0.000000 P R 0.012000
LINE 0.100000 0.000000 0.000000 -0.100000 P R 0.012000
LINE 0.000000 -0.100000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 -0.100000 -0.050000 P R 0.012000
LINE 0.000000 0.000000 0.100000 -0.100000 P R 0.012000
ECHAR

CHAR R
LINE -0.100000 -0.100000 -0.100000 0.200000 P R 0.012000
LINE -0.100000 0.200000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 0.100000 0.150000 P R 0.012000
LINE 0.100000 0.150000 0.100000 0.100000 P R 0.012000
LINE 0.100000 0.100000 0.050000 0.050000 P R 0.012000
LINE 0.050000 0.050000 -0.100000 0.050000 P R 0.012000
LINE -0.050000 0.050000 0.100000 -0.100000 P R 0.012000
ECHAR

CHAR S
LINE -0.100000 -0.050000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 0.100000 -0.050000 P R 0.012000
LINE 0.100000 -0.050000 0.100000 0.000000 P R 0.012000
LINE 0.100000 0.000000 0.050000 0.050000 P R 0.012000
LINE 0.050000 0.050000 -0.050000 0.050000 P R 0.012000
LINE -0.050000 0.050000 -0.100000 0.100000 P R 0.012000
LINE -0.100000 0.100000 -0.100000 0.150000 P R 0.012000
LINE -0.100000 0.150000 -0.050000 0.200000 P R 0.012000
LINE -0.050000 0.200000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 0.100000 0.150000 P R 0.012000
ECHAR

CHAR T
LINE -0.100000 0.200000 0.100000 0.200000 P R 0.012000
LINE 0.000000 0.200000 0.000000 -0.100000 P R 0.012000
ECHAR

CHAR U
```

```
LINE -0.100000 0.200000 -0.100000 -0.050000 P R 0.012000
LINE -0.100000 -0.050000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 0.100000 -0.050000 P R 0.012000
LINE 0.100000 -0.050000 0.100000 0.200000 P R 0.012000
ECHAR

CHAR V
LINE -0.100000 0.200000 0.000000 -0.100000 P R 0.012000
LINE 0.000000 -0.100000 0.100000 0.200000 P R 0.012000
ECHAR

CHAR W
LINE -0.100000 0.200000 -0.100000 -0.100000 P R 0.012000
LINE -0.100000 -0.100000 0.000000 0.050000 P R 0.012000
LINE 0.000000 0.050000 0.100000 -0.100000 P R 0.012000
LINE 0.100000 -0.100000 0.100000 0.200000 P R 0.012000
ECHAR

CHAR X
LINE -0.100000 0.200000 0.100000 -0.100000 P R 0.012000
LINE -0.100000 -0.100000 0.100000 0.200000 P R 0.012000
ECHAR

CHAR Y
LINE -0.100000 0.200000 0.000000 0.050000 P R 0.012000
LINE 0.000000 0.050000 0.100000 0.200000 P R 0.012000
LINE 0.000000 0.050000 0.000000 -0.100000 P R 0.012000
ECHAR

CHAR Z
LINE -0.100000 0.200000 0.100000 0.200000 P R 0.012000
LINE 0.100000 0.200000 -0.100000 -0.100000 P R 0.012000
LINE -0.100000 -0.100000 0.100000 -0.100000 P R 0.012000
ECHAR

CHAR [
LINE 0.050000 0.200000 -0.050000 0.200000 P R 0.012000
LINE -0.050000 0.200000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 0.050000 -0.100000 P R 0.012000
ECHAR

CHAR \
LINE -0.100000 0.200000 0.100000 -0.100000 P R 0.012000
ECHAR

CHAR ]
LINE -0.050000 0.200000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 -0.050000 -0.100000 P R 0.012000
ECHAR

CHAR ^
LINE -0.100000 0.100000 0.000000 0.200000 P R 0.012000
LINE 0.000000 0.200000 0.100000 0.100000 P R 0.012000
ECHAR

CHAR _
```

```
LINE -0.100000 -0.100000 0.100000 -0.100000 P R 0.012000
ECHAR

CHAR `
LINE -0.050000 0.200000 0.000000 0.050000 P R 0.012000
ECHAR

CHAR a
LINE -0.050000 0.100000 0.050000 0.100000 P R 0.012000
LINE 0.050000 0.100000 0.100000 0.050000 P R 0.012000
LINE 0.100000 0.050000 0.100000 -0.100000 P R 0.012000
LINE 0.100000 -0.100000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 -0.100000 -0.050000 P R 0.012000
LINE -0.100000 -0.050000 -0.050000 0.000000 P R 0.012000
LINE -0.050000 0.000000 0.100000 0.000000 P R 0.012000
ECHAR

CHAR b
LINE -0.100000 0.200000 -0.100000 -0.100000 P R 0.012000
LINE -0.100000 -0.100000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 0.100000 -0.050000 P R 0.012000
LINE 0.100000 -0.050000 0.100000 0.050000 P R 0.012000
LINE 0.100000 0.050000 0.050000 0.100000 P R 0.012000
LINE 0.050000 0.100000 0.000000 0.100000 P R 0.012000
LINE 0.000000 0.100000 -0.100000 0.050000 P R 0.012000
ECHAR

CHAR c
LINE 0.100000 0.100000 -0.050000 0.100000 P R 0.012000
LINE -0.050000 0.100000 -0.100000 0.050000 P R 0.012000
LINE -0.100000 0.050000 -0.100000 -0.050000 P R 0.012000
LINE -0.100000 -0.050000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 0.100000 -0.100000 P R 0.012000
ECHAR

CHAR d
LINE 0.100000 0.200000 0.100000 -0.100000 P R 0.012000
LINE 0.100000 -0.100000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 -0.100000 -0.050000 P R 0.012000
LINE -0.100000 -0.050000 -0.100000 0.050000 P R 0.012000
LINE -0.100000 0.050000 -0.050000 0.100000 P R 0.012000
LINE -0.050000 0.100000 0.000000 0.100000 P R 0.012000
LINE 0.000000 0.100000 0.100000 0.050000 P R 0.012000
ECHAR

CHAR e
LINE 0.050000 -0.100000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 -0.100000 -0.050000 P R 0.012000
LINE -0.100000 -0.050000 -0.100000 0.050000 P R 0.012000
LINE -0.100000 0.050000 -0.050000 0.100000 P R 0.012000
LINE -0.050000 0.100000 0.050000 0.100000 P R 0.012000
LINE 0.050000 0.100000 0.100000 0.050000 P R 0.012000
LINE 0.100000 0.050000 0.050000 0.000000 P R 0.012000
LINE 0.050000 0.000000 -0.100000 0.000000 P R 0.012000
ECHAR

CHAR f
LINE -0.050000 -0.100000 -0.050000 0.150000 P R 0.012000
```

```
LINE -0.050000 0.150000 0.000000 0.200000 P R 0.012000
LINE 0.000000 0.200000 0.050000 0.200000 P R 0.012000
LINE 0.050000 0.200000 0.100000 0.150000 P R 0.012000
LINE -0.100000 0.050000 0.000000 0.050000 P R 0.012000
ECHAR

CHAR g
LINE -0.050000 -0.150000 0.050000 -0.150000 P R 0.012000
LINE 0.050000 -0.150000 0.100000 -0.100000 P R 0.012000
LINE 0.100000 -0.100000 0.100000 0.050000 P R 0.012000
LINE 0.100000 0.050000 0.050000 0.100000 P R 0.012000
LINE 0.050000 0.100000 -0.050000 0.100000 P R 0.012000
LINE -0.050000 0.100000 -0.100000 0.050000 P R 0.012000
LINE -0.100000 0.050000 -0.100000 0.000000 P R 0.012000
LINE -0.100000 0.000000 -0.050000 -0.050000 P R 0.012000
LINE -0.050000 -0.050000 0.100000 -0.050000 P R 0.012000
ECHAR

CHAR h
LINE -0.100000 0.200000 -0.100000 -0.100000 P R 0.012000
LINE -0.100000 0.050000 0.000000 0.100000 P R 0.012000
LINE 0.000000 0.100000 0.050000 0.100000 P R 0.012000
LINE 0.050000 0.100000 0.100000 0.050000 P R 0.012000
LINE 0.100000 0.050000 0.100000 -0.100000 P R 0.012000
ECHAR

CHAR i
LINE 0.000000 0.100000 0.000000 -0.050000 P R 0.012000
LINE 0.000000 -0.050000 0.050000 -0.100000 P R 0.012000
LINE 0.000000 0.200000 0.000000 0.200000 P R 0.012000
ECHAR

CHAR j
LINE 0.050000 0.100000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 0.000000 -0.150000 P R 0.012000
LINE 0.000000 -0.150000 -0.050000 -0.150000 P R 0.012000
LINE -0.050000 -0.150000 -0.100000 -0.100000 P R 0.012000
LINE 0.050000 0.200000 0.050000 0.200000 P R 0.012000
ECHAR

CHAR k
LINE -0.100000 0.200000 -0.100000 -0.100000 P R 0.012000
LINE -0.100000 0.000000 0.000000 0.000000 P R 0.012000
LINE 0.000000 0.000000 0.100000 -0.100000 P R 0.012000
LINE -0.050000 0.000000 0.050000 0.100000 P R 0.012000
ECHAR

CHAR l
LINE -0.050000 0.200000 0.000000 0.200000 P R 0.012000
LINE 0.000000 0.200000 0.000000 -0.050000 P R 0.012000
LINE 0.000000 -0.050000 0.050000 -0.100000 P R 0.012000
ECHAR

CHAR m
LINE -0.100000 -0.100000 -0.100000 0.050000 P R 0.012000
LINE -0.100000 0.050000 -0.050000 0.100000 P R 0.012000
LINE -0.050000 0.100000 0.000000 0.050000 P R 0.012000
LINE 0.000000 0.050000 0.000000 0.000000 P R 0.012000
```

```
LINE 0.000000 0.050000 0.050000 0.100000 P R 0.012000
LINE 0.050000 0.100000 0.100000 0.050000 P R 0.012000
LINE 0.100000 0.050000 0.100000 -0.100000 P R 0.012000
ECHAR

CHAR n
LINE -0.100000 0.100000 -0.100000 -0.100000 P R 0.012000
LINE -0.100000 0.000000 0.000000 0.100000 P R 0.012000
LINE 0.000000 0.100000 0.050000 0.100000 P R 0.012000
LINE 0.050000 0.100000 0.100000 0.050000 P R 0.012000
LINE 0.100000 0.050000 0.100000 -0.100000 P R 0.012000
ECHAR

CHAR o
LINE -0.100000 -0.050000 -0.100000 0.050000 P R 0.012000
LINE -0.100000 0.050000 -0.050000 0.100000 P R 0.012000
LINE -0.050000 0.100000 0.050000 0.100000 P R 0.012000
LINE 0.050000 0.100000 0.100000 0.050000 P R 0.012000
LINE 0.100000 0.050000 0.100000 -0.050000 P R 0.012000
LINE 0.100000 -0.050000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 -0.100000 -0.050000 P R 0.012000
ECHAR

CHAR p
LINE -0.100000 -0.150000 -0.100000 0.050000 P R 0.012000
LINE -0.100000 0.050000 -0.050000 0.100000 P R 0.012000
LINE -0.050000 0.100000 0.050000 0.100000 P R 0.012000
LINE 0.050000 0.100000 0.100000 0.050000 P R 0.012000
LINE 0.100000 0.050000 0.100000 0.000000 P R 0.012000
LINE 0.100000 0.000000 0.050000 -0.050000 P R 0.012000
LINE 0.050000 -0.050000 -0.100000 -0.050000 P R 0.012000
ECHAR

CHAR q
LINE 0.100000 -0.150000 0.100000 0.050000 P R 0.012000
LINE 0.100000 0.050000 0.050000 0.100000 P R 0.012000
LINE 0.050000 0.100000 -0.050000 0.100000 P R 0.012000
LINE -0.050000 0.100000 -0.100000 0.050000 P R 0.012000
LINE -0.100000 0.050000 -0.100000 0.000000 P R 0.012000
LINE -0.100000 0.000000 -0.050000 -0.050000 P R 0.012000
LINE -0.050000 -0.050000 0.100000 -0.050000 P R 0.012000
ECHAR

CHAR r
LINE -0.100000 0.100000 -0.100000 -0.100000 P R 0.012000
LINE -0.100000 0.000000 0.000000 0.100000 P R 0.012000
LINE 0.000000 0.100000 0.050000 0.050000 P R 0.012000
ECHAR

CHAR s
LINE -0.100000 -0.100000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 0.100000 -0.050000 P R 0.012000
LINE 0.100000 -0.050000 0.050000 0.000000 P R 0.012000
LINE 0.050000 0.000000 -0.050000 0.000000 P R 0.012000
LINE -0.050000 0.000000 -0.100000 0.050000 P R 0.012000
LINE -0.100000 0.050000 -0.050000 0.100000 P R 0.012000
LINE -0.050000 0.100000 0.100000 0.100000 P R 0.012000
```

```
ECHAR

CHAR t
LINE -0.050000 0.200000 -0.050000 -0.050000 P R 0.012000
LINE -0.050000 -0.050000 0.000000 -0.100000 P R 0.012000
LINE 0.000000 -0.100000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 0.100000 -0.050000 P R 0.012000
LINE -0.100000 0.100000 0.050000 0.100000 P R 0.012000
ECHAR

CHAR u
LINE -0.100000 0.100000 -0.100000 -0.050000 P R 0.012000
LINE -0.100000 -0.050000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 0.000000 -0.100000 P R 0.012000
LINE 0.000000 -0.100000 0.100000 0.000000 P R 0.012000
LINE 0.100000 -0.100000 0.100000 0.100000 P R 0.012000
ECHAR

CHAR v
LINE -0.100000 0.100000 0.000000 -0.100000 P R 0.012000
LINE 0.000000 -0.100000 0.100000 0.100000 P R 0.012000
ECHAR

CHAR w
LINE -0.100000 0.100000 -0.050000 -0.100000 P R 0.012000
LINE -0.050000 -0.100000 0.000000 0.000000 P R 0.012000
LINE 0.000000 0.000000 0.050000 -0.100000 P R 0.012000
LINE 0.050000 -0.100000 0.100000 0.100000 P R 0.012000
ECHAR

CHAR x
LINE -0.100000 0.100000 0.100000 -0.100000 P R 0.012000
LINE -0.100000 -0.100000 0.100000 0.100000 P R 0.012000
ECHAR

CHAR y
LINE -0.100000 0.100000 -0.100000 0.000000 P R 0.012000
LINE -0.100000 0.000000 -0.050000 -0.050000 P R 0.012000
LINE -0.050000 -0.050000 0.100000 -0.050000 P R 0.012000
LINE 0.100000 0.100000 0.100000 -0.100000 P R 0.012000
LINE 0.100000 -0.100000 0.050000 -0.150000 P R 0.012000
LINE 0.050000 -0.150000 -0.050000 -0.150000 P R 0.012000
ECHAR

CHAR z
LINE -0.100000 0.100000 0.100000 0.100000 P R 0.012000
LINE 0.100000 0.100000 -0.100000 -0.100000 P R 0.012000
LINE -0.100000 -0.100000 0.100000 -0.100000 P R 0.012000
ECHAR

CHAR {
LINE 0.050000 0.200000 0.000000 0.200000 P R 0.012000
LINE 0.000000 0.200000 -0.050000 0.150000 P R 0.012000
LINE -0.050000 0.150000 -0.050000 0.100000 P R 0.012000
LINE -0.050000 0.100000 -0.100000 0.050000 P R 0.012000
LINE -0.100000 0.050000 -0.050000 0.000000 P R 0.012000
LINE -0.050000 0.000000 -0.050000 -0.050000 P R 0.012000
LINE -0.050000 -0.050000 0.000000 -0.100000 P R 0.012000
```

```
LINE 0.000000 -0.100000 0.050000 -0.100000 P R 0.012000
ECHAR

CHAR |
LINE 0.000000 0.200000 0.000000 -0.100000 P R 0.012000
ECHAR

CHAR }
LINE -0.050000 0.200000 0.000000 0.200000 P R 0.012000
LINE 0.000000 0.200000 0.050000 0.150000 P R 0.012000
LINE 0.050000 0.150000 0.050000 0.100000 P R 0.012000
LINE 0.050000 0.100000 0.100000 0.050000 P R 0.012000
LINE 0.100000 0.050000 0.050000 0.000000 P R 0.012000
LINE 0.050000 0.000000 0.050000 -0.050000 P R 0.012000
LINE 0.050000 -0.050000 0.000000 -0.100000 P R 0.012000
LINE 0.000000 -0.100000 -0.050000 -0.100000 P R 0.012000
ECHAR

CHAR ~
LINE -0.100000 0.150000 -0.050000 0.200000 P R 0.012000
LINE -0.050000 0.200000 0.050000 0.100000 P R 0.012000
LINE 0.050000 0.100000 0.100000 0.150000 P R 0.012000
ECHAR
```

Appendix 1
Metadata Content, Example, and Schema

Product model metadata can be supplied in a *metadata.xml* file.

See “misc/metadata.xml (Metadata)” on page 60.

Metadata Content

These node tags are supported by ODB++, and cannot be modified by a user. The tags are placed in the appropriate section of the *metadata.xml* file:

Header Metadata Section

These are the tags in the header section:

Table 1-1. Metadata Content - Header Section

Name	Description	Display Name	Type	Entity Level
buried_capacitor	Indicates whether this board contains buried capacitors.	Buried Capacitor	Boolean	product model
buried_resistor	Indicates whether this board contains buried resistors.	Buried Resistor	Boolean	product model
company_address	Company address.	Company Address	Text	product model

Table 1-1. Metadata Content - Header Section

Name	Description	Display Name	Type	Entity Level
company_name	Company name.	Company Name	Text	product model
counter_bore	Indicates whether the board contains counter bore holes.	Counter Bore	Boolean	product model
countersink	Indicates whether the board contains countersink holes.	Counter Sink	Boolean	product model
customer_control_number	Revision control number assigned by the customer.	Customer Control Number	Text	product model
designer_email	Email address of designer.	Designer Email	Text	product model
designer_name	Name of designer.	Designer Name	Text	product model
designer_phone_number	Phone number of the designer.	Phone Number	Text	product model
direct_connections_present	Indicates that direct connections to plans are known to exist.	Direct Connections Present	Boolean	step
edge_connectors	Indicates whether this board contains edge connectors.	Edge Connectors	Boolean	product model
layer_count	Number of copper layers in the board/panel.	Layer Count	Integer	product model
layout_name	Layout name.	Layout Name	Text	product model
part_x_size	Part size in the X direction.	Part X Size	Float (Distance)	step
part_y_size	Part size in the Y direction.	Part Y Size	Float (Distance)	step
pcb_part_number	Part number assigned to the PCB.	PCB Part Number	Text	product model
rohs_compliant	Indicates whether the process to produce the board or panel is ROHS compliant	RoHS Compliant	Boolean	product model

Table 1-1. Metadata Content - Header Section

Name	Description	Display Name	Type	Entity Level
ruling_ipc_spec	(Class 1, Class 2, Class 3, ...) IPC classification	Ruling IPC Specification (Class 1, 2, 3, etc.)	Option	product model
stacked_vias_ present	(yes/no) Indicates whether stacked vias are present.	Stacked Vias Present	Boolean	step
revision	(0-64 characters) Revision of this part.	Revision	Text	product model

Requirements Metadata Section

These are the tags in the requirements section:

Table 1-2. Metadata Content - Requirements

Name	Description	Display Name	Type	Entity Level
additional_copper_ lyr_require	Other requirements.	Additional Copper Layer Requirements	Text	step
additional_ requirements	Other requirements.	Additional Requirements	Text	product model
board_outline_ tolerance_minus	Minus tolerance for board outline.	Board Outline Tolerance Minus	Float (Distance)	product model
board_outline_ tolerance_plus	Plus tolerance for board outline.	Board Outline Tolerance Plus	Float (Distance)	product model
board_thickness	Thickness of board.	Board Thickness	Float (Distance)	product model
board_thickness_ tol_minus	Minus tolerance for board thickness.	Board Thickness Tol Minus	Float (Distance)	product model
board_thickness_ tol_plus	Plus tolerance for board thickness.	Board Thickness Tol Plus	Float (Distance)	product model

Table 1-2. Metadata Content - Requirements

Name	Description	Display Name	Type	Entity Level
board_thickness_ type	(on foil, laminate, plated copper, mask on plated copper) Defines how the board thickness is measured/specified.	Board Thickness Type	Text With Options	product model
bottom_legend_ color	(white, yellow) Color of the silkscreen or legend on the bottom side of the design.	Bottom Legend Color	Text With Options	product model
bottom_ soldermask_color	(green, red, blue) Color of the soldermask on the bottom side of the design.	Bottom Soldermask Color	Text With Options	product model
flammability_ rating_standard	(UL 94, UL 94V-O) Materials used in the fabrication of a printed circuit board design should have one of these flammability ratings.	Flammability Rating Standard	Option	product model
general_pcb_ standard	General PCB standard.	General PCB Standard	Text	product model
glass_transition_ temperature_tg	Degree fahrenheit / Celsius	Glass Transition Temperature (Tg)	Float (Temperature)	product model
gold_plating_ thickness_max	Required Max gold plating thickness.	Gold Plating Thickness Max	Float (Distance)	step
gold_plating_ thickness_min	Required minimum gold plating thickness.	Gold Plating Thickness Min	Float (Distance)	step
hole_position_ minus_tolerance	Minus tolerance for hole positions.	Hole Position Min Tolerance	Float (Distance)	step
hole_position_plus_ tolerance	Plus tolerance for hole positions.	Hole Position Plus Tolerance	Float (Distance)	step

Table 1-2. Metadata Content - Requirements

Name	Description	Display Name	Type	Entity Level
identify_ soldermask_define_ pads_that_need_to_ remain_untouched	Defines whether the open area of soldermask defining a pad should remain unedited.	Identify Soldermask Defined Pads to Remain Untouched	Boolean	step
legend_clearance_ from_component_ pads	Minimum spacing between legend and edge of copper component pads.	Legend Clearance from Component Pads	Float (Distance)	step
legend_clearance_ from_soldermask_ clearance	Minimum spacing between legend and edge of soldermask clearances.	Legend Clearance from Soldermask Clearance	Float (Distance)	step
legend_clearance_ from_test_points	Minimum spacing between legend and edge of copper test pads.	Legend Clearance from Test Points	Float (Distance)	step
legend_sides	(none, top, bottom, both) Sides of the board on which legend printing must be applied.	Legend Sides	Option	product model
lyr_to_lyr_reg_ max_tolerance	Maximum tolerance for layer to layer registration of copper layers.	Layer to Layer Registration Max Tolerance	Float (Distance)	step
max_bow_and_ twist_of_the_pcb	Maximum bow and twist requirement for this part.	Max Bow and Twist of PCB	Float (Distance)	step
max_operating_ temperature	Maximum operating temperature.	Max Operating Temperature	Float (Temperature)	product model
min_ar_blind_via	Minimum AR requirement for blind via holes.	Min AR Blind Via	Float (Distance)	step
min_ar_buried_via	Minimum AR requirement for buried via holes.	Min AR Buried Via	Float (Distance)	step
min_ar_pressfit_pth	Minimum AR requirement for Pressfit PTH.	Min AR Pressfit PTH	Float (Distance)	step

Table 1-2. Metadata Content - Requirements

Name	Description	Display Name	Type	Entity Level
min_ar_pth	Minimum AR requirement for PTH.	Min AR PTH	Float (Distance)	step
min_ar_via	Minimum AR requirement for Via holes.	Min AR Via	Float (Distance)	step
min_copper_to_board_edge	Minimum spacing between copper and board edge.	Min Copper to Board Edge	Float (Distance)	step
min_hole_to_edge	Minimum hole to board edge spacing.	Min Hole To Edge	Float (Distance)	step
min_hole_to_hole	Minimum hole to hole spacing.	Min Hole To Hole	Float (Distance)	step
min_soldermask_clearance_for_bga_pads	Minimum soldermask clearance for BGA pads.	Min Soldermask Clearance for BGA Pads	Float (Distance)	step
min_soldermask_clearance_for_npth	Minimum soldermask clearance for NPTH drills.	Min Soldermask Clearance for NPTH	Float (Distance)	step
min_soldermask_clearance_for_pth_pads	Minimum soldermask clearance for plated TH pads.	Min Soldermask Clearance for PTH Pads	Float (Distance)	step
min_soldermask_clearance_for_smd_pads	Minimum soldermask clearance for SMD pads.	Min Soldermask Clearance for SMD Pads	Float (Distance)	step
pcb_acceptability_standard	Standard for tolerances in the board.	PCB Acceptability Standard	Text	product model
peelable_mask_side	(none, top, bottom, both) Indicates whether peelable mask needs to be applied on this board, and if so, on which sides.	Peelable Mask Side	Option	product model
plated_edge	Indicates whether this job contains plated edges.	Plated Edge	Boolean	product model
plated_slots	Indicates whether this job contains plated slots.	Plated Slots	Boolean	product model

Table 1-2. Metadata Content - Requirements

Name	Description	Display Name	Type	Entity Level
qualification_ performance_std	Qualification and performance standard (free text).	Qualification and Performance Standard	Text	product model
soldermask_sides	(none, top, bottom, both) Indicates on which sides of the board soldermask must be applied.	Soldermask Sides	Option	product model
thermal_stress_ requirement	Thermal stress requirements for the board.	Thermal Stress Requirement	Text	product model
tolerance_standard	Standard for tolerances in the board.	Tolerance Standard	Text	product model
top_legend_color	(white, yellow) Color of the silkscreen or legend on the top side of the design.	Top Legend Color	Text With Options	product model
top_soldermask_ color	(none, green, white, yellow, black, blue, red) Color of the soldermask on the top side of the design.	Top Soldermask Color	Text With Options	product model

Manufacturing Metadata Section

These are the tags in the manufacturing section:

Table 1-3. Metadata Content - Manufacturing

Name	Description	Display Name	Type	Entity Level
allowed_remove_ nfp	Indicates whether it is allowed to remove non-functional pads.	Allowed To Remove Non-functional Pads	Boolean	step
barrel_cu_ thickness_blind_via	Required copper thickness in the barrel of blind via holes.	Barrel Cu Thickness Blind Via	Float (Distance)	step
barrel_cu_ thickness_buried_ via	Required copper thickness in the barrel of buried via holes.	Barrel Cu Thickness Buried Via	Float (Distance)	step

Table 1-3. Metadata Content - Manufacturing

Name	Description	Display Name	Type	Entity Level
barrel_cu_ thickness_pressfit_ pth	Required copper thickness in the barrel of Pressfit holes.	Barrel Cu Thickness Pressfit PTH	Float (Distance)	step
barrel_cu_ thickness_pth	Required copper thickness in the barrel of PTH.	Barrel Cu Thickness PTH	Float (Distance)	step
barrel_cu_ thickness_via	Required copper thickness in the barrel of Via holes.	Barrel Cu Thickness Via	Float (Distance)	step
circuit_copper_ repairs_allowed	(yes, only after approval, no) Indicates whether copper repairs, in order to resolve spacing problems, are allowed on the board.	PCB Circuit Copper Repairs Allowed	Text With Options	step
contour_processing_ type	(rout, V-score, laser cutting) Indicates how the board can be cut from the panel.	Contour Processing Type	Text With Options	step
copper_area_percent	Percentage of copper.	Copper Area Percent	Float	step
date_stamp_format	(WWYY, YYWW, YYMM, MMYY, LLMMYY, WWYY-LLL, YYWW-LLL, WW-YY-LLL, WWYYLL) Format of the date stamp. W = Week, Y = Year, M = Month, D = Day, L = Lot	Date Stamp Format	TextWith Options	step
date_stamp_ required	Indicates whether a date stamp should be added on the board.	Date Stamp Required	Boolean	step
electrical_test_ required	Indicates whether Electrical testing of this board is required.	Electrical Test Required	Boolean	step
fiducial_ soldermask_ clearance	Indicates the minimum soldermask clearance for fiducial pads.	Fiducial Soldermask Clearance	Float (Distance)	step

Table 1-3. Metadata Content - Manufacturing

Name	Description	Display Name	Type	Entity Level
gold_plating_ defined_by	(Gold Mask, Soldermask, NA) Type of gold plating used.	Gold Plating Defined by	Option	step
high_voltage_board	Indicates whether this board needs to be tested for high voltage.	High Voltage Board	Boolean	step
hole_breakout_ allowed	Indicates whether a hole can break out of the copper pad.	Hole Breakout Allowed	Boolean	step
legend_clipping_ allowed	Indicates whether clipping of the legend to clear exposed copper from being covered by legend is allowed.	Legend Clipping Allowed	Boolean	step
lot	Production lot number for a board.	Production Lot Number	Text	step
mfg_id_stamp	Indicates whether a manufacturer identification stamp should be added on the board.	Manufacturer Identification Stamp	Boolean	step
min_conductor_ width_inner	Minimum conductor width on inner layers.	Min Conductor Width Inner	Float (Size)	step
min_conductor_ width_outer	Minimum conductor width on outer layers.	Min Conductor Width Outer	Float (Size)	step
min_copper_ spacing_inner	Minimum copper spacing on inner layers.	Min Copper Spacing Inner	Float (Distance)	step
min_copper_ spacing_outer	Minimum copper spacing on outer layers.	Min Copper Spacing Outer	Float (Distance)	step
number_of_drill_ holes	Number of drill holes.	Number of Drill Holes	Integer	step
number_of_thru_ pads	Number of thru pads.	Number of Thru Pads	Integer	step
number_of_tooling_ holes	Number of tooling holes.	Number of Tooling Holes	Integer	step

Table 1-3. Metadata Content - Manufacturing

Name	Description	Display Name	Type	Entity Level
number_of_unique_drill_sizes	Number of unique drill sizes.	Number of Unique Drill Sizes	Integer	step
number_of_vias	Number of vias.	Number of Vias	Integer	step
part_number_stamp	Indicates whether a Part number stamp should be added on the board.	Part Number Stamp	Boolean	step
plug_vias	Indicates whether vias should be plugged.	Plug Vias	Boolean	step
print_et_test_stamp_on_board	Indicates whether it is required to print an ET stamp on the board.	Print ET Test Stamp on Board	Boolean	step
smallest_drill_size	Smallest drill size.	Smallest Drill Size	Float (Size)	step
smallest_trace_width	Smallest trace width.	Smallest Trace Width	Float (Size)	step
smallest_via_diameter	Smallest via diameter.	Smallest Via Diameter	Float (Size)	step
surface_finish_contact_fingers	(HASL, Lead-Free HASL, OSP (Intek), ENIG, Deep Gold, Wirebondable Gold, Bare Cu, Immersion Silver, White Tim, Other) Surface finish for contact fingers.	Surface Finish Contact Fingers	Text With Options	step
surface_finish_pads	(HASL, Lead-Free HASL, OSP (Intek), ENIG, Deep Gold, Wirebondable Gold, Bare Cu, Immersion Silver, White Tim, Other) Surface finish for exposed pads.	Surface Finish Pads	Text With Options	step
teardrop_allowed_on_inner	Indicates whether it is allowed to add teardrops on inner copper layers.	Teardrop Addition Allowed on Inner	Boolean	step

Table 1-3. Metadata Content - Manufacturing

Name	Description	Display Name	Type	Entity Level
teardrop_allowed_on_outer	Indicates whether it is allowed to add teardrops on outer copper layers.	Teardrop Addition Allowed on Outer	Boolean	step
thieving_restrictions_on_inner	Restrictions for thieving patterns on inner layers (free text).	Thieving Restrictions on Inner	Text	step
thieving_restrictions_on_outer	Restrictions for thieving patterns on inner layers (free text).	Thieving Restrictions on Outer	Text	step
ul_code_stamp	Indicates whether a UL Code stamp should be added on the board.	UL Code Stamp	Boolean	step
unique_serial_nr_stamp	Indicates whether a unique serial number stamp should be added on the board.	Unique Serial nr Stamp	Boolean	step
via_tenting	Indicates whether there are tented vias.	Via Tenting	Boolean	step
via_treatments	(Via Fill - Conductive, Via Fill - Non-conductive, Via Fill - Copper Plated, Via Fill - Soldermask Hole Filled, Via Capped, Clearance - Standard Soldermask Clearance, Dry Film Via Tenting, Soldermask Defined - Soldermask over Via) The via treatment that should be applied.	Via Treatments	Text With Options	step

Assembly Metadata Section

These are the tags in the requirements section:

Table 1-4. Metadata Content - Assembly

Name	Description	Display Name	Data Type	Entity Level
cad_package_ analysis	CAD package analysis: Number of unique cells per side and total with associated pin count and pin pitch.	CAD Package Analysis	Integer	step
complete_test_ required	Indicates whether a test including barrel testing is required.	Complete Test Required	Boolean	step
component_count	Component count on board or panel.	Components per Step	Integer	step
number_of_ accessible_nets	Number of accessible nets.	Number of Accessible Nets		step
number_of_fiducials	Number of fiducials.	Number of Fiducials	Integer	step
number_of_multi_ pin_nets	Number of multi-pin nets.	Number of Multi-pin Nets	Integer	step
number_of_nets	Number of nets.	Number of Nets	Integer	step
number_of_nets_ with_probes	Number of nets with probes.	Number of Nets with Probes	Integer	step
number_of_single_ pin_nets	Number of single pin nets.	Number of Single Pin Nets	Integer	step
number_of_test_ points	Number of test points (net ends) on the board or panel.	Number of Test Points	Integer	step
number_of_ un-connected_nets	Number of unconnected nets.	Number of Unconnected Nets	Integer	step
pressfit_technology	Indicates whether pressfit technology should be used on this board.	Pressfit Technology	Boolean	step

Table 1-4. Metadata Content - Assembly

Name	Description	Display Name	Data Type	Entity Level
smallest_pin_pitch	Smallest pin pitch.	Smallest Pin Pitch	Float (Distance)	step
smd_technology	(single sided, double sided, none) The sides in the design that have SMD technology.	SMD Technology	Option	step
total_number_of_ smd_pads	Total number of SMD pads.	Total Number of SMD Pads	Integer	step

Final Metadata Section

These are the tags in the final section:

Table 1-5. Metadata Content - Final

Name	Description	Display Name	Data Type	Entity Type
qual_coupon_ crosssection_rpt	Type of quality coupon cross section report (free text).	Quality Coupon Cross Section Report	Boolean	step
ship_as	(Single board, assembly panel) How the product is delivered.	Ship as	Option	step
solder_resist_ adhesion_tst_rpt	Type of solder resist adhesion test report (free text).	Solder Resist Adhesion Test Report	Boolean	step
solderability_test_ report	Type of solderability test report (free text).	Solderability Test Report	Boolean	step

Metadata Example

This is an example of a *metedata.xml* file:

```
<metadata>

<Header description="General Information" >

<node display="Buried Resistor" value="no" name="buried_resistor" />
<node display="Buried Capacitor" value="no" name="buried_capacitor" />
```

```
<node display="Company Address" value="Smallville" name="company_address" />
<node display="Company Name" value="OurCo" name="company_name" />
<node display="Counter Bore" value="no" name="counter_bore" />
<node display="Counter Sink" value="yes" name="countersink" />

...

</Header>

<Requirements description="Board Requirements" >

<node display="Additional Requirements" value="" name="additional_requirements" />
<node display="Board Outline Tolerance Plus" units="MM" value="" name="board_outline_
tolerance_plus" />
<node display="Board Outline Tolerance Minus" units="MM" value="" name="board_
outline_tolerance_minus" />
<node display="Board Thickness" units="MM" value="1.71603" name="board_thickness" />
<node display="Board Thickness Tol Plus" units="MM" value="" name="board_thickness_
tol_plus" />
<node display="Board Thickness Tol Minus" units="MM" value="" name="board_thickness_
tol_minus" />

...

</Requirements>

<Steps description="Steps" >

<Step name="cad" >

<Header description="General Information" >

<node display="Direct Connections Present" value="" name="direct_connections_present"
/>
<node display="Part X Size" units="MM" value="235.6" name="part_x_size" />
<node display="Part Y Size" units="MM" value="447.68" name="part_y_size" />
<node display="Stacked Vias Present" value="" name="stacked_vias_present" />

</Header>

<Requirements description="Board Requirements" >

<node display="Additional Copper Layer Requirements" value="" name="additional_
copper_lyr_require" />
<node display="Gold Plating Thickness Min" units="MM" value="" name="gold_plating_
thickness_min" />
<node display="Gold Plating Thickness Max" units="MM" value="" name="gold_plating_
thickness_max" />
<node display="Hole Position Plus Tolerance" units="MM" value="" name="hole_position_
plus_tolerance" />
<node display="Hole Position Minus Tolerance" units="MM" value="" name="hole_
position_minus_tolerance" />
<node display="Idenitfy Soldermask Define Pads that Need to Remain Untouched" value=""
name="identify_soldermask_define_pads_that_need_to_remain_untouched" />

...

</Requirements>

<Manufacturing description="Manufacturing Process" >
```

```
<node display="Allowed to Remove non-functional Pads" value="" name="allowed_remove_
nfp" />
<node display="Barrel Cu Thickness Blind Via" units="MM" value="" name="barrel_cu_
thickness_blind_via" />
<node display="Barrel Cu Thickness Buried Via" units="MM" value="" name="barrel_cu_
thickness_buried_via" />
<node display="Barrel Cu Thickness Pressfit PTH" units="MM" value="" name="barrel_cu_
thickness_pressfit_pth" />
<node display="Barrel Cu Thickness PTH" units="MM" value="" name="barrel_cu_
thickness_pth" />
<node display="Barrel Cu Thickness Via" units="MM" value="" name="barrel_cu_
thickness_via" />

...

</Manufacturing>

<Assembly description="Assembly" >

<node display="Cad Package Analysis" value="53" name="cad_package_analysis" >

<classify_by display="CAD Packages on Top Layer" value="36" name="Top" >

  <classify_by display="CAD Package" value="13" name="EB020A0" >
    <classify_by display="Pin Count" value="20" name="pin_count" />
    <classify_by display="Pitch" units="MM" value="1.27" name="pitch" />
  </classify_by>

  <classify_by display="CAD Package" value="1" name="EG064A3" >
    <classify_by display="Pin Count" value="65" name="pin_count" />
    <classify_by display="Pitch" units="MM" value="0.5" name="pitch" />
  </classify_by>

...

</classify_by>

<classify_by display="CAD Packages on Bottom Layer" value="17" name="Bottom" >

  <classify_by display="CAD Package" value="2" name="EB048A0" >
    <classify_by display="Pin Count" value="50" name="pin_count" />
    <classify_by display="Pitch" units="MM" value="0.5" name="pitch" />
  </classify_by>

  <classify_by display="CAD Package" value="11" name="EUF03S0" >
    <classify_by display="Pin Count" value="3" name="pin_count" />
    <classify_by display="Pitch" units="MM" value="1.8999975" name="pitch" />
  </classify_by>

...

</classify_by>

</node>

<node display="Complete Test Required" value="" name="complete_test_required" />

<node display="Components per Step" value="2436" name="component_count" >
  <classify_by display="Components in Top Layer" value="677" name="Top" />
  <classify_by display="Components in Bottom Layer" value="1759" name="Bottom" />
</node>
```

```
<node display="Number of Fiducials" value="16" name="number_of_fiducials" >
  <classify_by display="Fiducials in Top Layer" value="8" name="Top" />
  <classify_by display="Fiducials in Bottom Layer" value="8" name="Bottom" />
</node>

<node display="Number of Nets" value="1870" name="number_of_nets" />

<node display="Number of Accessible Nets" value="" name="number_of_accessible_nets"
/>

<node display="Number of Multi-pin Nets" value="2" name="number_of_multi_pin_nets" >
  <classify_by display="Nets with Multiple Test Points in Top Layer" value="0"
name="Top" />
  <classify_by display="Nets with Multiple Test Points in Bottom Layer" value="2"
name="Bottom" />
</node>

<node display="Number of nets with probes" value="" name="number_of_nets_with_probes"
/>

<node display="Number of single pin nets" value="929" name="number_of_single_pin_
nets" >
  <classify_by display="Nets with a Single Test Point in Top Layer" value="0"
name="Top" />
  <classify_by display="Nets with a Single Test Point in Bottom Layer" value="929"
name="Bottom" />
</node>

<node display="Number of unconnected nets" value="939" name="number_of_un-connected_
nets" >
  <classify_by display="Nets with No Test Points in Top Layer" value="937" name="Top"
/>
  <classify_by display="Nets with No Test Points in Bottom Layer" value="2"
name="Bottom" />
</node>

<node display="Number of Test Points" value="933" name="number_of_test_points" >
  <classify_by display="Test Points in Top Layer" value="0" name="Top" />
  <classify_by display="Test Points in Bottom Layer" value="933" name="Bottom" />
</node>

<node display="Pressfit Technology" value="N/A" name="pressfit_technology" />

<node display="Smallest Pin Pitch" units="MM" value="N/A" name="smallest_pin_pitch"
/>

<node display="SMD Technology" value="None" name="smd_technology" />

<node display="Total Number of SMD Pads" value="0" name="total_number_of_smd_pads" >
  <classify_by display="SMD Pads in Top Layer" value="0" name="Top" />
  <classify_by display="SMD Pads in Bottom Layer" value="0" name="Bottom" />
</node>

</Assembly>

<Final description="Final QC" >

<node display="Quality Coupon Cross Section Report" value="" name="qual_coupon_
crosssection_rpt" />
<node display="Ship as" value="Single board" name="ship_as" />
<node display="Solder Resist Adhesion Test Report" value="" name="solder_resist_adhesion_tst_
```

```
rpt" />
<node display="Solderability Test Report" value="" name="solderability_test_report" />

</Final>
</Step>
</Steps>
</metadata>
```

Metadata Schema

This is the metadata schema.

```
<?xml version="1.0" encoding="utf-8" ?>
<metadataSchema description="">

  <metadataSection internalName="Header" description="General
Information">

    <metadataAttr internalName="buried_resistor" displayName="Buried
Resistor" entityLevel="JOB" type="Boolean" callback="ODBAttr"
description="Defines whether board contains buried resistors"/>
    <metadataAttr internalName="buried_capacitor" displayName="Buried
Capacitor" entityLevel="JOB" type="Boolean" callback="ODBAttr"
description="Defines whether board contains buried capacitors"/>
    <metadataAttr internalName="company_address" displayName="Company
Address" entityLevel="JOB" type="Text" callback="ODBAttr"
description="Defines company address"/>
    <metadataAttr internalName="company_name" displayName="Company Name"
entityLevel="JOB" type="Text" callback="ODBAttr" description="Defines
designer company name"/>
    <metadataAttr internalName="counter_bore" displayName="Counter Bore"
entityLevel="JOB" type="Boolean" callback="ODBAttr" description="Defines
whether job requires counter bore"/>
    <metadataAttr internalName="countersink" displayName="Counter Sink"
entityLevel="JOB" type="Boolean" callback="ODBAttr" description="Defines
whether job requires counter sink drilling"/>
    <metadataAttr internalName="customer_control_number"
displayName="Customer Control Number" entityLevel="JOB" type="Text"
callback="ODBAttr" description="Defines revision control number if
avaliable"/>
    <metadataAttr internalName="designer_email" displayName="Designer
Email" entityLevel="JOB" type="Text" callback="ODBAttr"
description="Defines designer email address"/>
   <metadataAttr internalName="designer_name" displayName="Designer Name"
entityLevel="JOB" type="Text" callback="ODBAttr" description="Defines
designer name"/>
    <metadataAttr internalName="designer_phone_number"
displayName="Designer Phone Number" entityLevel="JOB" type="Text"
callback="ODBAttr" description="Defines designer phone number"/>
    <metadataAttr internalName="direct_connections_present"
displayName="Direct Connections Present" entityLevel="STEP"
type="Boolean" callback="ODBAttr" description="Indicates that direct
connections to planes are known to exist."/>
    <metadataAttr internalName="edge_connectors" displayName="Edge
Connectors" entityLevel="JOB" type="Boolean" callback="ODBAttr"
description="Defines whether board contains edge connectors"/>
```

```
    <metadataAttr internalName="layer_count" displayName="Layer Count"
entityLevel="JOB" type="Integer" callback="Function" description="Defines
number of copper layers in board"/>
    <metadataAttr internalName="layout_name" displayName="Layout Name"
entityLevel="JOB" type="Text" callback="ODBAttr" description="Defines
layout name of design"/>
    <metadataAttr internalName="part_x_size" displayName="Part X Size"
entityLevel="STEP" type="float" unitsType="Distance" unitsSize="INCH_MM"
callback="Function" description="Defines part X dimension"/>
    <metadataAttr internalName="part_y_size" displayName="Part Y Size"
entityLevel="STEP" type="float" unitsType="Distance" unitsSize="INCH_MM"
callback="Function" description="Defines part Y dimension"/>
    <metadataAttr internalName="pcb_part_number" displayName="PCB Part
Number" entityLevel="JOB" type="Text" callback="ODBAttr"
description="Defines part number of job"/>
    <metadataAttr internalName="revision" displayName="Revision"
entityLevel="JOB" type="Text" callback="ODBAttr" description="Defines
part revision"/>
    <metadataAttr internalName="rohs_compliant" displayName="RoHS
Compliant" entityLevel="JOB" type="Boolean" callback="ODBAttr"
description="Defines whether board needs RoHS compliance"/>
    <metadataAttr internalName="ruling_ipc_spec" displayName="Ruling IPC
Spec (Class 1, 2, 3 etc.)" entityLevel="JOB" type="Option"
callback="ODBAttr" description="IPC Classification" options="Class 1,
Class 2, Class 3, ?"/>
   <metadataAttr internalName="stacked_vias_present" displayName="Stacked
Vias Present" entityLevel="STEP" type="Boolean" callback="ODBAttr"
description="Indicates if stacked vias are present"/>
  </metadataSection>

  <metadataSection internalName="Requirements" description="Board
Requirements">

    <metadataAttr internalName="additional_requirements"
displayName="Additional Requirements" entityLevel="JOB" type="Text"
callback="ODBAttr" description="Free text - up to 20000 characters"/>
    <metadataAttr internalName="additional_copper_lyr_require"
displayName="Additional Copper Layer Requirements" entityLevel="STEP"
type="Text" callback="ODBAttr" description="Other Requirements"/>
    <metadataAttr internalName="board_outline_tolerance_plus"
displayName="Board Outline Tolerance Plus" entityLevel="JOB" type="float"
unitsType="Distance" unitsSize="MIL_MICRON" callback="ODBAttr"
description="Defines +tolerance of board size"/>
    <metadataAttr internalName="board_outline_tolerance_minus"
displayName="Board Outline Tolerance Minus" entityLevel="JOB" type="float"
unitsType="Distance" unitsSize="MIL_MICRON" callback="ODBAttr"
description="Defines -tolerance of board size"/>
    <metadataAttr internalName="board_thickness" displayName="Board
Thickness" entityLevel="JOB" type="float" unitsType="Distance"
unitsSize="MIL_MICRON" callback="Function" description="Defines board
thickness"/>
    <metadataAttr internalName="board_thickness_tol_plus"
displayName="Board Thickness Tol Plus" entityLevel="JOB" type="float"
unitsType="Distance" unitsSize="MIL_MICRON" callback="ODBAttr"
description="Defines +tolerance of board thickness"/>
```

```
    <metadataAttr internalName="board_thickness_tol_minus"
displayName="Board Thickness Tol Minus" entityLevel="JOB" type="float"
unitsType="Distance" unitsSize="MIL_MICRON" callback="ODBAttr"
description="Defines -tolerance of board thickness"/>
    <metadataAttr internalName="board_thickness_type" displayName="Board
Thickness Type" entityLevel="JOB" type="TextWithOptions"
callback="ODBAttr" description="Defines how board thickness is
measured/specified" options="over foil, over laminate, over plated copper,
over mask on plated copper"/>
    <metadataAttr internalName="bottom_legend_color" displayName="Bottom
Legend Color" entityLevel="JOB" type="TextWithOptions" callback="ODBAttr"
description="Defines color of bottom legend" options="none , green , white
, yellow , black , blue , red"/>
    <metadataAttr internalName="bottom_soldermask_color"
displayName="Bottom Soldermask Color" entityLevel="JOB"
type="TextWithOptions" callback="ODBAttr" description="Defines color of
bottom soldermask" options="none , green , white , yellow , black , blue ,
red"/>
    <metadataAttr internalName="flammability_rating_standard"
displayName="Flammability Rating Standard" entityLevel="JOB" type="Text"
callback="ODBAttr" description="Defines Flammability Rating Standard (free
text)"/>
   <metadataAttr internalName="general_pcb_standard" displayName="General
PCB Standard" entityLevel="JOB" type="Text" callback="ODBAttr"
description="Defines general PCB Standard (free text)"/>
    <metadataAttr internalName="glass_transition_temperature_tg"
displayName="Glass Transition Temperature (Tg)" entityLevel="JOB"
type="Float" unitsType="Temperature" callback="ODBAttr"
description="Degree Fahrenheit/Celcius"/>
    <metadataAttr internalName="gold_plating_thickness_min"
displayName="Gold Plating Thickness Min" entityLevel="STEP" type="float"
unitsType="Distance" unitsSize="MIL_MICRON" callback="ODBAttr"
description="Defines minimum gold plating thickness"/>
    <metadataAttr internalName="gold_plating_thickness_max"
displayName="Gold Plating Thickness Max" entityLevel="STEP" type="float"
unitsType="Distance" unitsSize="MIL_MICRON" callback="ODBAttr"
description="Defines maximum gold plating thickness"/>
    <metadataAttr internalName="hole_position_plus_tolerance"
displayName="Hole Position Plus Tolerance" entityLevel="STEP" type="float"
unitsType="Distance" unitsSize="MIL_MICRON" callback="ODBAttr"
description="Defines +tolerance for hole positions"/>
    <metadataAttr internalName="hole_position_minus_tolerance"
displayName="Hole Position Minus Tolerance" entityLevel="STEP"
type="float" unitsType="Distance" unitsSize="MIL_MICRON"
callback="ODBAttr" description="Defines -tolerance for hole positions"/>
   <metadataAttr internalName="identify_soldermask_define_pads_that_need_
to_remain_untouched" displayName="Idenitfy Soldermask Define Pads that
Need to Remain Untouched" entityLevel="STEP" type="Boolean"
callback="ODBAttr" description="Attribute indicates soldermask pads that
are to remain untouched"/>
    <metadataAttr internalName="legend_clearance_from_component_pads"
displayName="Legend  Clearance from Component Pads" entityLevel="STEP"
type="float" unitsType="Distance" unitsSize="MIL_MICRON"
callback="Function" description="Defines minimum spacing between legend
and edge of copper component pads"/>
```

```
    <metadataAttr internalName="legend_clearance_from_soldermask_
clearance" displayName="Legend Clearance from Soldermask Clearance"
entityLevel="STEP" type="float" unitsType="Distance" unitsSize="MIL_
MICRON" callback="ResultAttr" description="Defines minimum spacing between
legend and edge of soldermask clearances" arguments="min, sm"/>
    <metadataAttr internalName="legend_clearance_from_test_points"
displayName="Legend Clearance from Test Points" entityLevel="STEP"
type="float" unitsType="Distance" unitsSize="MIL_MICRON"
callback="Function" description="Defines minimum spacing between legend
and edge of copper test pads"/>
    <metadataAttr internalName="legend_sides" displayName="Legend Sides"
entityLevel="JOB" type="Option" callback="ODBAttr" description="Defines
sides on which legend printing is to be applied"
options="none,Top,Bottom,Both"/>
    <metadataAttr internalName="lyr_to_lyr_reg_max_tolerance"
displayName="Layer to Layer Registration Max Tolerance" entityLevel="STEP"
type="float" unitsType="Distance" unitsSize="MIL_MICRON"
callback="ODBAttr" description="Defines maximum tolerance for layer to
layer registration of copper layers"/>
    <metadataAttr internalName="max_bow_and_twist_of_the_pcb"
displayName="Max Bow and Twist of the PCB" entityLevel="STEP" type="float"
unitsType="Distance" unitsSize="MIL_MICRON" callback="ODBAttr"
description="Defines maximim bow&amp;amp;twist requirement for this part"/>
    <metadataAttr internalName="max_operating_temperature"
displayName="Max Operating Temperature" entityLevel="JOB" type="Float"
unitsType="Temperature" callback="ODBAttr" description="Degree
Fahrenheit/Celcius"/>
    <metadataAttr internalName="min_ar_blind_via" displayName="Min AR
Blind Via" entityLevel="STEP" type="float" unitsType="Distance"
unitsSize="MIL_MICRON" callback="ResultAttr" description="Defines the AR
requirement for blind via holes" arguments="min, blind_via_ar"/>
    <metadataAttr internalName="min_ar_buried_via" displayName="Min AR
Buried Via" entityLevel="STEP" type="float" unitsType="Distance"
unitsSize="MIL_MICRON" callback="ResultAttr" description="Defines min AR
requirement for buried via holes" arguments="min, buried_via_ar"/>
    <metadataAttr internalName="min_ar_pressfit_pth" displayName="Min AR
Pressfit PTH" entityLevel="STEP" type="float" unitsType="Distance"
unitsSize="MIL_MICRON" callback="Function" description="Defines min AR
requirement for Pressfit PTH"/>
    <metadataAttr internalName="min_ar_pth" displayName="Min AR PTH"
entityLevel="STEP" type="float" unitsType="Distance" unitsSize="MIL_
MICRON" callback="ResultAttr" description="Defines min AR requirement for
PTH" arguments="min, pth_ar"/>
    <metadataAttr internalName="min_ar_via" displayName="Min AR Via"
entityLevel="STEP" type="float" unitsType="Distance" unitsSize="MIL_
MICRON" callback="ResultAttr" description="Defines min AR requirement for
via holes" arguments="min, via_ar"/>
   <metadataAttr internalName="min_copper_to_board_edge" displayName="Min
Copper to Board Edge" entityLevel="STEP" type="float" unitsType="Distance"
unitsSize="MIL_MICRON" callback="ResultAttr" description="Defines minimum
spacing between copper and board edge" arguments="min, r2c"/>
   <metadataAttr internalName="min_hole_to_edge" displayName="Min Hole to
Edge" entityLevel="STEP" type="float" unitsType="Distance"
unitsSize="MIL_MICRON" callback="Function" description="Defines minimum
hole to board edge spacing"/>
```

```
   <metadataAttr internalName="min_hole_to_hole" displayName="Min Hole to
Hole" entityLevel="STEP" type="float" unitsType="Distance"
unitsSize="MIL_MICRON" callback="ResultAttr" description="Defines minimum
hole to hole spacing" arguments="min, closeh"/>
    <metadataAttr internalName="min_soldermask_clearance_for_bga_pads"
displayName="Min Soldermask Clearance for BGA Pads" entityLevel="STEP"
type="float" unitsType="Distance" unitsSize="MIL_MICRON"
callback="Function" description="Defines minimum soldermask clearance for
BGA pads"/>
    <metadataAttr internalName="min_soldermask_clearance_for_npth"
displayName="Min Soldermask Clearance for NPTH" entityLevel="STEP"
type="float" unitsType="Distance" unitsSize="MIL_MICRON"
callback="ResultAttr" description="Defines minimum soldermask clearance
for NPTH drills" arguments="min, ar_npth"/>
    <metadataAttr internalName="min_soldermask_clearance_for_pth_pads"
displayName="Min Soldermask Clearance for PTH Pads" entityLevel="STEP"
type="float" unitsType="Distance" unitsSize="MIL_MICRON"
callback="ResultAttr" description="Defines minimum soldermask clearance
for plated TH pads" arguments="min, ar_pth"/>
    <metadataAttr internalName="min_soldermask_clearance_for_smd_pads"
displayName="Min Soldermask Clearance for SMD Pads" entityLevel="STEP"
type="float" unitsType="Distance" unitsSize="MIL_MICRON"
callback="ResultAttr" description="Defines minimum soldermask clearance
for SMD pads" arguments="min, ar_smd"/>
    <metadataAttr internalName="pcb_acceptability_standard"
displayName="PCB Acceptability Standard" entityLevel="JOB" type="Text"
callback="ODBAttr" description="Defines PCB Acceptability Standard (free
text)"/>
   <metadataAttr internalName="peelable_mask_side" displayName="Peelable
Mask Side" entityLevel="JOB" type="Option" callback="ODBAttr"
description="Defines whether peelable mask is to be applied, and if yes,
on which sides" options="none,Top,Bottom,Both"/>
    <metadataAttr internalName="plated_edge" displayName="Plated Edge"
entityLevel="JOB" type="Boolean" callback="ODBAttr" description="Defines
whether plating on the board edge is required"/>
    <metadataAttr internalName="plated_slots" displayName="Plated Slots"
entityLevel="JOB" type="Boolean" callback="ODBAttr" description="Defines
whether job contains plated slots"/>
    <metadataAttr internalName="qualification_performance_std"
displayName="Qualification and Performance Standard" entityLevel="JOB"
type="Text" callback="ODBAttr" description="Defines Qualification and
Performance Standard (free text)"/>
    <metadataAttr internalName="soldermask_sides" displayName="Soldermask
Sides" entityLevel="JOB" type="Option" callback="ODBAttr"
description="Defines sides on which sides soldermask is to be applied"
options="None,Top,Bottom,Both"/>
    <metadataAttr internalName="thermal_stress_requirement"
displayName="Thermal Stress Requirement" entityLevel="JOB" type="Text"
callback="ODBAttr" description="Defines thermal stress requirements for
board (free text)"/>
   <metadataAttr internalName="tolerance_standard" displayName="Tolerance
Standard" entityLevel="JOB" type="Text" callback="ODBAttr"
description="Defines standard for tolerances in board (free text)"/>
    <metadataAttr internalName="top_legend_color" displayName="Top Legend
Color" entityLevel="JOB" type="TextWithOptions" callback="ODBAttr"
description="Defines color of top legend" options="none , green , white ,
yellow , black , blue , red"/>
```

```
    <metadataAttr internalName="top_soldermask_color" displayName="Top
Soldermask Color" entityLevel="JOB" type="TextWithOptions"
callback="ODBAttr" description="Defines color of top soldermask"
options="none , green , white , yellow , black , blue , red"/>
  </metadataSection>

  <metadataSection internalName="Manufacturing"
description="Manufacturing Process">

    <metadataAttr internalName="allowed_remove_nfp" displayName="Allowed
to Remove non-functional Pads" entityLevel="STEP" type="Boolean"
callback="ODBAttr" description="Defines whether removing non-functional
pads is allowed"/>
    <metadataAttr internalName="barrel_cu_thickness_blind_via"
displayName="Barrel Cu Thickness Blind Via" entityLevel="STEP"
type="float" unitsType="Distance" unitsSize="MIL_MICRON"
callback="ODBAttr" description="Defines required Cu thickness in barrel of
blind via holes"/>
    <metadataAttr internalName="barrel_cu_thickness_buried_via"
displayName="Barrel Cu Thickness Buried Via" entityLevel="STEP"
type="float" unitsType="Distance" unitsSize="MIL_MICRON"
callback="ODBAttr" description="Defines required Cu thickness in barrel of
buried via holes"/>
    <metadataAttr internalName="barrel_cu_thickness_pressfit_pth"
displayName="Barrel Cu Thickness Pressfit PTH" entityLevel="STEP"
type="float" unitsType="Distance" unitsSize="MIL_MICRON"
callback="ODBAttr" description="Defines required Cu thickness in barrel of
Pressfit holes"/>
    <metadataAttr internalName="barrel_cu_thickness_pth"
displayName="Barrel Cu Thickness PTH" entityLevel="STEP" type="float"
unitsType="Distance" unitsSize="MIL_MICRON" callback="ODBAttr"
description="Defines required Cu thickness in barrel of PTH"/>
    <metadataAttr internalName="barrel_cu_thickness_via"
displayName="Barrel Cu Thickness Via" entityLevel="STEP" type="float"
unitsType="Distance" unitsSize="MIL_MICRON" callback="ODBAttr"
description="Defines required Cu thickness in barrel of via holes"/>
    <metadataAttr internalName="circuit_copper_repairs_allowed"
displayName="PCB Circuit Copper Repairs Allowed" entityLevel="STEP"
type="TextWithOptions" callback="ODBAttr" description="Defines whether
copper repairs, to resolve spacing problems, are allowed on board"
options="Yes , Only after approval , No"/>
    <metadataAttr internalName="contour_processing_type"
displayName="Contour Processing Type" entityLevel="STEP"
type="TextWithOptions" callback="ODBAttr" description="Defines how to cut
board from panel" options="Route , V-score , Laser cutting"/>
    <metadataAttr internalName="copper_area_percent" displayName="Copper
Area Percent" entityLevel="STEP" type="float" callback="ODBAttr"
description="Percentage of copper"/>
   <metadataAttr internalName="date_stamp_format" displayName="Date Stamp
Format" entityLevel="STEP" type="text" callback="ODBAttr"
description="Defines format of date stamp"/>
    <metadataAttr internalName="date_stamp_required" displayName="Date
Stamp Required" entityLevel="STEP" type="Boolean" callback="ODBAttr"
description="Defines whether date stamp should be added on board"/>
    <metadataAttr internalName="electrical_test_required"
displayName="Electrical Test Required" entityLevel="STEP" type="Boolean"
callback="ODBAttr" description="Defines whether electrical testing of
board is required"/>
```

```
    <metadataAttr internalName="fiducial_soldermask_clearance"
displayName="Fiducial Soldermask Clearance" entityLevel="STEP"
type="float" unitsType="Distance" unitsSize="MIL_MICRON"
callback="Function" description="Defines the minimum soldermask clearance
for fiducial pads"/>
   <metadataAttr internalName="gold_plating_defined_by" displayName="Gold
Plating Defined by" entityLevel="STEP" type="Option" callback="ODBAttr"
description="Gold mask/Soldermask/NA" options="Gold mask, Solder mask,
N/A"/>
    <metadataAttr internalName="high_voltage_board" displayName="High
Voltage Board" entityLevel="STEP" type="Boolean" callback="ODBAttr"
description="Defines whether board needs to be tested for high voltage"/>
    <metadataAttr internalName="hole_breakout_allowed" displayName="Hole
Breakout Allowed" entityLevel="STEP" type="Boolean" callback="ODBAttr"
description="Defines whether hole can break out of copper pad"/>
    <metadataAttr internalName="legend_clipping_allowed"
displayName="Legend Clipping Allowed" entityLevel="STEP" type="Boolean"
callback="ODBAttr" description="Defines whether clipping of legend to
clear exposed copper from being covered by legend is allowed"/>
    <metadataAttr internalName="lot" displayName="Production Lot Number"
entityLevel="STEP" type="text" callback="ODBAttr" description="Defines
the production lot number for a board"/>
    <metadataAttr internalName="mfg_id_stamp" displayName="Manufacturer
Identification Stamp" entityLevel="STEP" type="Boolean"
callback="ODBAttr" description="Defines whether a manufacturer
identification stamp should be added on the board"/>
    <metadataAttr internalName="min_conductor_width_inner"
displayName="Min Conductor Width Inner" entityLevel="STEP" type="float"
unitsType="Distance" unitsSize="MIL_MICRON" callback="Function"
description="Defines minimum conductor width on inner layers"/>
    <metadataAttr internalName="min_conductor_width_outer"
displayName="Min Conductor Width Outer" entityLevel="STEP" type="float"
unitsType="Distance" unitsSize="MIL_MICRON" callback="Function"
description="Defines minimum conductor width on outer layers"/>
   <metadataAttr internalName="min_copper_spacing_inner" displayName="Min
Copper Spacing Inner" entityLevel="STEP" type="float" unitsType="Distance"
unitsSize="MIL_MICRON" callback="Function" description="Defines minimum
copper spacing on inner layers"/>
   <metadataAttr internalName="min_copper_spacing_outer" displayName="Min
Copper Spacing Outer" entityLevel="STEP" type="float" unitsType="Distance"
unitsSize="MIL_MICRON" callback="Function" description="Defines minimum
copper spacing on outer layers"/>
   <metadataAttr internalName="number_of_drill_holes" displayName="Number
of Drill Holes" entityLevel="STEP" type="Integer" callback="Function"
description="Number of drill holes"/>
    <metadataAttr internalName="number_of_thru_pads" displayName="Number
of Thru Pads" entityLevel="STEP" type="Integer" callback="ResultAttr"
description="Number of through pads" arguments="num, pth"/>
    <metadataAttr internalName="number_of_tooling_holes"
displayName="Number of Tooling Holes" entityLevel="STEP" type="Integer"
callback="ResultAttr" description="Number of tooling holes"
arguments="num, tooling_holes"/>
    <metadataAttr internalName="number_of_unique_drill_sizes"
displayName="Number of Unique Drill Sizes" entityLevel="STEP"
type="Integer" callback="ResultAttr" description="Number of unique drill
sizes" arguments="num, unique_drill_size"/>
```

```
    <metadataAttr internalName="number_of_vias" displayName="Number of
Vias" entityLevel="STEP" type="Integer" callback="ResultAttr"
description="Number of vias" arguments="num, via"/>
    <metadataAttr internalName="part_number_stamp" displayName="Part
Number Stamp" entityLevel="STEP" type="Boolean" callback="ODBAttr"
description="Defines whether part number stamp should be added on board"/>
    <metadataAttr internalName="plug_vias" displayName="Plug Vias"
entityLevel="STEP" type="Boolean" callback="ODBAttr" description="Defines
whether vias should be plugged"/>
    <metadataAttr internalName="print_et_test_stamp_on_board"
displayName="Print ET Test Stamp on Board" entityLevel="STEP"
type="Boolean" callback="ODBAttr" description="Defines whether required to
print ET stamp on the board"/>
   <metadataAttr internalName="smallest_drill_size" displayName="Smallest
Drill Size" entityLevel="STEP" type="Float" unitsType="Distance"
unitsSize="MIL_MICRON" callback="Function" description="Smallest drill
size"/>
    <metadataAttr internalName="smallest_trace_width"
displayName="Smallest Trace Width" entityLevel="STEP" type="Float"
unitsType="Distance" unitsSize="MIL_MICRON" callback="Function"
description="Smallest trace width"/>
    <metadataAttr internalName="smallest_via_diameter"
displayName="Smallest Via Diameter" entityLevel="STEP" type="Float"
unitsType="Distance" unitsSize="MIL_MICRON" callback="Function"
description="Smallest via diameter"/>
    <metadataAttr internalName="surface_finish_contact_fingers"
displayName="Surface Finish Contact Fingers" entityLevel="STEP"
type="TextWithOptions" callback="ODBAttr" description="Defines surface
finish for contact fingers" options="HASL , Lead-Free HASL , OSP (Entek) ,
ENIG , Deep Gold , Wirebondable Gold , Bare Cu , Immersion Silver , White
Tin , Other"/>
   <metadataAttr internalName="surface_finish_pads" displayName="Surface
Finish Pads" entityLevel="STEP" type="TextWithOptions" callback="ODBAttr"
description="Defines surface finish for exposed pads" options="HASL ,
Lead-Free HASL , OSP (Entek) , ENIG , Deep Gold , Wirebondable Gold , Bare
Cu , Immersion Silver , White Tin , Other"/>
    <metadataAttr internalName="teardrop_allowed_on_inner"
displayName="Teardrop Addition Allowed on Inner" entityLevel="STEP"
type="Boolean" callback="ODBAttr" description="Defines whether adding
teardrops on inner copper layers is allowed"/>
    <metadataAttr internalName="teardrop_allowed_on_outer"
displayName="Teardrop Addition Allowed on Outer" entityLevel="STEP"
type="Boolean" callback="ODBAttr" description="Defines whether adding
teardrops on outer copper layers is allowed"/>
    <metadataAttr internalName="thieving_restrictions_on_inner"
displayName="Thieving Restrictions on Inner" entityLevel="STEP"
type="Text" callback="ODBAttr" description="Defines restrictions for
thieving patterns on inner  layers (free text)"/>
    <metadataAttr internalName="thieving_restrictions_on_outer"
displayName="Thieving Restrictions on Outer" entityLevel="STEP"
type="Text" callback="ODBAttr" description="Defines restrictions for
thieving patterns on outer layers (free text)"/>
   <metadataAttr internalName="ul_code_stamp" displayName="UL Code Stamp"
entityLevel="STEP" type="Boolean" callback="ODBAttr" description="Defines
whether UL Code stamp should be added on board"/>
```

```
    <metadataAttr internalName="unique_serial_nr_stamp"
displayName="Unique Serial  NR Stamp" entityLevel="STEP" type="Boolean"
callback="ODBAttr" description="Defines whether a unique serial nr stamp
should be added on board"/>
    <metadataAttr internalName="via_tenting" displayName="Via Tenting"
entityLevel="STEP" type="Boolean" callback="ODBAttr" description="Defines
whether via tenting can be used"/>
    <metadataAttr internalName="via_treatments" displayName="Via
Treatments" entityLevel="STEP" type="TextWithOptions" callback="ODBAttr"
description="Determine which via treatment should be applied" options="Via
Fill - Conductive, Via Fill - Non-conductive, Via Fill - Copper Plated,
Via Fill - Soldermask Hole Filled, Via Capped, Clearance - Standard
Soldermask Clearance, Dry Film Via Tenting, Soldermask Defined -
Soldermask over Via"/>
  </metadataSection>

  <metadataSection internalName="Assembly" description="Assembly">

    <metadataAttr internalName="cad_package_analysis" displayName="Cad
Package Analysis" entityLevel="STEP" type="Integer" callback="Collective"
description="CAD package analysis: Number of unique cells per side and
total associated pin count and pin pitch"/>
    <metadataAttr internalName="complete_test_required"
displayName="Complete Test Required" entityLevel="STEP" type="Boolean"
callback="ODBAttr" description="Defines whether a test including barrel
testing is required"/>
    <metadataAttr internalName="component_count" displayName="Components
per Step" entityLevel="STEP" type="Integer" callback="Collective"
description="Total component count&#xA;Classified by layer (top, bottom,
etc),&#xA;Then classified by technology (SMD, Thru, etc)"/>
    <metadataAttr internalName="number_of_fiducials" displayName="Number
of Fiducials" entityLevel="STEP" type="Integer" callback="Collective"
description="Number of top and bottom fiducials"/>
    <metadataAttr internalName="number_of_nets" displayName="Number of
Nets" entityLevel="STEP" type="Integer" callback="Function"
description="Number of nets"/>
    <metadataAttr internalName="number_of_accessible_nets"
displayName="Number of Accessible Nets" entityLevel="STEP" type="Integer"
callback="ODBAttr" description="Number of accessible nets"/>
    <metadataAttr internalName="number_of_multi_pin_nets"
displayName="Number of Multi-pin Nets" entityLevel="STEP" type="Integer"
callback="Collective" description="Number of multi-pin nets"/>
    <metadataAttr internalName="number_of_nets_with_probes"
displayName="Number of nets with probes" entityLevel="STEP" type="Integer"
callback="ODBAttr" description="Number of nets with probes"/>
    <metadataAttr internalName="number_of_single_pin_nets"
displayName="Number of single pin nets" entityLevel="STEP" type="Integer"
callback="Collective" description="Number of single pin nets"/>
    <metadataAttr internalName="number_of_un-connected_nets"
displayName="Number of unconnected nets" entityLevel="STEP" type="Integer"
callback="Collective" description="Number of unconnected nets"/>
   <metadataAttr internalName="number_of_test_points" displayName="Number
of Test Points" entityLevel="STEP" type="Integer" callback="Collective"
description="Defines the number of test points (net ends) on top and
bottom sides"/>
```

```
   <metadataAttr internalName="pressfit_technology" displayName="Pressfit
Technology" entityLevel="STEP" type="Boolean" callback="Function"
description="Defines whether Pressfit technology should be used on this
board."/>
    <metadataAttr internalName="smallest_pin_pitch" displayName="Smallest
Pin Pitch" entityLevel="STEP" type="Float" unitsType="Distance"
unitsSize="MIL_MICRON" callback="ResultAttr" description="Smallest pin
pitch" arguments="min, c_pitch"/>
    <metadataAttr internalName="smd_technology" displayName="SMD
Technology" entityLevel="STEP" type="Option" callback="Function"
description="Defines number of sides that have SMD technology."
options="Single sided, Double sided, None"/>
    <metadataAttr internalName="total_number_of_smd_pads"
displayName="Total Number of SMD Pads" entityLevel="STEP" type="Integer"
callback="Collective" description="Total number of SMD pads, classified by
side"/>
  </metadataSection>

  <metadataSection internalName="Final" description="Final QC">

    <metadataAttr internalName="qual_coupon_crosssection_rpt"
displayName="Quality Coupon Cross Section Report" entityLevel="STEP"
type="Text" callback="ODBAttr" description="Defines type of quality coupon
cross section report (free text)"/>
    <metadataAttr internalName="ship_as" displayName="Ship as"
entityLevel="STEP" type="Option" callback="ODBAttr" description="Single
board / assembly panel" options="Single board, Assembly panel"/>
    <metadataAttr internalName="solder_resist_adhesion_tst_rpt"
displayName="Solder Resist Adhesion Test Report" entityLevel="STEP"
type="Text" callback="ODBAttr" description="Defines type of solder resist
adhesion test report (free text)"/>
    <metadataAttr internalName="solderability_test_report"
displayName="Solderability Test Report" entityLevel="STEP" type="Text"
callback="ODBAttr" description="Defines type of solderability test report
(free text)"/>
  </metadataSection>
</metadataSchema>
```

Appendix 2
Frequently Asked Questions

Why is the data in ASCII?

ASCII data provides the user with numerous advantages:

- It is easy to read and understand.
- Translators to and from the data formats are easier to write.
- The data is portable between different architectures, independent of byte order, floating point formats, and so forth.

By compressing the ASCII files using standard compress commands, the size of the data is smaller than the binary equivalent. This is due to the fact that the compression algorithm is adaptive and work very well when certain strings are repeated.

When I wish to rotate a feature pad by 90 degrees, is the aperture rotated left or right?

Clockwise.

Regarding donuts, butterflies, thermals, and moires, do any of these symbols have negative components?

Standard symbols are all positive. All holes in symbols are see-through by definition.

When the start-point and end-point of a feature coincide, is this considered a 360-degree arc or a single point? Can I draw an arc with a square symbol?

A 360-degree arc; there are no single point arcs in the ODB++ format. Arcs can be drawn only with a round symbol.

When I specify an x,y location for text, where will the text string be located?

The version parameter for text records determines text placement. See “T - Text Records” on page 150.

What is the meaning of the optimize field in a netlist file?

It indicates that the net has been optimized and the end-point markers have been removed from mid-points.

In a netlist file, how is the radius field supposed to be set for drills of 0.002 inches through non-SMDs?

The radius field will be 0.001 mils in this case.

In a netlist file, what does the term staggered points mean?

These are points that have been staggered to make them accessible to test probes.

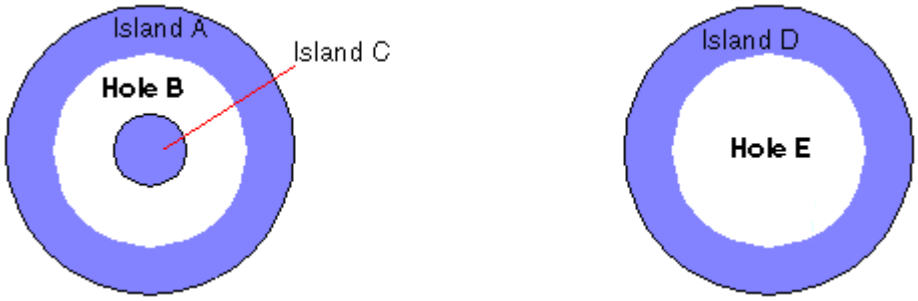
Can feature files of user-defined symbols contain references to other user-defined symbols?

Yes, they can. But recursion, direct or indirect is not allowed.

Regarding surfaces, is there a particular order in specifying holes and islands?

The order of containment must be preserved. Islands precede holes that are contained in them. Holes precede islands that are contained in them.

Take, for example, the following containment order:



Regarding surfaces, does the outermost island come first?

Yes.

Regarding rounded or chamfered rectangles, how do I specify corners?

See “Rounded/Chamfered Rectangles” on page 185.

If I want to offset a rectangular pad in X or Y, should I use a standard symbol and use the standard ODB++ to create the offset? As an example, suppose we have a rect pad 70x50 with an X offset of 5. The feature file for the symbol I create would contain:

```
#
# Symbol name
#
$0 rect70x50
#
```

```
# Pad definition
#
P .005 .0 0 P 0 0
```

Is this correct?

Instead of defining your own symbols with offsets, you should use the standard **rect** symbol, and offset the coordinate that references it (in the layer features file).

Regarding properties (PRP) on components, is there a list of properties that are recognized by the system (such as with system attributes). Where can I find it? The same goes for PRP in the eda/data file.

There is no list of predefined properties in the ODB++ format. These are EDA-specific.

When defining a PKG record (using /steps/step_name/eda/data), it seems that ODB++ expects closed geometries. Is it critical to have only closed elements?

Yes, you should close all polygons.

In what order should the matrix layers be in?

The recommended sequence is: top component, solderpaste, silkscreen, soldermask, followed by copper layers in their order, closed with soldermask, silkscreen, solderpaste, bottom component. Then the drill layers in sequence from where the drills started, all top to the largest depth (largest to smallest), then **layer_2** to the largest depth (largest to smallest), etc. Then rout. At the end are all document/misc layers.

How are the net_num records numbered?

The **net_num** used in the TOP record corresponds to the sequence of the Net records in the **eda/data** file. The first Net record is **net_num 0**, the second is **net_num 1**, and so on.

We have an elaborate tool set to define routing slots and milling contours. How is this data written to ODB++?

Milling (referred to as routing in Fabrication, and not to be confused with routing of traces in design) is handled by defining a 'rout' layer (similar to a drill layer). The features in this layer correspond to the outline of the shapes that need to be cut out. A rout layer is like any other layer, but in order for it to be used during fabrication, should contain only lines, arcs and circular pads (rout machines can also drill).

What net do I assign to points that have no net defined?

All features that do not have a net defined should be assigned to net **$NONE$** that should be defined in the eda/data file. You must add the net and all points to the cadnet file.

Is there any restriction on the maximum line length in an ODB++ file? Can comment lines be more than 500 characters?

The restrictions are different for different files. In general 1,400 characters is the limit, but there are exceptions. Any line over the defined limit will be read with the remainder of the line ignored, so comments can be longer than the limit.