TABLE OF CONTENTS

Generic Requirements for Printed Board Assembly Products Manufacturing Description Data and Transfer Methodology

1 SCOPE

This standard specifies the XML schema that represents the intelligent data file format used to describe printed board and printed board assembly products with details sufficient for tooling, manufacturing, assembly, and inspection requirements. This format may be used for transmitting information between a printed board designer and a manufacturing or assembly facility. The data is most useful when the manufacturing cycle includes computer-aided processes and numerical control machines. The data can be defined in either English or International System of Units (SI) units.

1.1 Focus and intent

The generic format requirements are provided in a series of standards focused on printed board manufacturing, assembly, and inspection testing. This standard series consists of a generic standard (IPC-2581) that contains all the general requirements. There are seven sectional standards that are focused on the XML details necessary to accumulate information in the single file, that address the needs of the manufacturing disciplines producing a particular product.

The sectional standards (IPC-2582 through 2588) paraphrase the important requirements and provide suggested usage and examples for the topic covered by the sectional standard.

1.2 Notation

Although the data would be contained in a single file, the file can have different purposes as described in Section 4. The XML schema used for this standard follows the notations set forth by the W3C and is as follows:

element – Element appears exactly one time

element? – Element may appear 0 or 1 times

element* – Element may appear 0 or more times

element+ – Element may appear 1 or more times

Any IPC-2581 file is composed of a high level element (`IPC-2581`) that contains up to six sub-elements:

Content – information about the contents of the IPC-2581 file

LogisticHeader – information pertaining to the order and supply data

HistoryRec – change information of the file

Bom – Bill of Materials (Material List) information

Ecad – Computer Aided Design (engineering) information

Avl – Approved Vendors List information

2 APPLICABLE DOCUMENTS

The following documents contain requirements which, when referenced, constitutes provisions of IPC-2581. At the time of publication, the editions indicated were valid. All documents are subject to revision and parties entering into agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below.

The revision of the document in effect at the time of solicitation **shall** take precedence.

IPC-T-50 Terms and Definitions for Interconnecting and Packaging Electronic Circuits

IPC-2501 Definition for Web-Based Exchange of XML Data

IPC-2524 PWB Fabrication Data Quality Rating System

IPC-2511 Generic Requirements for Implementation of Product Manufacturing Description Data and Transfer XML Schema Methodology

IPC-2571 Generic Requirements for Electronics Manufacturing Supply Chain Communication - Product Data eXchange (PDX)

IPC-2576 Sectional Requirements for Electronics Manufacturing Supply Chain Communication of As-Built Product Data - Product Data eXchange

IPC-2578 Sectional Requirements for Supply Chain Communication of Bill of Material and Product Design Configuration Data - Product Data eXchange

IPC-7351 Generic Requirements for Surface Mount Design and Land Patterns

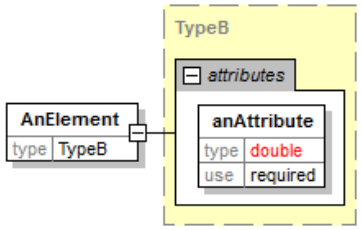
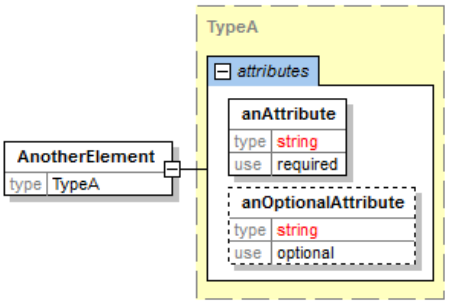
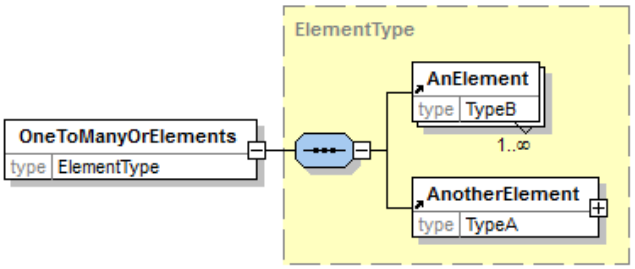
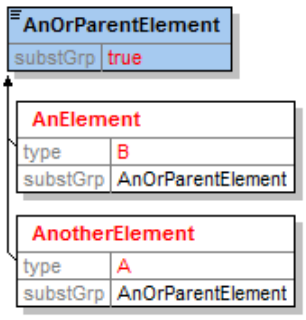
2.1 Documentation conventions

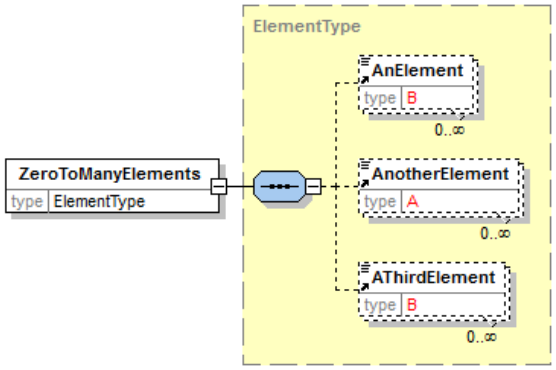
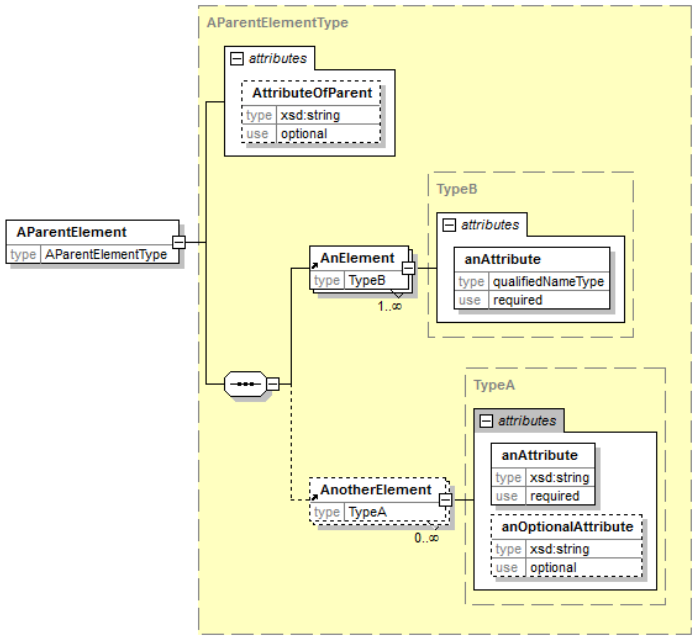
The XML file format standard and the XML Schema definition language standard, as defined the by World Wide Web Consortium (W3C), have been adopted by IPC for use in the IPC-2500 series of standards.

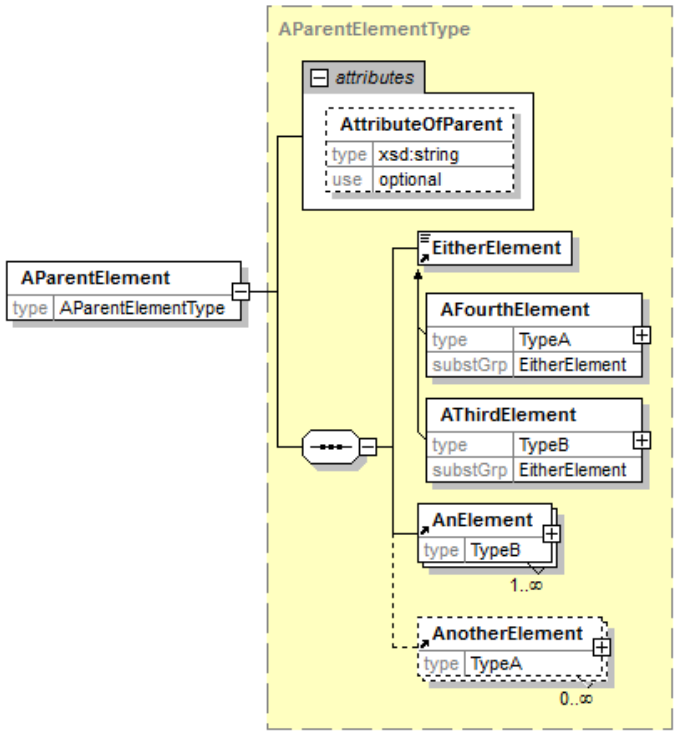
In addition to the text based schema notation, this document provides graphical representation of the structure of the file format. The XML diagrams are designed to effectively illustrate the structure and cardinality of elements and attributes that make up any IPC-258X file. The notation in the graphics does not provide a complete visualization of the schema definition for the file format, but it does provide a good top down overview. Should there be any conflict between the graphical notation and the schema notation, the authoritative definition is the schema notation.

Table 1 provides an overview of the graphical notation used in the document.

Table 1 Graphical Notation Overview

This diagram depicts an element named AnElement that is of type TypeB. There is one attribute, named anAttribute, which is of type double. The attribute is required.	
Example: <AnElement anAttribute="14.44e-3"/> Note that all attribute values must be enclosed in quotes, regardless of type.	
This diagram depicts an element named AnotherElement that is of type TypeA with two attributes. The attribute anAttribute is required. The second attribute, anOptionalAttribute, is optional. Both attributes are of type string.	
Examples: <AnotherElement anAttribute="red" anOptionalAttribute="a string" /> <AnotherElement anAttribute="blue" />	
The ElementType OneToManyOrElements is the parent of an unordered list of one or more instances of the elements AnElement and AnotherElement. The type TypeA occurs only once while TypeB can occur many times.	
The arrow indicates a substitution of the element named AnOrParentElement. The substitution declares that one and only one occurrence is allowed. The AnOrParentElement can have one of AnElement or AnotherElement as a child element.	

The optional indication for the three element children shows that none may be present or many versions of all three are possible.	
This diagram depicts an element, AParentElement, of type AParentElementType. This element has one attribute, attributeOfParent, which is optional. The lines indicate that occurrences of AnElement and AnotherElement appear in the order with AnElement indicated as required while AnotherElement is optional. Both elements have a required attribute; however, if AnotherElement is instantiated, anOptionalAttribute is also possible.	

This diagram depicts a type, AParentElementType, that contains a sequence starting with one of AThirdElement or AFourth element followed by 1-n AnElement and an optional final AnotherElement. The AParentElement also has an optional attribute.	

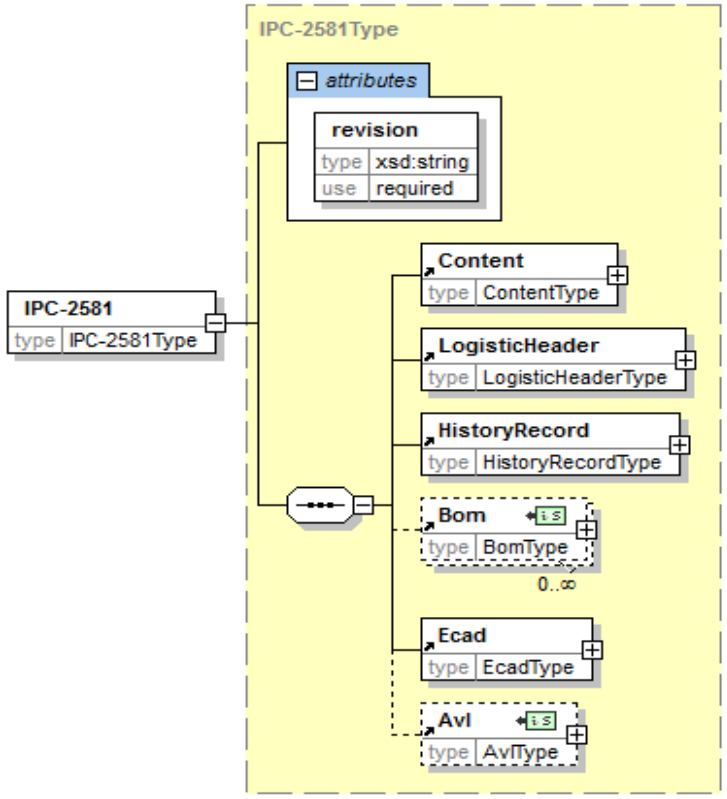
3 REQUIREMENTS

The XML schema contained in this document describes the structure of a generic computer-aided manufacturing exchange format, called IPC-2581. The document specifies data elements specifically designed to establish the information exchange related to the data needed by printed board manufacturing, and assembly including inspection of those products.

The XML schema defines the configuration of mandatory and optional elements, as well as mandatory and optional attributes. The Top Level (TopElement) of the schema contains six major elements. The schema notation specifies that the 6 top-level elements are required to appear in the order shown in Figure 1. The order of appearance in the file is significant. For instance, the appearance of graphics on a layer is dependent on the order of appearance in the file. The order is also important because elements often reference information that is defined elsewhere in the file in order to eliminate redundancy within the file. The file is structured to allow all references to be resolved in one pass.

An implementation of the XML schema must be able to facilitate the reading and/or writing of all characteristics defined within the requirements stated in the `Mode` function of this standard. Some tools may have only read capability; some may have only write capability. Some tools may have both read/write capability. All schema defined in the standard as mandatory (1-1, 1-n occurrences) **shall** be executed as appropriate. Tool providers **shall** identify their capability by Mode (e.g. FULL, FABRICATION, ASSEMBLY etc.).

Each element has a specific function or task. Accordingly, the information interchange for a specific purpose is possible only if that element is populated. The ability to select those characteristics that are appropriate for a given task makes the schema a robust methodology for defining only those areas and characteristics that are necessary to produce a given product. Figure 1 shows the potential children elements of the Top Element (IPC-2581).



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
IPC-2581	IPC-2581Type	The generic name of the file used to describe Design, Fabrication, Assembly, and Test information in accordance with the IPC-2581 standard.	1-1
revision	string	The status of the IPC-2581 file content.	1-1
Content	ContentType	The `Content` element defines the function of the file, and references the major sections of the product description (i.e., Step, Layer, Bom and Avl). In addition, there are six dictionaries indicated in Content that would contain the pre-described information needed for the file details.	1-1
LogisticHeader	LogisticHeaderType	The `LogisticHeader` describes information pertaining to ordering and delivery, by identifying individuals and locations responsible for these functions.	1-1
HistoryRecord	HistoryRecordType	The `HistoryRecord` element provides a sequential change number for the IPC-2581 file. The number is changed every time the controlled version of the IPC-2581 file is modified. Also identified are the change approval conditions.	1-1
Bom	BomType	The Bill of Materials for the board is a list of all the different components to be used for the assembly of the board, arranged by their OEM Design Number (ODN) and the materials used to fabricate the printed board.	0-n
Ecad	EcadType	The `Ecad` section describes the Computer Aided Design data of the job, including all the graphical description of the layers, component location, panel design, etc.	1-1
Avl	AvlType	The `AVL` element contains the list of matching manufacturer's part numbers (MPNs) and vendor information of a certain component parts. Approved vendors of printed board materials may also be included.	0-1

Figure 1 - IPC-2581 Children Element

3.1 Rules concerning the use of XML and XML Schema

The rules required to define syntax and semantics of the 258X file format notation have been simplified by the adoption of the W3C standards for XML Schema and XML file formats. These two standards are well specified by the W3C.

The popularity of these standards has led to the development of many commercial and open source software tools and libraries that conform to the W3C standards.

A IPC-2581 file begins with the notation of the revision of the generic standard and the latest revision letter and amendment number, if any amendments exist, e.g. <IPC-2581 revision="B1">, and ends with the </IPC-2581> tag. The content between these tags must match the .xsd definition of the IPC-2581 schema as defined by the IPC-2581 XML. The latest specification and schema release is found at: http://webstds.ipc.org/standards.htm#x2580.

3.1.1 File readability and uniformity

A valid IPC-2581 file must conform to the W3C Canonical XML format. The format is defined by the http://www.w3.org/TR/xml-c14n specification. Software tools exist that will take malformed XML and automatically generate Canonical XML.

3.1.2 File markers

An optional checksum can be appended following the </IPC-2581> tag. The checksum is an MD5 message digest algorithm (see Internet RFC 1321: http://www.ietf.org/rfc/rfc1321.txt) that is base64 encoded. The checksum starts with the "<" character of the <IPC-2581> tag and ending with the ">" character of the closing </IPC-2581> tag. The checksum follows immediately after the ">" character of the closing </IPC-2581> tag.

The digest provides a 128-bit checksum of the IPC-2581 file contents. The MD5 signature must be base64 encoded (see IETF RFC 1421 for the base64 algorithm) to convert the MD5 signature to a US-ASCII, base64 string. An end of line character will indicate the end of the base64 encoded MD5 signature.

3.1.3 File extension

The file extension for an IPC-2581 file is .xml.

3.1.4 File remarks

The IPC-2581 format permits file remarks using the standard XML commenting notation. They are only to be used to support debugging software. A parser may ignore and discard remarks when reading an IPC-2581 file. File remarks are never to be used to represent design or manufacturing information.

3.1.5 Character set definition

The XML standard uses the Unicode character set. This character set covers the characters used in hundreds of written languages. The XML standard allows several of the Unicode encoding formats to be used in an XML file. IPC-2581 requires the use of the UTF-8 character encoding of the Unicode character set. Although comments and user assigned names may be in any language of choice, all qualified names or enumerated string names **shall** be in English only.

3.1.6 Child Element Order

The IPC-2581 schema uses the <sequence> order indicator throughout, which means that all child elements of a given parent element must be instantiated in an IPC-2581 file in the order in which they are defined in the schema. For example, referring to section 8.2.3.10.3, consider the element `Pad`, which has has four child elements `Xform`, `Location`, `Feature` and `PinRef`, defined in that order, as shown below:

```
<xsd:element name="Pad" type="PadType"/>
<xsd:complexType name="PadType">
    <xsd:sequence>
      <xsd:element ref="Xform" minOccurs="0"/>
      <xsd:element ref="Location"/>
      <xsd:element ref="Feature"/>
      <xsd:element ref="PinRef" minOccurs="0"/>
    </xsd:sequence>
```

The element Xform, if it exists (as it is optional) must occur first in any instance of Pad, followed by Location, and so on, as shown in the example below:

```
<Pad padstackDefRef="LS32X67">
    <Xform rotation="90.000"/>
    <Location x="7.1500" y="10.1500"/>
    <StandardPrimitiveRef id="RECTCENTER_3"/>
    <PinRef pin="34"/>
</Pad>
```

If the child elements appear in any other order in an IPC-2581 file then the XML schema of that file is strictly invalid.

3.1.7 XML Validation and Schema File Location

An attribute `schemaLocation` can be appended to the attribute list of the initial <IPC-2581> tag of any given IPC-2581 file. Its value must be two URIs separated by a space, the first URI, http://webstds.ipc.org/2581, being fixed as the namespace location, and the second, http://www.ipc.org/2581/IPC-2581C.xsd, being the location of the IPC-2581 revision C schema definition (.xsd) file. This can enable validation tools to be able to validate the file contents against the schema definition. It is highly recommened that validation is done, to ensure your IPC-2581 files conform to the schema. The following is an example of the `schemaLocation` attribute, pointing to the IPC-2581 revision C .xsd file on the IPC website.

```
<IPC-2581 revision="C" xmlns="http://webstds.ipc.org/2581" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://webstds.ipc.org/2581 http://www.ipc.org/2581/IPC-2581C.xsd">
```

The following is a list of all current and previous revision schema files available:
http://www.ipc.org/2581/IPC-2581C.xsd
http://www.ipc.org/2581/IPC-2581B1.xsd
http://www.ipc.org/2581/IPC-2581B.xsd
http://www.ipc.org/2581/IPC-2581A.xsd

3.2 Naming attributes within a IPC-2581 file

There is a type definition called `qualifiedNameType` that is used to name certain types of attributes in an IPC-2581 file. This type includes an attribute prefix in the name to allow for multiple products (boards, assemblies, or panels) to be defined in the single IPC-2581 file. See section 3.2.1 for more details.

3.2.1 Data organization and identification rules

To prevent collisions between attributes having the same value used by the different partitions (boards, assemblies, or panels) within a single IPC-2581 file, an attribute prefix in the `qualifiedNameType` attribute type can be used to separate the values into ‘namespaces’ per partition. (Note that these ‘namespaces’ alter attribute values, so they should not to be confused with XML namespaces, which alter attribute or element names.) This is necessary because an IPC-2581 file may contain information describing an arbitrary collection of partitions that are allowed within a single IPC-2581 file. That is, if an attribute is of type `qualifiedNameType` then it can be of the form A:B, where B is the attribute value, and A is a prefix that can make the value unique if repeated across multiple products in the same file. For example, if 3 boards in the same file all have a refDes R1, then R1 could be represented by X:R1, Y:R1, Z:R1, where X,Y, and Z are unique prefixes assigned to each of the 3 boards. This mechanism also prepares the way for a distributed database of IPC-2581 design data in which the data can be trusted to be universally unambiguous.

3.2.2 The Use of XML elements and types

A comprehensive overview of XML Schema can be found in the W3C XML Schema Primer. This section briefly describes the decisions that were made in the development of the IPC-258X schema. Reviewing the Primer is recommended prior to reading this section.

The XML Schema defines a namespace mechanism that can be used when defining element names. The W3C also provides a set of general purpose element and attribute types, such as xsd:string, xsd:double, and xsd:datetime. The IPC-2581 format uses these standard types, however the documentation of the IPC-258X standard has been defined without the use of a namespace prefix for element names within an IPC-258X file.

Each of the schema elements has a prefix, “xsd:”, which is associated with the XML Schema namespace through the declaration, xmlns:xsd=”http://www.w3.org/2000/08/XMLSchema”, that appears in the schema element. The prefix xsd: is used by convention to denote the XML Schema namespace, although any prefix can be used. The same prefix, and hence the same association, also appears on the names of built-in simple types, e.g. xsd:string. The purpose of the association is to identify the elements and simple types as belonging to the vocabulary of the XML Schema language rather than the vocabulary of the schema author.

In XML Schema, there is a basic difference between complex types that allow elements in their content and may carry attributes, and simple types that cannot have element content and cannot carry attributes. There is also a major distinction between definitions that create new types (both simple and complex), and declarations that enable elements and attributes with specific names and types (both simple and complex) to appear in document instances.

New complex types are defined using the `complexType` element and such definitions typically contain a set of element declarations, element references, and attribute declarations. The declarations are not themselves types, but rather an association between a name and constraints that govern the appearance of that name in documents governed by the associated schema. Elements are declared using the “element,” and attributes are declared using the “attribute.”

3.2.3 Attribute base types (governing templates)

The attribute basic types (SimpleTypes) provided by XML Schema are defined by the W3C. They are easy to distinguish from the IPC-258X types because the W3C type is always prefixed with “xsd:”. The W3C datatypes are defined in http://www.w3.org/2000/10/XMLSchema (XML Schema Part 2).

Table 2 defines those W3C basic types that are used to define attributes in the IPC-258X schema. The xsd:string type is constrained to create specific base types for special purpose strings, such as `qualifiedNameType`. The rules for special number types and the date format are also defined. Table 3 defines those basic types that have been standardized for use within the IPC-258X format.

Table 2 Governing Template Basic Types Defined by W3C

xsd:string	A W3C standard data type for a Unicode character string. The characters are from the UTF-8 character set as defined in http://www.ietf.org/rfc/rfc2279.txt.
xsd:double	A W3C standard data type for a binary floating-point number. The W3C definition of xsd:double is in http://www.w3.org/TR/xmlschema-2/. The xsd:double is a number where the value can be positive, negative, integer or floating point, with at least 7 digits of precision. Numbers are assumed to be positive but can be explicitly designated as positive by preceding the number with a '+' (ASCII decimal 43) character. Negative numbers must be explicitly designated as negative by a preceding '–' (ASCII decimal 45) character. An internal representation of an IEEE double precision floating-point number is assumed. This range of values for IEEE doubles is defined as 3.4x10-38 ≤ value ≤ 3.4x10+38. The format for representing a double is the same as the format used in the computer languages C, Perl, Python, or TCL. For example, all the following are legal numbers: 1.005 ; 0.01; .01; -2.334e-33; .224e-2
xsd:nonNegativeInteger	A W3C standard data type for non-negative integer numbers. The W3C definition of xsd:nonNegativeInteger is in http://www.w3.org/TR/xmlschema-2/. The range of values allowed are 0 ≤ value ≤ 2147483647 (the non-negative values that fit in a 32 bit signed integer).
xsd:positiveInteger	A W3C standard data type for positive integer numbers. The W3C definition of xsd:positiveInteger is in http://www.w3.org/TR/xmlschema-2/. The range of values allowed are 1 ≤ value ≤ 2147483647 (the positive values that fit in a 32 bit signed integer).
xsd:dateTime	The W3C standard data type for the current date and time is xsd:dateTime. (See http://www.w3.org/TR/NOTE-datetime-970915.html.) The following formats from the W3C specification are recommended for 258X files: Complete date plus hours, minutes and seconds: YYYY-MM-DDThh:mm:ssTZD (e.g. 1997-07-16T19:20:30.4536+01:00) Complete date plus hours, minutes, seconds and a decimal fraction of a second: YYYY-MM-DDThh:mm:ss.sTZD (e.g. 1997-07-16T19:20:30.45+01:00) where: YYYY = four-digit year MM = two-digit month (01=January, etc.) DD = two-digit day of month (01 through 31) hh = two digits of hour (00 through 23) (am/pm NOT allowed) mm = two digits of minute (00 through 59) ss = two digits of second (00 through 59) s = one or more digits representing a decimal fraction of a second TZD = time zone designator (Z or +hh:mm or –hh:mm)
xsd:anyURI	A W3C standard data type for hyperlinks. The W3C definition of xsd:anyURI is in http://www.w3.org/TR/xmlschema-2/.
xsd:unsignedByte	The W3C standard for an unsigned byte (an unsigned 8 bit integer with a value between 0-255.) The W3C definition of xsd:unsignedByte is in http://www.w3.org/TR/xmlschema-2/.
xsd:base64Binary	The data is encoded using base64. (see IETF *RFC 1421* for the base64 algorithm and http://www.w3.org/TR/xmlschema-2/#base64Binary)

Table 3 Governing Template Basic Types Defined by IPC

qualifiedNameType	The `qualifiedNameType` data type is a data type defined for the IPC-258X series. The type is a restricted xsd:string data type where the pattern of the string must match the regular expression as shown by the pattern value in the following definition: <xsd:simpleType name = “qualifiedNameType”> <xsd:restriction base = “xsd:string”> <xsd:pattern value="([^:]+)(:[^:]+)?"/>;]*)*"/> </xsd:restriction> </xsd:simpleType> Note that, in the pattern, + means one or more occurrence, and ? means zero or one occurrence. So the pattern restricts a string to: • one or more occurrences of ANY character that is NOT a colon, followed by • zero or one occurrences of (a colon followed by one or more occurrences of ANY character that is NOT a colon) Examples of strings that match the pattern are: “prefix:name”, “name”, “3%+_?\22”.
nonNegativeDouble	The `nonNegativeDouble` data type is defined for the 258X series. The type restricts an xsd:double to positive numbers, inclusive of 0. The non-negative range of values for IEEE doubles is defined as $0.0 \leq \text{value} \leq 3.4 \times 10^{38}$.

3.2.4 Coordinate system and transformation rules

Any geometry defined in a IPC-2581 file is defined in a Cartesian coordinate system. The x coordinates become more positive going from left to right (west to east). The y coordinates become more positive going from bottom to top (south to north). The primary side (TOP) of the board, coupon, or panel is in the x-y plane of the coordinate system with the primary side facing up.

The illustration in Figure 2 provides a perspective drawing of a board and a coordinate system. Each product in a 258X file is defined relative to a local coordinate system for the product. The point of origin of the product is located at (0,0) in the local coordinate system.

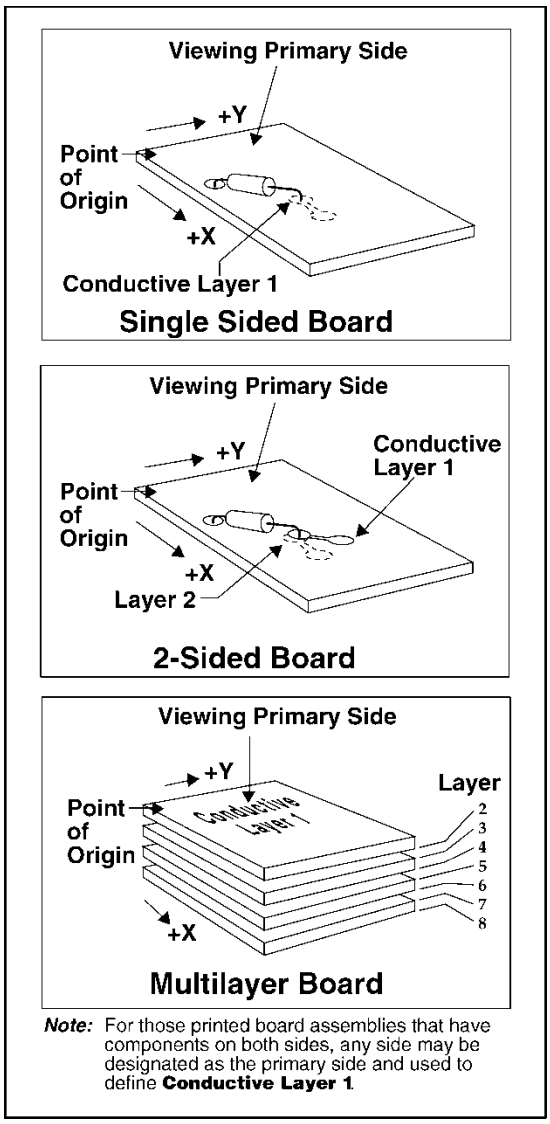


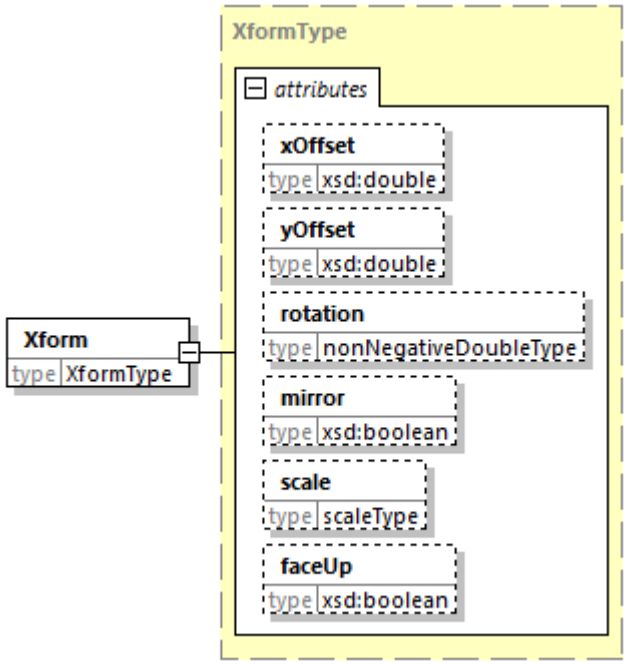
Figure 2 - Printed Board Viewing

3.3 Transformation characteristics (Xform)

The `Xform` element defines a transformation that may be used throughout this specification to define a modification of the original stored data in the Dictionaries, the location and orientation of physical features. `Features`, `Shapes`, `Primitives` or other graphics in the file may manipulate relative to their local Cartesian coordinate system by the values set in the transform. The `Xform` element can define a modification of the pre-defined feature's point of origin, and then apply rotation, mirroring, scaling and location (x and y) of the image.

The units of measure are defined in the `CadHeader` as an attribute that describes the details of all the features in the `Ecad` section. In addition, units of measure are also defined in each of the Dictionaries that contain graphical information. These are `DictionaryStandard`, `DictionaryUser`, `DictionaryFont`, and `DictionaryLineDesc`. When a pre-defined image from one of the dictionaries is used in the `Ecad` section, the units of measure must match.

The order of the transformation **shall** always follow the order of the fields in the description. This would be to 1) modify the origin; 2) apply rotation; 3) mirror image; and 4) scale.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Xform	XformType	An element that provides the ability to reset the point of origin of a predefined target that can be scaled, mirror imaged or rotated.	0-1
xOffset	double	The x offset of a previously defined shape relative to its original point of origin in order to define a new origin used for rotation, mirroring, scaling and location. The `CadHeader` defines the units of measure.	0-1
yOffset	double	The y offset of a previously defined shape relative to its original point of origin in order to define a new origin used for rotation, mirroring, scaling and location. The `CadHeader` defines the units of measure.	0-1
rotation	nonNegativeDouble Type	Defines the rotation of a shape about the local origin in degrees. Positive rotation is always counter-clockwise as viewed from the board TOP (primary side). Degree accuracy is expressed as a two place decimal i.e., 45.15; 62.34	0-1
mirror	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lower case). True indicates that the image is mirrored; False indicates that it is not. Mirror is implemented as “false” by default and MUST be set to “true” if the features/components are to be inverted when applied to a component it is placed on the underside of its given layer. When the given layer is not the bottom layer then the component is embedded within the PCB substrate.	0-1
scale	scaleType	An attribute that defines a “double” dimension whose minExclusiveValue=0.0 representing the multiplication factor of all x and y dimensions. The scale factor does not apply to angular values.	0-1
faceUp	boolean	An attribute applicable to the placement ot a component. When false the component is placed “face down”, which is the default orientation per the library definition. When true the component is placed on its back, so that any pins on its underside are now on its topside and vice versa.	0-1

3.3.1 The x and y Offset attributes

The `xOffset` and `yOffset` attributes are of type xsd:double. They define the x and y offset of a shape relative to the origin of a Cartesian coordinate system. The definition of shapes can be nested and the x and y attributes are always interpreted relative to the local coordinate system of the shape to which the transformation applies. The default value for x and y is 0.0.

3.3.2 The rotation attribute

The `rotation` attribute is of type `nonNegativeDouble` that defines the rotation of a shape about the local origin. The interpretation of the value is set globally in the file to units of degrees. The range of the rotation parameter for DEGREES is 0.00 to 360.00 expressed as a `nonNegativeDouble` with an accuracy of a two place decimal. Positive rotation is always counter-clockwise as viewed from the board TOP (primary side), even if the component that is being rotated is on the board BOTTOM (secondary side). Rotation defaults to 0.0, and can be applied to text, or any physical shape.

3.3.3 The mirror attribute

The `mirror` attribute is of data type `boolean`. This type is an enumerated string indicating true or false. The default value for MIRROR is “false”. When mirror is set to “true” it indicates that all x dimensions are set to a –x value. The proper interpretation of the mirror and rotate attributes are shown in Figure 3. The example shows a unique artwork (14-pin DIP device) placed on the top and bottom of a board at 90.00° rotations.

3.3.4 The scale attribute

The `scale` attribute is of data type `scaleType`. The `scale` attribute is a “double” that must have a value greater than zero. All x and y dimensions of a geometry are multiplied by the scale attribute. The scale factor does not apply to angular values. The default value is 1.0.

3.3.5 The x and y Location attributes

The `xLocation` and `yLocation` attributes are of type xsd:double. They define the x and y position where a feature, component, text or other shape is placed. The `xLocation` and `yLocation` coordinate positions a shape by its original origin or its modified origin (x and y Offset) relative to the origin of the image upon which the feature, component etc. is to be located. Mirroring, rotating, and scaling may all apply to the location of the shape as indicated by the `Xform`.

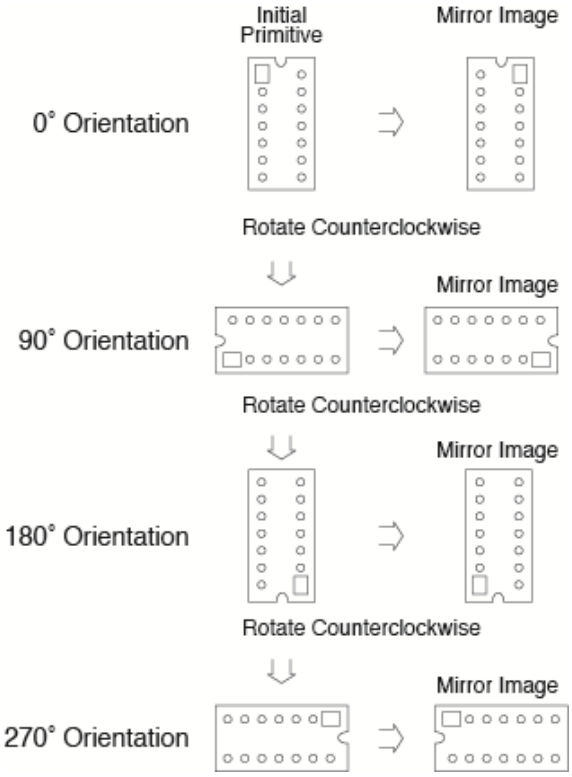


Figure 3 - Rotation and Mirror Diagram

3.3.6 The faceUp attribute

As with the `mirror` attribute, when `faceUp` is set to true in the `Xform` of a component then all x dimensions are set to a –x value. The difference is that `mirror` places the component on the underside of the mounting layer whereas `faceUp` places the component on the same side as the mounting layer.
There is one exception, which is when both `mirror` and `faceUp` are true. In this case the –x transformations cancel each other out, resulting in no -x transformation, and the `mirror` placement wins, so that the component is placed on the underside of the layer. The result is an effective shift in the -y direction such that the component "moves through" the mounting layer to its other side.
Figure 4 shows the outcome of combining `mirror` and `faceUp` attributes, on a component with square, red pins on its underside, and round, yellow pins on its topside.

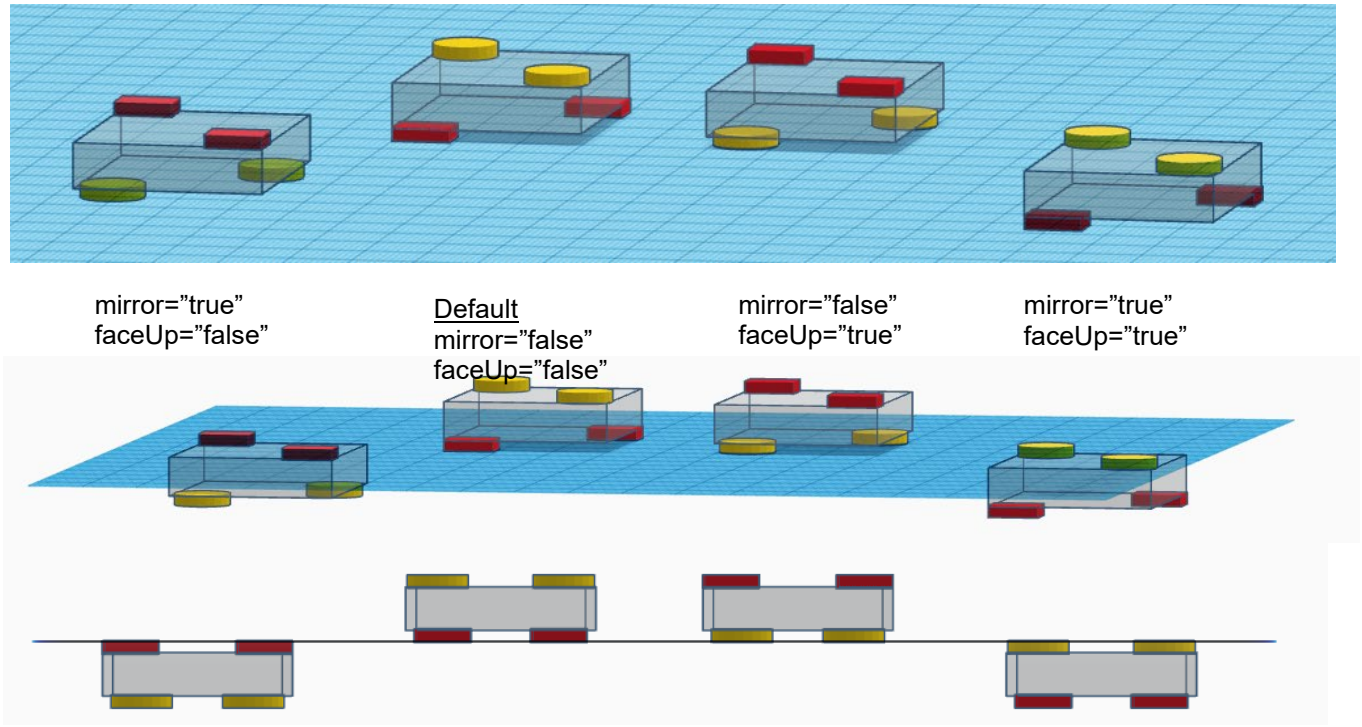


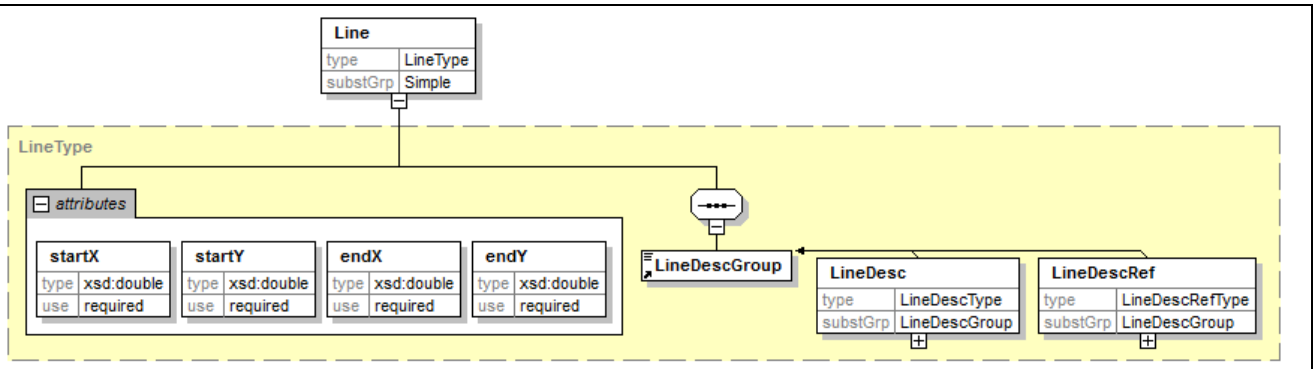
Figure 4 – Mirror and FaceUp Diagram

3.4 Base Elements

The following elements are used throughout the IPC-2581 schema, either standalone or as ‘building block‘ child elements of more complex elements.

3.4.1 Line

The Line element defines an individual line segment. The start and end points and description of the line are defined.

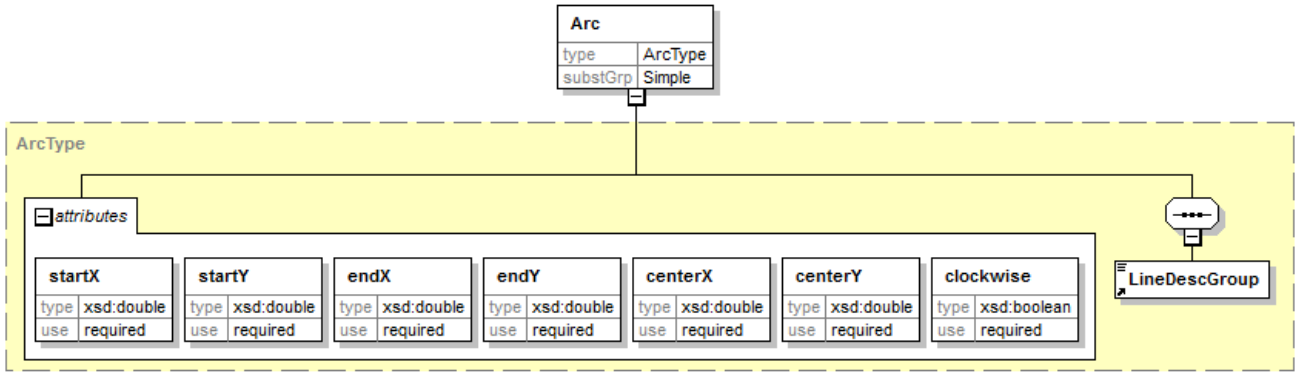


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Line	LineType	The Line element is used to describe an individual line segment. The lineWidth and lineEnd conditions are defined by the LineDescGroup or when the line is instantiated. The lineEnd default is ROUND.	0-n
startX	double	Beginning x-coordinate of the `Line`.	1-1
startY	double	Beginning y-coordinate of the `Line`.	1-1
endX	double	Ending x-coordinate of the `Line`.	1-1
endY	double	Ending y-coordinate of the `Line`.	1-1
LineDescGroup	ABSTRACT	A substitution group that specifies the properties (width, style, etc.) of the line. The elements to be substituted are either `LineDesc` or `LineDescRef`. Refer to section 3.5.5 LineDesGroup	1-1

```
<Line startX = "0.0" startY = "0.0" endX = "12.6" endY = "9.4">
    <LineDesc lineEnd = "ROUND" lineWidth = "1.0"/>
</Line>

<Line startX = "3.6" startY = "4.8" endX = "-4.8" endY = "-3.6">
    <LineDescRef id = "MediumLine"/>
</Line>
```

3.4.2 Arc

The Arc element describes a curved line relative to a center of rotation. The start and end points of the line forming the arc are defined, as well as the center of rotation and description of the line.

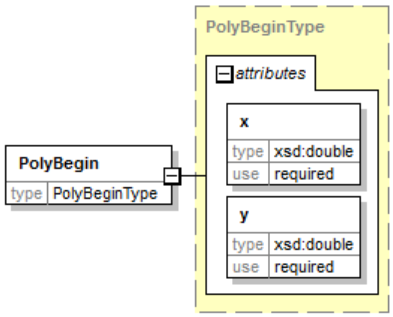


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Arc	ArcType	The `Arc` element represents an arc. Arcs are curves (defined by three sets of coordinates: `startX`, `startY`, `endX`, `endY` and `centerX`, `centerY`. The width of the arc is set by the `LineDescGroup` substitution group or applied when the `Arc` is instantiated.	1-1
startX	double	Beginning x-coordinate of the `Arc`.	1-1
startY	double	Beginning y-coordinate of the `Arc`.	1-1
endX	double	Ending x-coordinate of the `Arc`.	1-1
endY	double	Ending y-coordinate of the `Arc`.	1-1
centerX	double	The X location for the origin of the radius of the circular `Arc`.	1-1
centerY	double	The Y location for the origin of the radius of the circular `Arc`.	1-1
clockwise	Boolean	A required boolean to set the direction of rotation of the curve. If “true” then the direction of rotation is clockwise. If “false” then the direction of rotation is counterclockwise	1-1
LineDescGroup	ABSTRACT	A substitution group that specifies the properties (width, style, etc.) of the line that forms the arc. The elements to be substituted are either `LineDesc or LineDescRef`. Refer to section 3.5.5 LineDesGroup	1-1

```
<Arc startX = "8.0" startY = "0.0" endX = "-8.0" endY = "0.0" centerX = "0.0" centerY = "0.0" clockwise = “false”>
    <LineDescRef id = "MediumLine"/>
</Arc>
<Arc startX = "-8.0" startY = "0.0" endX = "0.0" endY = "8.0" centerX = "0.0" centerY = "0.0" clockwise = “true”>
    <LineDesc lineEnd = "ROUND" lineWidth = "1.0"/>
</Arc>
```

3.4.3 PolyBegin

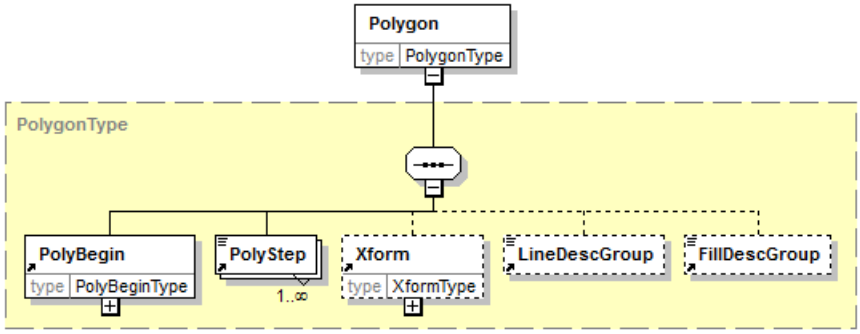
The `PolyBegin` element defines the starting point of Polygon or Polyline element.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
PolyBegin	PolyBeginType	The `PolyBegin` element defines the starting point of a polygon or polyline.	1-1
x	Double	The X co-ordinate starting point.	1-1
y	Double	The Y co-ordinate starting point.	1-1

3.4.4 Polygon

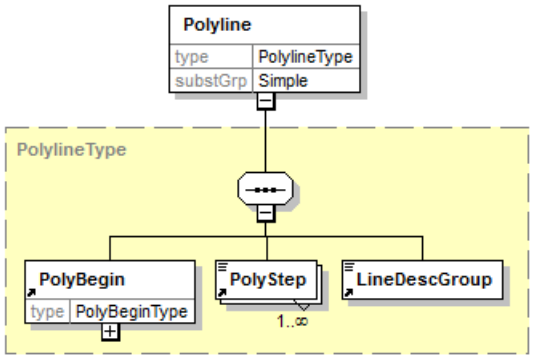
A polygon is a closed shape whose edges do not cross.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Polygon	PolygonType	A closed shape whose edges do not cross, the coordinates of which are defined relative to the local coordinate system of the polygon.	1-1
PolyBegin	PolyBeginType	The `PolyBegin` element defines the starting point of the polygon.	1-1
PolyStep	ABSTRACT	The `PolyStep` element is a substitution group that defines a curved (`PolyStepCurve`) or straight line (`PolyStepSegment`) continuation of the polygon. Refer to section 3.5.7 PolyStep. The final X and final Y coordinates must match those of the `PolyBegin` element to signify that the shape is closed.	1-n
Xform	XformType	An element that provides the ability to reset the point of origin of a predefined target that can be scaled, mirror imaged or rotated. *Note: the use of this is not recommended, as it may be removed in the next revision. It is better to assign the Xform to the shape instance.*	0-1
LineDescGroup	LineDescGroupType	A substitution group that specifies either LineDesc or LineDescRef. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either FillDesc or FillDescRef. Refer to section 3.5.6 FillDesGroup	0-1

3.4.5 Polyline

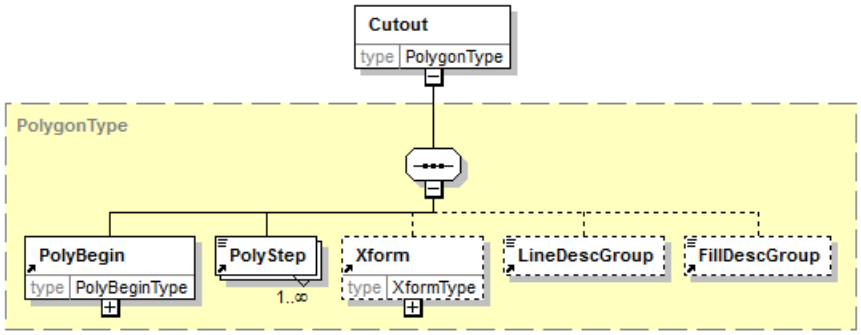
The `Polyline` element consists of a series of lines that define a particular grouping configuration. These line segments do NOT result in a closed shape, however they can be pre-defined and re-used as needed.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Polyline	PolylineType	The `Polyline` element consists of a series of lines that define a particular grouping configuration. These line segments do NOT result in a closed shape, however they can be pre-defined and re-used as needed. The `lineWidth` and `lineEnd` of the `Polyline` are defined at the time the `Polyline` is instantiated or predefined.	1-1
PolyBegin	PolyBeginType	The `PolyBegin` element defines the starting point of the polyline.	1-1
PolyStep	ABSTRACT	The `PolyStep` element is a substitution group that defines a curved (`PolyStepCurve`) or straight line (`PolyStepSegment`) continuation of the polyline. The final X and final Y coordinates must NOT match those of the `PolyBegin` element to signify that the shape is open	1-n
LineDescGroup	ABSTRACT	A substitution group that specifies the properties (width, style, etc.) of the line that forms the polyline. The elements to be substituted are either `LineDesc or LineDescRef`. Refer to section 3.5.5 LineDesGroup	1-1

3.4.6 Cutout

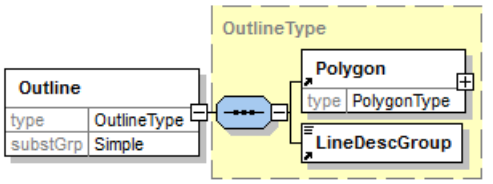
A `Cutout` element defines a polygon closed shape whose edges do not cross. `Cutout` elements must always be paired with a sibling `Polygon` element, with co-ordinates and size such that the cutout shapes are fully enclosed within the boundry of the polygon, so that the cutouts represent the absence of material within that polygon.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Cutout	PolygonType	A polygon closed shape whose edges do not cross, which represents the absence of material within the sibling polygon shape.	1-1
PolyBegin	PolyBeginType	The `PolyBegin` element defines the starting point of the cutout	1-1
PolyStep	ABSTRACT	The `PolyStep` element is a substitution group that defines a curved (`PolyStepCurve`) or straight line (`PolyStepSegment`) continuation of the polygon. Refer to section 3.5.7 PolyStep. The final X and final Y coordinate must match those of the `PolyBegin` element to signify that the shape is closed.	1-n
LineDescGroup	LineDescGroupType	A substitution group that specifies either LineDesc or LineDescRef. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either FillDesc or FillDescRef. Refer to section 3.5.6 FillDesGroup	0-1

3.4.7 Outline

The `Outline` element consists of the following characteristics using a `Polygon` shape to represent a closed shaped group of lines.

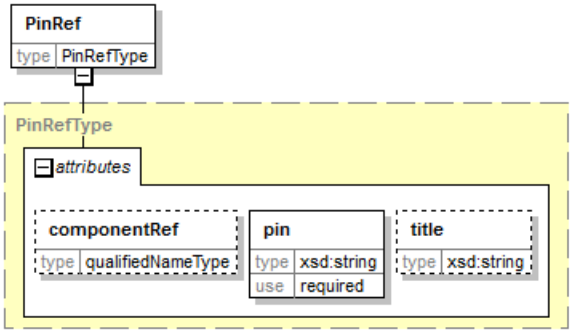


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Outline	OutlineType	An element that has as its sub elements the `Polygon` and `LineDesc` elements in order to define a closed shape that has a line width.	1-1
Polygon	PolygonType	The standard description for the `Polygon` characteristic must be a closed shape. The point of origin of the shape is identified as the 0:0 coordinate. The element includes the start of the polygon definition (`PolyBegin`) and the appropriate number of PolySteps to complete the closed shape. The lineWidth is through the `LineDesc` substitution group or defined at a time when the `Polygon` is instantiated.	1-1
LineDescGroup	LineDescGroupType	A substitution group that specifies the properties (width, style, etc.) of the line that forms the outline. The elements to be substituted are either `LineDesc or LineDescRef`. Refer to section 3.5.4.1 LineDesGroup	1-1

```
<Outline>
   <Polygon>
      <PolyBegin x = "0.0" y = "6.4"/>
      <PolyStepSegment x = "12.6" y = "6.4"/>
      <PolyStepCurve x = "14.8" y = "4.2" centerX = "12.6" centerY = "4.2" clockwise = “true”/>
      <PolyStepSegment x = "14.8" y = "0.0"/>
      <PolyStepSegment x = "0.0" y = "0.0"/>
      <PolyStepSegment x = "0.0" y = "6.4"/>
   </Polygon>
   <LineDescRef id = "FineLine"/>
</Outline>

<Outline>
   <Polygon>
      <PolyBegin x = "-10.4" y = "-2.0"/>
      <PolyStepSegment x = "-10.4" y = "0.0"/>
      <PolyStepCurve x = "10.4" y = "0.0" centerX = "0.0" centerY = "0.0" clockwise = “true”/>
      <PolyStepSegment x = "10.4" y = "-2.0"/>
      <PolyStepSegment x = "0.0" y = "-6.0"/>
      <PolyStepSegment x = "-10.4" y = "-2.0"/>
   </Polygon>
   <LineDesc lineEnd = "ROUND" lineWidth = "0.2"/>
</Outline>
```

3.4.8 PinRef

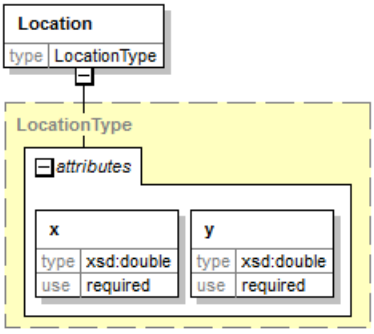
`PinRef` is an element that can be used to associate a layer feature, such as a pad or fiducial, to a component or an attribute of a component



PinRef	PinRefType	An individual `Pin` related to the place where a component attaches to the net. This description facilitates location of lands on the surface of the board or places where through-hole components are mounted.	1-n
componentRef	qualifiedNameType	The qualifiedNameType that identifies the reference designator used as the attribute `refDes` of the `Component` element in `Step` It is the reference to the component that is connected by the particular Pin and becomes a part of the electrical description of the net. The `componentRef` attribute is not required when used for a `PinRef` element under `Pad` elements which are under `LandPattern` elements. Otherwise, the `componentRef` attribute is required for `PinRef` elements under `Pad` and `LogicalNet` elements.	0-1
pin	string	An identification of the component pin, such as the pin number, that becomes a part of the electrical description.	1-1
title	string	An alternate method of relating the pin information providing characteristics of the component lead or termination description.	0-1

3.4.9 Location

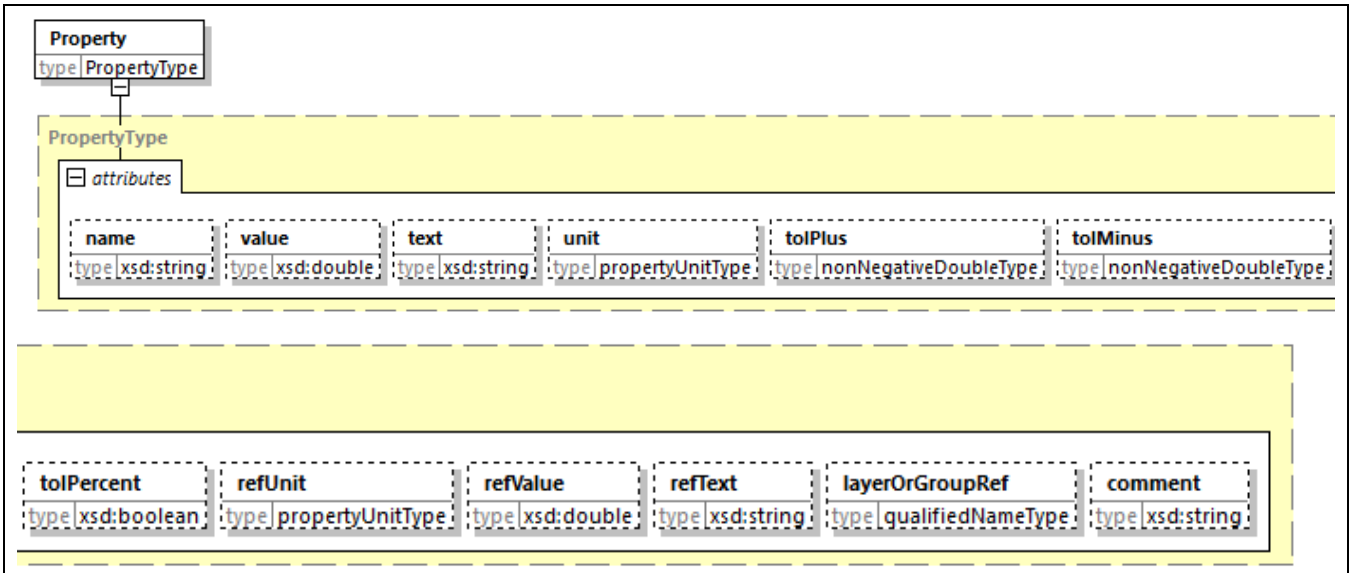
The `Location` element defines the location of an object in terms of the x and y coordinates of its origin relative to the CAD origin. The `CadHeader` element defines the units of measure.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Location	LocationType	The location of an object, with reference to its origin. Its origin may have been reorientated by the Xform.	1-1
x	double	The x coordinate of the object, in accordance with the Cartesian coordinate system. The `CadHeader` defines the units of measure.	1-1
y	double	The y coordinate of the object, in accordance with the Cartesian coordinate system. The `CadHeader` defines the units of measure.	1-1

3.4.10 Property

The Property element provides a group of attributes that may be used in order to describe the conditions that are to be associated with a particular specification. The attributes describe values, tolerances instructions and when they apply to a stackup, a reference is provided to which specific stackup layer the requirements apply. When the properties need to take into account other units or values a reference to those is also included. All attributes that are instantiated become a mandatory part of the specification to which they apply.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Property	PropertyType	A set of attributes that becomes a mandatory requirement to a specification and include values, tolerances, units and instructions. Appropriate references may also be provided. Since a specification may include several Property elements any ambiguity must be resolved prior to considering the impact the specification has on the products identified in the IPC-2581 file.	1-1
name	string	An optional name that can be given to the property being described. Only to be used for custom properties when the IPC-2581 file sender and receiver have a prior arrangement of what names are to be used.	0-1
value	double	A number that identifies the characteristics of the Value to be applied to any specified Unit attribute identified from the propertyUnitType enumerations.	0-1
text	string	Specific conditions or instructions that can be used as a property value when the value in non numeric or can be applied to the combination of Unit, Value and any tolerance attribute conditions.	0-1
unit	propertyUnitType	A set of enumerated string descriptions that identify the Unit properties of the Value attribute requirement. They consist of MM | INCH | MICRON | MILS | OHMS | MHO/CM | SIEMENS/M | CELCIUS | FARANHEIT | PERCENT | Hz | DEGREES | RMAX | RZ | RMS | SECTION | CLASS | ITEM | GAUGE | IN-LB | IN-OZ | FT-LB | N-m | N-cm | MIN | MAX | OZ | OZ/SQ-FT | GRAMS | HENRYS | AMPS | WATTS | VOLTS | FARAD | dB | dB/INCH | dB/MM | OTHER. If OTHER is noted a comment attribute is required.	0-1
tolPlus	nonNegativeDouble	The plus tolerance that may be applied to the nominal number defined in the Value attribute setting the upper control limit.	0-1
tolMinus	nonNegativeDouble	The minus tolerance that may be applied to the nominal number defined in the Value attribute setting the lower control limit.	0-1
tolPercent	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lower case). True means that the attribute(s) `tolPlus` and/or `tolMinus` are a percentage. False indicates that they represent a nonNegativeDouble. The Default is “false”.	0-1

refUnit	propertyUnitType	A second Unit set of enumerated string descriptions that are to be used as reference Unit properties of the basic attribute requirement. They consist of MM | INCH | MICRON | MILS | OHMS | MHO/CM | SIEMENS/M | CELCIUS | FARANHEIT | PERCENT | Hz | DEGREES | RMAX | RZ | RMS | SECTION | CLASS | ITEM | GAUGE | IN-LB | IN-OZ | FT-LB | N-m | N-cm | MIN | MAX | OZ | OZ/SQ-FT | GRAMS | HENRYS | AMPS | WATTS | VOLTS | FARAD | dB | dB/INCH | dB/MM | OTHER. If OTHER is noted a comment attribute is required.	0-1
refValue	double	A second Value number that is to be used as reference Value in considering the properties of the basic attribute requirement.	0-1
refText	string	A second set of instructions that need to be applied when considering the references established for the Value number, unit, tolerances or instructions applied to the properties of the basic attribute requirement.	0-1
layerOrGroupRef	qualifiedNameType	A reference to the identification of the unique name assigned to a specific layer or group to which the properties of the specification are to be applied.	0-1
comment	string	A description of any unit or refUnit characteristic that is different or supplements the propertyUnitType enumerations	0-1

3.5 Substitution Groups

The IPC-2581 uses the concept of substitution within the XML schema. Various groups of elements have been identified in the body of the standard and have been designated as having a specific focus or purpose. Within the schema, these substitution groups are provided with a name. When a group exists and if they are required according to the instances of the schema, it is mandatory that the substitution name be replaced by one of the acceptable descriptions identified within the group.

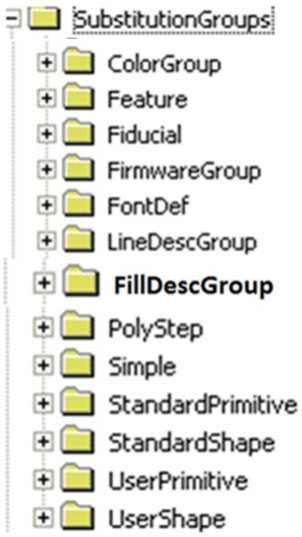
Often a schema needs to specify that one of several different XML Elements can be used with equal validity. For example, in every case where a `Triangle` can be used, it is also permissible to use a `Diamond`, `Hexagon`, `Octagon`, `Oval`, or one of several others: even though these shapes are quite different, they are equivalent as far as the schema is concerned. IPC-2581 handles this by using “substitution groups.”

A substitution group consists of two types of elements: a “head” and elements which may substitute for the head. Furthermore, when the head is denoted as ABSTRACT, the substitution is required, rather than optional. In IPC-2581, the heads of all substitution groups are ABSTRACT. Thus, it means that a valid instance document is not allowed to contain a `StandardPrimitive` element, but instead, (where `StandardPrimitive` is called for in the schema) a `Triangle`, `Diamond`, `Hexagon`, etc. must be used.

It should be noted that the head of one substitution group may be used within a different substitution group. As an example, the `StandardPrimitive` element is part of the `StandardShape` substitution group, which in turn is part of the `Feature` substitution group. This means that a `Triangle`, `Diamond`, `Hexagon`, etc. may be used wherever a `Feature` or `StandardShape` is called for, as well as wherever a `StandardPrimitive` is called for.

IPC-2581 features several dictionaries that permit specifying some type of information (such as a `StandardPrimitive` or a `LineDesc`) one time, and to reuse that definition as often as necessary. Some substitution groups in IPC-2581 are present to enable specifying either a dictionary entry or the same kind of information defined in specific detail within the body of the file. Any predefined image contained in the Dictionaries must have a unique “id”. It is the “id” name that is used to instantiate information from any of the dictionaries.

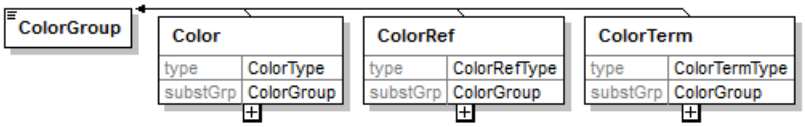
There are 13 substitution groups within the IPC-2581 schema. These are shown in the following table.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
ColorGroup	ABSTRACT	A substitution group that permits the substitution of the `Color` element when it is a child of the parent `Set`, or `Text` Elements.	3
Feature	ABSTRACT	A substitution group that permits the substitution of the `Feature` element when it is a child of the parent `Set` element.	1
Fiducial	ABSTRACT	A substitution group that permits the substitution of the `Fiducial` element when it is a child of the parent `Set` element.	1
FirmwareGroup	ABSTRACT	A substitution group that permits the substitution of the `FirmwareGroup` element when it is a child of the parent `Firmware` element.	1
FontDef	ABSTRACT	A substitution group that permits the substitution of the `FontDef` element when it is a child of the parent `EntryFont` element.	1
LineDescGroup	ABSTRACT	A substitution group that permits the substitution of the `LineDescGroup` element when it is a child of the parent of most `StandardPrimitive, Outline, Polyline,` or `Set` elements.	16
FillDescGroup	ABSTRACT	A substitution group that permits the substitution of the `FillDescGroup` element when it is a child of the parent of most `StandardPrimitive, Outline, Polyline,` or `Set` elements.	16
PolyStep	ABSTRACT	A substitution group that permits the substitution of the `PolyStep` element when it is a child of the parent `Polyline` or `Polygon` elements.	2
Simple	ABSTRACT	A substitution group that permits the substitution of the `Simple` element when it is a child of the parent `DfxMeasurement`, `Glyph`, or `Slot` elements.	3
StandardPrimitive	ABSTRACT	A substitution group that permits the substitution of the `StandardPrimitive` element when it is a child of the parent `EntryStandard` element.	1
StandardShape	ABSTRACT	A substitution group that permits the substitution of the `StandardShape` element when it is a child of the parent `Pad` element.	2
UserPrimitive	ABSTRACT	A substitution group that permits the substitution of the `UserPrimitive` element when it is a child of the parent `EntryUser` element.	1
UserShape	ABSTRACT	A substitution group that permits the substitution or classification of a higher level substitution group. The `UserShape` element may be used to further classify `Feature`. In so doing, `UserShape` can be substituted by a `UserPrimitive` or `UserPrimitiveRef`.	0

3.5.1 ColorGroup

The `ColorGroup` substitution group consists of three elements, `Color`, `ColorRef` and `ColorTerm`, with which it is possible to define a color in the body of the IPC-2581 standard. `ColorGroup` is used by the `FillDesc`, `Set`, `Layerfeature`, `Text`, and `General` (specification) elements.



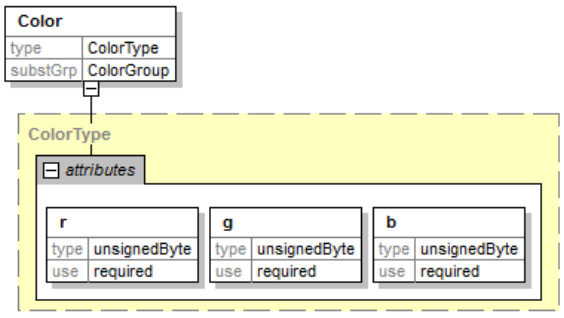
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Color	ColorType	The description of a specific color using the three attributes of red, green, and blue.	0-n
ColorRef	ColorRefType	The id of a previously defined 'rgb' color stored in the DictionaryColor.	0-n
ColorTerm	ColorTermType	A color name which is one of an enumerated list of basic color terms.	0-n

There is no default color. If no color is defined anywhere within the XML hierarchy of a given object then the implementing system's color setting shall be used. If different colors are set at different levels within the XML hierarchy of a given object then the value set at the lowest level shall take precedence. For example in the XML hierarchy below the color defined at the lower level, green, shall be used for the Circle.

```
<Set>
  <Color r="255" g="0" b="0"/>
  <Pad>
    <Location x="277.6" y="0"/>
    <Circle diameter=”1.0>
      <FillDesc fillProperty=”FILL”>
          <Color r="0" g="255" b="0"/>
      </FillDesc>
  </Pad>
</Set>
```

3.5.1.1 Color

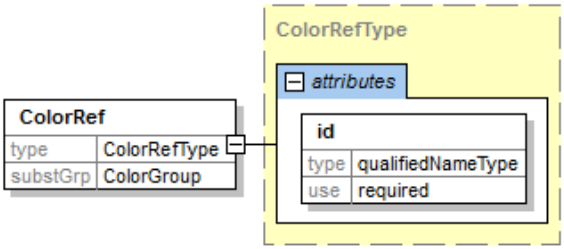
`Color` is defined by three values that represent the red, green and blue components of the composite color. If r, g, and b are all set to 0, the color is black. If all values are 255 then the color is white. The attributes of a `Color` element are defined as follows:



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Color	ColorType	The specific description of the color represented by the three attributes.	1-1
r	unsignedByte	Defines the red color intensity as a value between 0 and 255.	1-1
g	unsignedByte	Defines the green color intensity as a value between 0 and 255.	1-1
b	unsignedByte	Defines the blue color intensity as a value between 0 and 255.	1-1

3.5.1.2 ColorRef

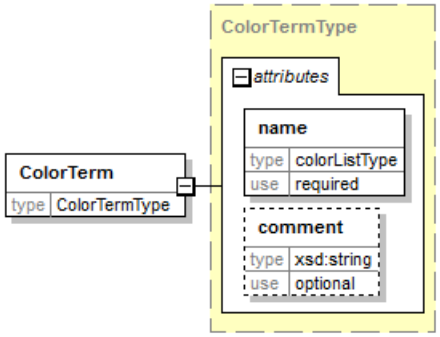
The `ColorRef` element is used throughout the IPC-2581 file to establish the relationship to a previously defined `Color`. The `ColorRef` definition is according to the following characteristics.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
ColorRef	ColorRefType	The specific reference to a predefined color represented by the three attributes contained in a `DictionaryColor` (see section **Error! Reference source not found.**)	1-1
id	qualifiedNameType	The qualified description name assigned as an `id` standard for XML schema color substitution.	1-1

3.5.1.3 ColorTerm

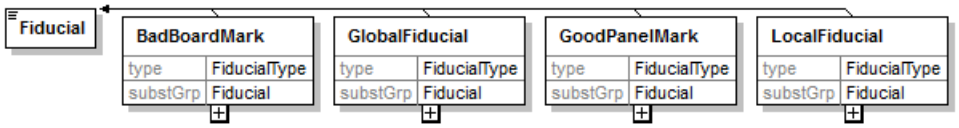
The `ColorTerm` element is used to describe a color as one of 11 basic color terms, from an enumerated list. It can be used by a customer to describe to a supplier the required color of a material, or physical object, when the precise hue of the color can be left up to the supplier. A 12th enumerated list member OTHER can be used if an additional color term is required, in which case a description must be given in the `comment` attribute. The `ColorTerm` definition is according to the following characteristics.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
ColorTerm	ColorTermType	An element to describe a color as one of an enumerated lit of basic color terms	1-1
name	colorListType	An enumerated list consisting of basic color terms. The list consists of BLACK, WHITE, RED, GREEN, YELLOW, BLUE, BROWN, ORANGE, PINK, PURPLE, GRAY, OTHER.	1-1
comment	string	Required if OTHER is chosen as the name, or can be used to describe any other information about the color requirement	0-1

3.5.2 Fiducial

The `Fiducial` substitution group consists of four elements that may be used to replace the `Fiducial` element. When the `Fiducial` element is substituted, it **shall** be by either a `BadBoardMark`, `GlobalFiducial`, `GoodPanelMark`, or `LocalFiducial` fiducial type. The `Fiducial` elements contain an Xform and a substitution capability to a `StandardShape`. An optional `Pin` attribute allows reference to a component pin.

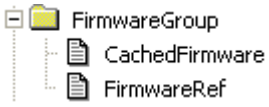


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
BadBoardMark	FiducialType	A set of Standard Shapes used as an aid to the board assembler by defining those boards in a panel that did not pass inspection or electrical test. The fiducial shape is positioned near each board in the assembly panel array and covered over to signify that the board is defective.	0-n
GlobalFiducial	FiducialType	A set of Standard Shapes used in the description, arrangement or positioning of a group of features on an individual board, assembly, or panel.	0-n
GoodPanelMark	FiducialType	A single Standard Shape used to define a panel where all boards on the panel are good. The fiducial is positioned once on the panel and enables reduction in inspection time.	0-n
LocalFiducial	FiducialType	A set of fiducials (usually a pair) used in the description and arrangement of features related to a specific component on a board, assembly, or panel which aide in the location/positioning process.	0-n

For further details refer to section 8.2.3.10.4 Fiducial.

3.5.3 FirmwareGroup

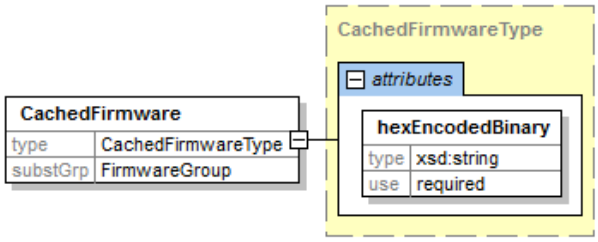
The `FirmwareGroup` substitution group consists of the description element for the firmware that defines the data to be added to a component through the `RefDes` element of a particular `BomItem`. The information may be provided as a `CachedFirmware` element or as a reference to the firmware which has been stored and identified with an “id” in the `DictionaryFirmware`.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
CachedFirmware	CachedFirmwareType	An element that identifies the firmware needed for a particular component in the Bill of Material. The information is in a `hexEncodeBinary` format.	0-n
FirmwareRef	FirmwareType	An element that references `CachedFirmware` stored in the `DictionaryFirmware` through the callout of the firmware “id”	0-n

3.5.3.1 CachedFirmware

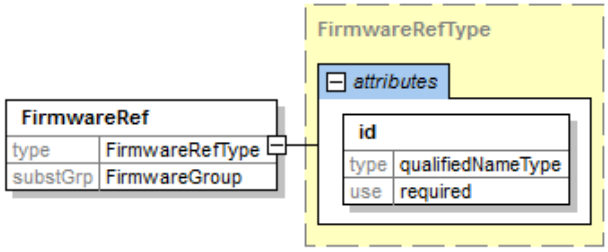
The `CachedFirmware` element is used to describe firmware that will be contained in the `DictionaryFirmware`. Refer to section 0 DictionaryFirmware. The details are in accordance to the following characteristics.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
CachedFirmware	CashedFirmwareType	The firmware description needed by a particular component that becomes part of the predefined firmware in the `DictionaryFirmware`.	1-1
hexEncodedBinary	string	An attribute defining the binary code that **shall** be added to a particular component and which is contained in the `DictionaryFirmware`.	1-1

3.5.3.2 FirmwareRef

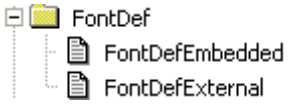
The `FirmwareRef` element is used throughout the IPC-2581 file to establish the relationship to a previously defined `CashedFirmware`. The `FirmwareRef` definition is according to the following characteristics.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
FirmwareRef	FirmwareRefType	The specific reference to firmware previously defined and contained in the `DictionaryFirmware`.	1-1
id	qualifiedNameType	The qualified name of `CachedFirmware` contained in the `DictionaryFirmware`.	1-1

3.5.4 FontDef

The `FontDef` substitution group consists of the description of a font that is different than the standard Helvetica and which is contained in the `DictionaryFont`. Fonts in the dictionary have an “id” which is called out when a `FontRef` is instantiated. `FontRef` is used by the element `Text`, which is called for in `SilkScreen` and `AssemblyDrawing`. `Text` can also be substituted whenever a `UserPrimitive`, `UserShape` or `Feature` is called for.

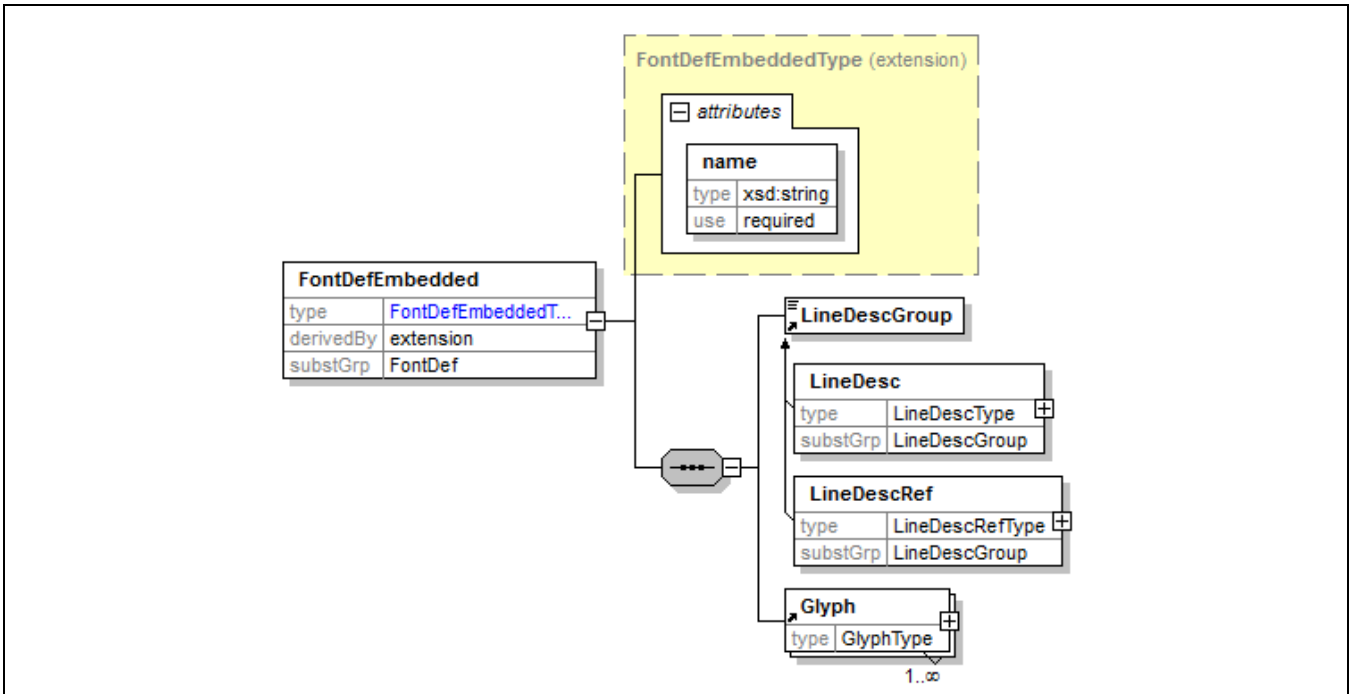


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
FontDefEmbedded	FontDefEmbedded Type	A description of a font using individual characters that are defined in accordance with the `Glyph` element and are contained in the `DictionaryFont`. Fonts must be predefined and may not be instantiated as part of the data in the file.	0-n
FontDefExternal	FontDefExternal Type	A reference to an external font description through the instantiation of a URN. The font is named and the reference is contained in the `DictionaryFont`. The appropriate character set is defined by the URN. External Fonts must be predefined and may not be instantiated as part of the data in the file.	0-n

The term "Uniform Resource Name" (URN) refers to the subset of URI that are required to remain globally unique and persistent even when the resource ceases to exist or becomes unavailable. A URI can be further classified as a locator, a name, or both. The term "Uniform Resource Locator" (URL) refers to the subset of URI that identify resources via a representation of their primary access mechanism (i.e., their network "location"), rather than identifying the resource by name or by some other attribute(s) of that resource.

3.5.4.1 FontDefEmbedded

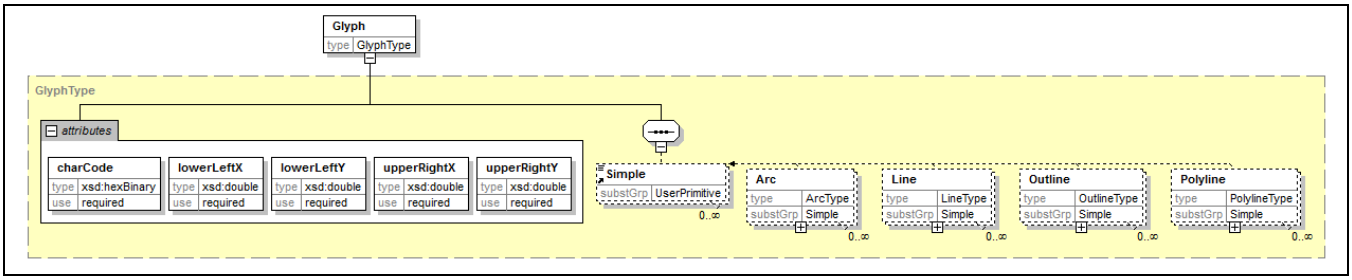
The `FontDefEmbedded` element is used to capture individual characters and store them in the `DictionaryFont`. Refer to section **Error! Reference source not found.** DictionaryFont.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
FontDefEmbedded	FontDefEmbeddedType	A substitution for `FontDef` that identifies an individual `Glyph` character by a specific name and the `Glyph` characteristics.	0-n
name	string	A unique name related to the `charCode` of the `Glyph` character.	0-n
LineDescGroup	ABSTRACT	A substitution group that specifies the `LineWidth` and `LineEnd` characteristics of the `Font` by reference to a predefined `LineDesc` or specified when the `font` is instantiated.	1-1
Glyph	GlyphType	The element that contains the description of all the character definitions defined in the user developed font	1-n

3.5.4.1.1 Glyph

The `Glyph` character set is a group of user defined characters that will be reference by the text command in the file. `Glyph` permits the user to define a special set of characters that need description as a part of the IPC-2581 file. Each character is in a `BoundingBox` that contains all the line strokes needed to completely define each character in the set. The point of origin is the lower left hand corner of the `BoundingBox`. The `lowerLeftX` and `lowerLeftY` point of origin will be used to position, rotate or mirror image all `Glyph` characters.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Glyph	GlyphType	The element that contains the description of all the character definitions defined in the user developed font.	0-n
charCode	hexBinary	A code used by the user to identify a special character.	0-1
lowerLeftX	double	The lower left hand x dimension of the rectangular area encompassing the character.	1-1
lowerLeftY	double	The lower left hand y dimension of the rectangular area encompassing the character.	1-1
upperRightX	double	The upper right hand x dimension of the rectangular area encompassing the character.	1-1
upperRightY	double	The upper right hand y dimension of the rectangular area encompassing the text character.	1-1
Simple	ABSTRACT	A substitution set of simple primitive shapes that may be any one of four geometries: `Arc, Line, Outline`, or `PolyLine`. The `LineWidth` and `LineEnd` characteristics are established by the Simple substitution	0-n

3.5.4.1.2 Glyph combination

The developers of individual `Glyph` characters are encouraged to consider the manner in which the characters will be used. Since the `BoundingBox` surrounding the character must entirely encompass the `Glyph`, it is important to leave room in the `BoundingBox` so that the spacing between characters is consistent with the character style.

Using individual `Glyph` characters does not present a problem, however `Glyph` combinations should match the style of `Glyph` chosen by the user. Since it is mostly the `Text` element that instantiates fonts, the `Glyph BoundingBox` must fit into the `Text BoundingBox`. This is a simple strategy when all the `Glyph` characters are of a similar height. In this instance the “Y” dimensional differences between `Glyph` characters bounding boxes and `Text` bounding boxes should be identical in order to keep the `Glyph` characters within the `Text` box. Under those circumstances, only the spacing between characters needs to be considered.

As an example consider the word simple instantiated in capital letters or lower case. When instantiating a `Text` string, the `Glyph` for “SIMPLE” would only require equal bounding boxes in the character height even though the character “I” would have a smaller character width than the character “M”. A different strategy for `Glyph` development must be used if the `Text` string were to call for “Simple”. Since character height is different, it is recommended that the `Glyph BoundingBox` consider its location position in a `Text BoundingBox` according to the rule that all characters must be inside the `Text` box.

Figure 5 shows and example of `Glyph` bounding boxes related to the `Text BoundingBox`. The characters line up even though they are positioned on the lower y-coordinate. They were designed along a construction line to have this condition occur.

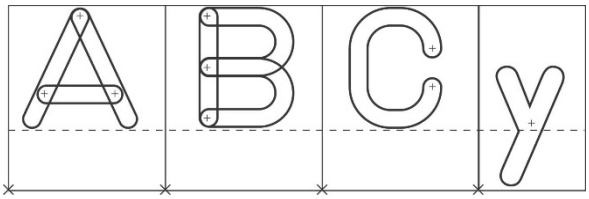
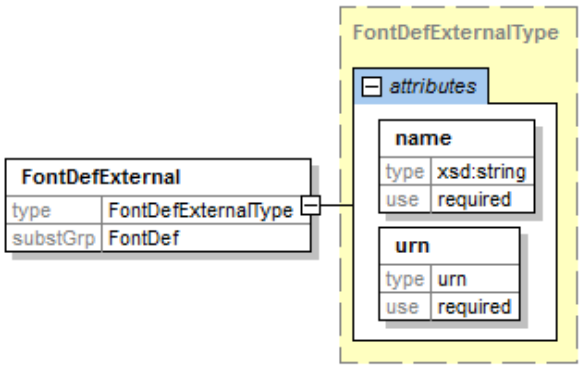


Figure 5 - Glyph bounding rectangles to Text bounding box relationships

3.5.4.2 FontDefExternal

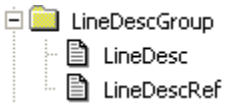
The `FontDefExternal` element is used to capture known font characters and store the reference in the `DictionaryFont`.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
FontDefExternal	FontDefExternalType	A substitution for `FontDef` that identifies a known group of characters through their font style.	0-n
name	string	A unique name related to the character set of a known font.	0-n
urn	urn	A specific `urn` that makes reference to a known font that has the appropriate permission to use the font substitution in a IPC-2581 file.	1-1

3.5.5 LineDescGroup

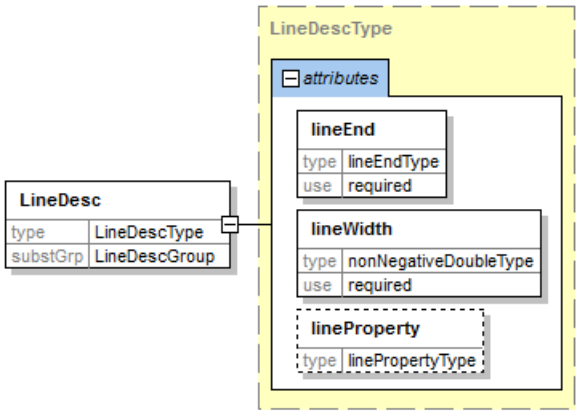
The `LineDescGroup` substitution specifies the `LineDesc or LineDescRef`. The LineDesc specifies the `LineWidth, LineEnd and LineProperty` characteristics of any feature that requires that definition. Line descriptions are a part of the `Outline`, `Polyline` and `Set` element definitions. The substitution is also instantiated by the substitution group `Simple` which calls for `Arc`, `Line`, `Outline` and `Polyline`. In addition to `UserPrimitive`, `StandardPrimitive` also makes use of the line descriptions when required. The `LineDescRef` requires an "id" that must be unique within the IPC-2581 file.



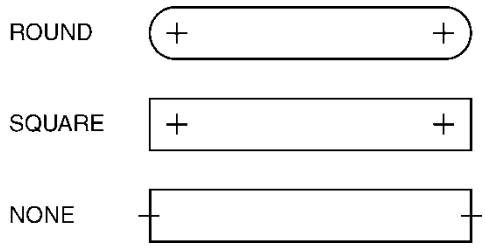
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
LineDesc	LineDescType	An element that identifies the `LineEnd`, `LineWidth and LineProperty` characteristics	0-n
LineDescRef	LineDescRefType	A reference to a `LineDesc` that is contained in the `DictionaryLineDesc` and identified by its unique "id". The units for the dictionary are defined and must be consistent with the units of the `CadHeader` when referenced from the `Ecad` section.	0-n

3.5.5.1 LineDesc

The `LineDesc` element is used throughout the IPC-2581 file to establish the characteristics of `lineEnd lineWidth,` and `lineProperty` descriptions. The `LineDesc` definition is according to the following characteristics.

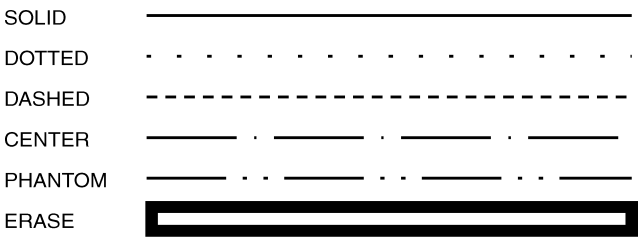


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
LineDesc	LineDescType	An element that defines the characteristics of a line. The `Units` are defined when the `LineDesc` is instantiated; Units is a part of the characteristics when `LineDesc` is contained in the `DictionaryLineDesc.`	1-n
lineEnd	lineEndType	A description of the line ends used in any graphic description. The lineEnd description is an enumerated string that may be ROUND | SQUARE | NONE. The default is ROUND.	1-1



This diagram illustrates the `LineDesc/@lineEnd` attribute

lineWidth	nonNegativeDouble Type	A dimensional characteristic that defines the nominal `lineWidth` of a `Line`, `Polyline`, or `Arc` element. The dimensions are in the same category as all dimensions contained in the IPC-2581 file.	1-1
lineProperty	linePropertyType	The line property is one of SOLID, DOTTED, DASHED, CENTER, PHANTOM or (solid) ERASE. The default is SOLID.	0-1

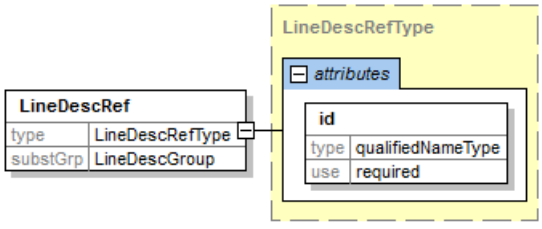


This diagram illustrates the `lineProperty` attribute. lineProperty attributes require the following rules to define width, end, and spacing characteristics for the line segments, dashes and dots.

SOLID Lines	Requires lineWidth with lineEnd default as ROUND
DOTTED Lines	Dot Diameter 1X lineWidth; Space is 2X lineWidth as the default
DASHED Lines	Dash Length 3X lineWidth; space 3X line Width as the default
CENTER Lines	Length 6X lineWidth; space is 2X lineWidth; Dot =1X lineWidth
PHANTOM Lines	Length 6X lineWidth; Space is 2X lineWidth; Dots =1X lineWidth
ERASE Lines	Requires lineWidth with lineEnd default as NONE

3.5.5.2 LineDescRef

The `LineDescRef` element is used throughout the IPC-2581 file to establish the relationship to a previously defined `LineDesc`. The `Units` of the predefined `LineDesc` must match the `Units` of the `Ecad` section in which it is instantiated. The `LineDescRef` definition is according to the following characteristics.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
LineDescRef	LineDescRefType	An element used to reference a previously defined `LineDesc`, contained in the `DictionaryLineDesc`, described in section 4.7	1-n
id	qualifiedNameType	The identification of the `LineDesc` being referenced from the `DictionaryLineDesc`.	1-1

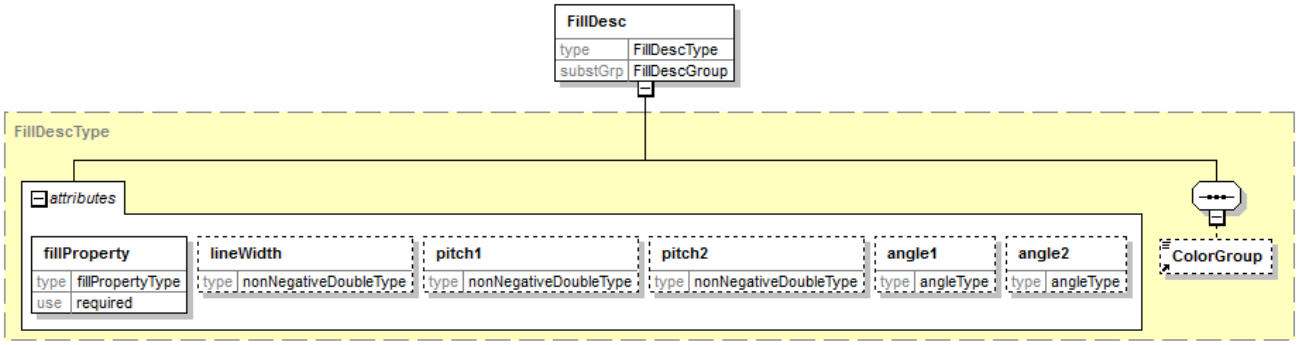
3.5.6 FillDescGroup

The `FillDescGroup` substitution specifies either `FillDesc` or `FillDescRef`. The `FillDesc` element provides the `FillProperties, LineDesc` and `Color` characteristics of any feature that requires that definition. Fill descriptions are a part of the `StandardPrimitives`, any `UserPrimitive` closed shape and `Set` element definitions. The `FillDescRef` requires an "id" that must be unique within the IPC-2581 file

Attribute / Element Name	Attribute / Element Type	Description	Occurrence
FillDesc	FillDescType	An element that identifies the `FillProperties, LineDesc and Color` characteristics	0-n
FillDescRef	FillDescRefType	A reference to a `FillDesc` that is contained in the `DictionaryFillDesc` and identified by its unique "id". The units for the dictionary are defined and must be consistent with the units of the `CadHeader` when referenced from the `Ecad` section.	0-n

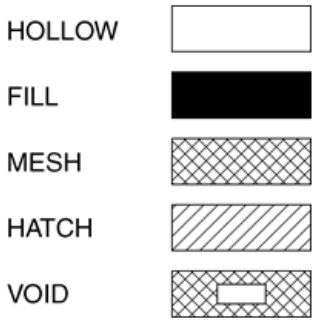
3.5.6.1 FillDesc

The `FillDesc` element is used throughout the IPC-2581 file to establish the characteristics of `FillProperty` and `Color` descriptions. The `FillDesc` definition is according to the following characteristics.



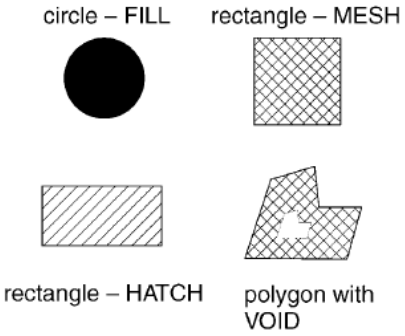
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
FillDesc	FillDescType	An element that defines the characteristics of a fill property. The `Units` are defined when the `LineDesc` is instantiated; Units is a part of the characteristics when `FillDesc` is contained in the `DictionaryFillDesc`.	1-1
fillProperty	fillPropertyType	The texture of the fill is one of HOLLOW, HATCH, MESH, FILL or VOID. The value VOID is reserved for Cutout elements – refer to section 3.4.5 Cutout. The default is FILL.	1-1
lineWidth	nonNegativeDouble Type	The width of the lines used in a HATCH or MESH. (This attribute only applies to the fill types HATCH and MESH.)	0-1
pitch1	nonNegativeDouble Type	The distance between the first set of lines in a HATCH or MESH. The default pitch between the hatch lines is 4 times the `lineWidth`. (This attribute only applies to the fill types HATCH and MESH.)	0-1
pitch2	nonNegativeDouble Type	The distance between the second set of lines in a MESH. The default pitch between the hatch lines is 4 times the `lineWidth`. (This attribute only applies to the fill type MESH.)	0-1

angle1	angleType	The angle of a set of lines in a HATCH or a MESH. The angle is measured relative to the x-axis of the local coordinate system. The range of values is limited to between 0 and 180 degrees. The default is 45 degrees. (This attribute only applies to the fill types HATCH and MESH.)	0-1
angle2	angleType	The angle of the second set of lines in a MESH. The angle is measured relative to the x-axis of the local coordinate system. The angle must be between 90 and 180 degrees. The default is 135 degrees. (This attribute only applies to the fill type MESH.)	0-1
ColorGroup	ABSTRACT	An element that is a substitution group which can be used to define a particular `Color` for the shape being filled. In the absence of this element the default color is black, but this can be overridden by a color set at the parent element, or above, level.	0-1



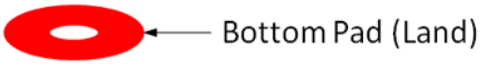
This diagram illustrates the `FillDesc` type Attribute.

The `LineDesc lineProperty` is set to SOLID in the example shapes to provide an outline around the shapes that are painted with MESH, HATCH, or VOID. The VOID rectangle feature is within a previously defined rectangle that has been defined with HATCH lines. As shown all of the VOID characteristics are within the boundaries of the larger HATCH rectangle feature



This diagram illustrates the `FillDesc` type Attribute

The `LineDesc lineProperty` is set to SOLID in the example shapes to provide an outline around the shapes that are painted with MESH, HATCH, or VOID. The VOID within the polygon is completely contained within its boundaries as required.



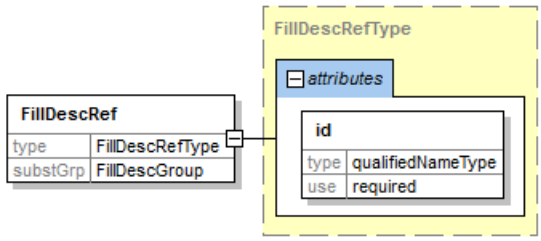
The example may be defined as a UserPrimitive using a Circle fill description with a circular VOID in its center. The User Primitive name may be used in the padstack descriptions.

The fillProperty VOID means the element shape clears an area with respect to contours. Elements (and references to elements) with fillProperty VOID can appear only as elements in a Set element or a UserSpecial element shape. Elements with fillProperty VOID apply only to the elements that appear before the VOID element in the Set element or in the UserSpecial element. Elements with fillProperty VOID only clear filled contours.

A contour is a polygon which is a closed shape whose edges do not cross. The edges of elements with fillProperty VOID do not cross other elements with fillProperty VOID and do not cross the edges of the owning contour element. The order of voids and polygons within Contours and Sets determines their containment order. The outermost polygon comes first. Polygons are before voids that are contained in them. Voids are before polygons that are contained inside them.

3.5.6.2 FillDescRef

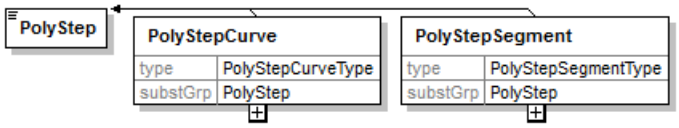
The `FillDescRef` element is used throughout the IPC-2581 file to establish the relationship to a previously defined `FillDesc`. The `Units` of the predefined `FillDesc` must match the `Units` of the `Ecad` section in which it is instantiated. The `FillDescRef` definition is according to the following characteristics.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
FillDescRef	FillDescRefType	An element used to reference a previously defined `FillDesc`, contained in the `DictionaryFillDesc`.	1-n
id	qualifiedNameType	The identification of the `FillDesc` being referenced from the `DictionaryFillDesc`.	1-1

3.5.7 PolyStep

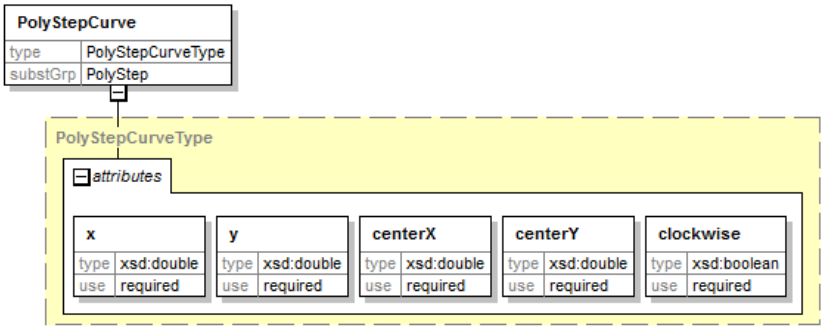
The `Polystep` substitution consists of defining either a `Line` or an `Arc` as the continuation of a `Polyline` or `Polygon` description. The location information is interpreted as being the end point to which the curve (`Arc`), or segment (`Line`) is drawn. Straight or curved line segments must not cross. The substitution may take place anywhere within the file where the elements `Polyline` and `Polygon` occur. This action includes the dictionaries where graphic descriptions are predefined. The `Units` of measure must be consistent with the `Units` parameter of the three dictionaries where this substitution can take place; `DictionaryStandard`, `DictionaryUser`, and `DictionaryFont`.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
PolyStepCurve	PolyStepCurveType	The continuation of the linear description of a `Polyline` or `Polygon` if the next portion to be defined is an arc. The end point of the arc is defined as well as the center of rotation	0-n
PolyStepSegment	PolyStepSegmentType	The continuation of the linear description of a `Polyline` or `Polygon` if the next portion to be defined is a line segment. The end point of the line is defined.	0-n

3.5.7.1 PolyStepCurve

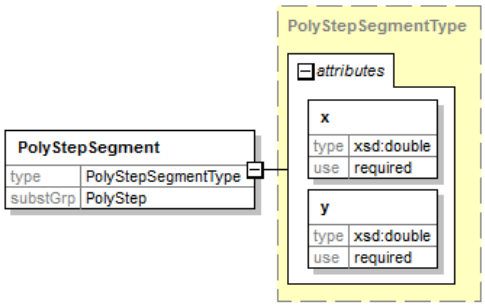
A `PolyStepCurve` is a continuation of the linear description of a `Polyline` or `Polygon` if the next portion to be defined is an arc. The end point of the arc is defined as well as the center of rotation. The start point of the arc is defined by either a `PolyBegin` element or the previous `PolyStep`.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
PolyStepCurve	PolyStepCurveType	The continuation of the linear description of a `Polyline` or `Polygon` if the next portion to be defined is an arc. The end point of the arc is defined as well as the center of location.	1-1
x	Double	The X end point of the curve.	1-1
y	Double	The Y end point of the curve.	1-1
centerX	Double	The X point that is the center of rotation of the curve.	1-1
centerY	Double	The Y point that is the center of rotation of the curve.	1-1
clockwise	Boolean	A required boolean to set the direction of rotation of the curve. If “true” then the direction of rotation is clockwise. If “false” then the direction of rotation is counterclockwise	1-1

3.5.7.2 PolyStepSegment

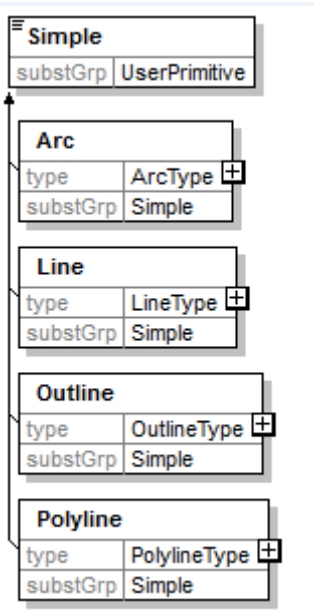
A `PolyStepSegment` is a continuation of the linear description of a `Polyline` or `Polygon` if the next portion to be defined is a line segment. It defines the end point of the line. The start point of the line is defined by either a `PolyBegin` element or the previous `PolyStep`.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
PolyStepSegment	PolyStepCurveType	The continuation of the linear description of a `Polyline` or `Polygon` if the next portion to be defined is an arc. The end point of the arc is defined as well as the location of the radius.	1-1
x	Double	The X end point of the line.	1-1
y	Double	The Y end point of the line.	1-1

3.5.8 Simple

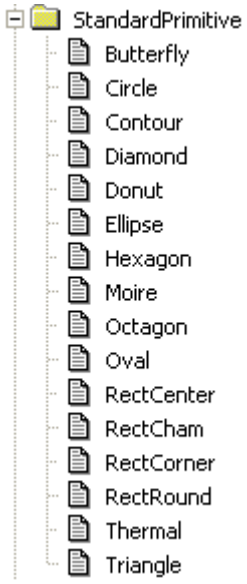
The `Simple` substitution consists of base elements `Arc`, `Line`, `Outline` or `Polyline`. The `Simple` substitution is called for in the `DfxMeasurement`, `Glyph`, and `Slot` elements. `Simple` is also identified as a `UserPrimitive`, `UserShape`, or `Feature` and the four elements may be substituted when called for in conjunction with those descriptions. When predefined in the `DictionaryUser`, or `DictionaryFont` the `Units` must match those of the dictionary.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Arc	ArcType	`Arc` elements are curves (defined by three sets of coordinates: `startX`, `startY`, `endX`, `endY` and `centerX`, `centerY`. The width of the arc is applied when the arc is instantiated or predefined.	0-n
Line	LineType	`Line` elements are individual line segments. The `lineWidth` and `lineEnd` conditions are defined when the line is instantiated or pre-defined. The `lineEnd` default is ROUND.	0-n
Outline	OutlineType	`Outline` has `Polygon` and `LineDescGroup` as children elements. The characteristics of the `Polygon` must be a closed shape. The point of origin of the shape is identified as the 0:0 coordinate. The element includes the start of the polygon definition (`PolyBegin`) and the appropriate number of PolySteps to complete the closed shape. The `lineWidth` is defined at a time when the `Outline` is instantiated or predefined.	0-n
Polyline	PolylineType	The `Polyline` element consists of a series of lines that define a particular grouping configuration. These line segments do NOT result in a closed shape, however they can be pre-defined and re-used as needed. The `lineWidth` and `lineEnd` of the `Polyline` are defined at the time the `Polyline` is instantiated or predefined.	0-n

3.5.9 StandardPrimitive

There are sixteen Standard Primitives defined in the IPC-2581 structure. Any of the primitive shapes may be a candidate for substitution when `StandardPrimitive` is called for in the schema. The names of the various shapes indicate their type; each has its attributes that identify the physical requirements. Any `StandardPrimitive` may be predefined, provided a unique "id", and contained in the `DictionaryStandard`. All `StandardPrimitive` shapes are developed in accordance with their description requirements in the preferred orientation of this standard.

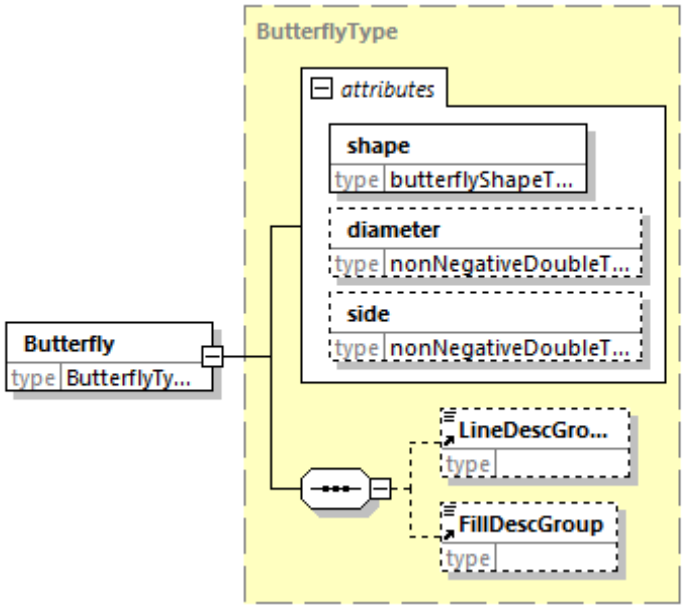


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Butterfly	ButterflyType	A primitive shape (either ROUND or SQUARE) that has two quadrants of the Cartesian coordinate system removed (0 to 90° and 180 to 270°).	0-n
Circle	CircleType	A primitive shape that defines a round object where the circular shape consists of a diameter.	0-n
Contour	ContourType	A closed primitive shape that has as its children a polygon and optional cutout(s) The sequence of connected edges that form the polygon may be straight or circular.	0-n
Diamond	DiamondType	A primitive shape with four equal sides that are extended from its horizontal center to its vertical center. The lines converge into a point both horizontally and vertically. The overall description of the shape is controlled by the width (distance between vertical point) and the height (distance between horizontal point).	0-n
Donut	DonutType	A round, square, hexagon, or octagon shaped primitive consisting of an outer diameter and inner diameter that define physical characteristics of the donut.	0-n
Ellipse	EllipseType	An elliptical primitive shape that follows the standard ellipse characteristics and is defined by a width and height dimension, establishing the overall limits of the feature.	0-n
Hexagon	HexagonType	A six-sided primitive shape with all sides being equal in length and with all angles between adjacent sides being equal. The orientation of the Hexagon is with one of its points facing North. Only the dimension across the points is required.	0-n
Moire	MoireType	A series of circles, each consisting of a smaller diameter than the previous. The details of the `Moire` is defined by the number of rings, their center line spacing, and the ring width. The pattern may also contain a crosshair representing its point of origin. Restrictions apply so that `ringWidth` is smaller than `ringGap`.	0-n
Octagon	OctagonType	An eight-sided primitive shape with all sides being equal in length and with all angles between adjacent sides being equal. The orientation of the `Octagon` is with one of its points facing north. Only the dimension across the points is required.	0-n

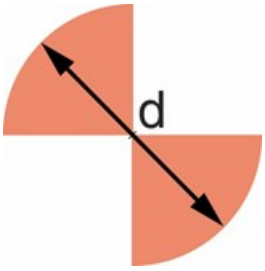
Oval	OvalType	A rectangular primitive shape with a complete radius (180° arc) at each end. The limits of the feature are controlled by the length and width of the oval across the outer extremities.	0-n
RectCenter	RectCenterType	The characteristics of a rectangle defined by a width and height dimension consistent with a horizontal position on the Cartesian coordinate system. The center point is the point of origin and is used to locate the `RectCenter`. A "`square`" is a `RectCenter` with the width and height equal.	0-n
RectCham	RectChamType	A rectangle with one or more corners chamfered. The user has the option to define any of the corners as containing the chamfer as well as the chamfered dimensions. All chamfers (or opportunities for chamfers) must be identical in size.	0-n
RectCorner	RectCornerType	A constraining rectangular area (bounding box) that describes a rectangle consistent with a horizontal position on the Cartesian coordinate system. The point of origin is the lower left corner. A Square positioned by its corners is a `RectCorner` that is defined by having the X and Y offset be equal.	0-n
RectRound	RectRoundType	A rectangle with one or more corners rounded. The user has the option to define any of the corners as containing the radius as well as the radiused dimensions. All corners (or opportunities for corners) must be identical in size.	0-n
Thermal	ThermalType	A primitive shape consisting either of ROUND, SQUARE, HEXAGON, or OCTAGON configuration that historically defines the removal of material from a plane or conductive fill area in accordance to the shape attribute description.	0-n
Triangle	TriangleType	A primitive isosceles triangular shape that has two equal sides and a base. The feature is defined by a base and height dimension.	0-n

3.5.9.1 Butterfly

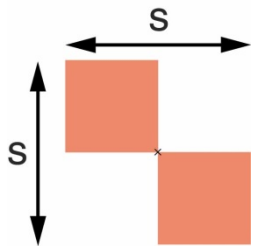
A `Butterfly` is a `StandardPrimitive` shape that may have the external periphery be either round or square with two quadrants of the Cartesian coordinate system removed (0 to 90° and 180 to 270°). The round shape is defined by its diameter; the square shape is defined by an equal side dimension. The `Butterfly` is positioned by its point of origin, which is at the center of the `Butterfly`.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Butterfly	ButterflyType	A primitive shape (either ROUND or SQUARE) that has two quadrants of the Cartesian coordinate system removed (0 to 90° and 180 to 270°).	1-1
shape	butterflyShapeType	The inner and outer shapes are one of ROUND or SQUARE: ROUND: The inner and outer shapes are like circleDef. SQUARE: The inner and outer shapes are like RectCenterDef with height and width of each shape being equal.	1-1
diameter	nonNegativeDoubleType	The diameter applies to the circular butterfly and is the outer boundary of the butterfly.	0-1
side	nonNegativeDoubleType	The width along the x-axis and the height along the y-axis of a square.	0-1
LineDescGroup	LineDescGroupType	A substitution group that specifies either `LineDesc` or `LineDescRef`. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either `FillDesc` or `FillDescRef`. Refer to section 3.5.6 FillDesGroup	0-1



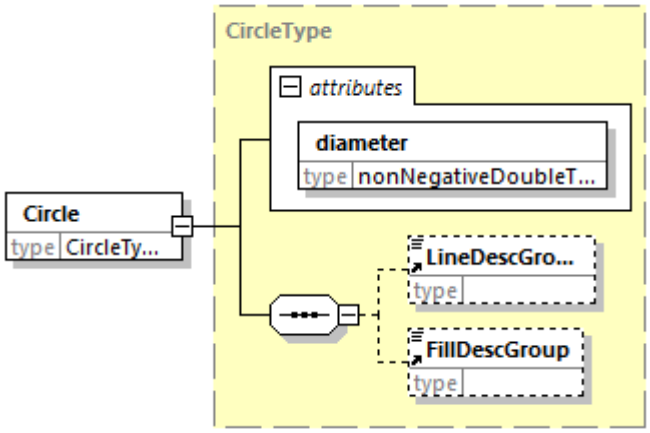

```
<Butterfly shape = "ROUND" diameter = "3.2">
        <FillDesc fillProperty="FILL">
                <Color r="150" g="0" b="0"/>
        </FillDesc>
</Butterfly>
```



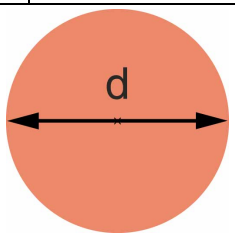

```
<Butterfly shape = "SQUARE" side = "1.8">
        <FillDesc fillProperty="FILL">
                <Color r="150" g="0" b="0"/>
        </FillDesc>
</Butterfly>
```

3.5.9.2 Circle

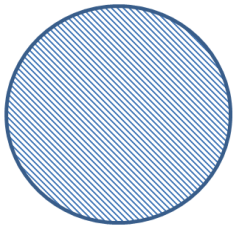
A `Circle` is a `StandardPrimitive` shape that defines a circle by the diameter of the circle. The point of origin is the center of a circle.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Circle	CircleType	An embedded element that defines a circular shape consisting of a diameter.	1-1
diameter	nonNegativeDoubleType	The diameter of the circle.	1-1
LineDescGroup	LineDescGroupType	A substitution group that specifies either LineDesc or LineDescRef. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either FillDesc or FillDescRef. Refer to section 3.5.6 FillDesGroup	0-1



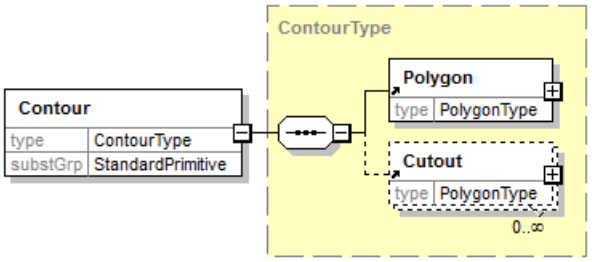

```
<Circle diameter = "3.6">
    <FillDesc fillProperty="FILL">
            <Color r="150" g="0" b="0"/>
    </FillDesc>
</Circle>
```



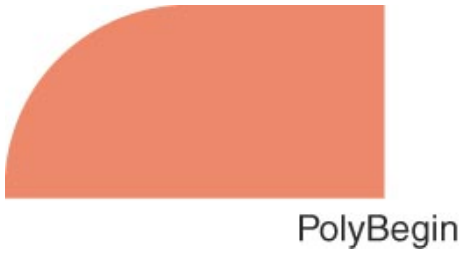
```
<Circle diameter = "3.6">
      <LineDesc lineEnd="ROUND" lineWidth="0.1"/>
      <FillDesc fillProperty="HATCH" lineWidth="0.02"
                pitch1="0.08" angle1="135">
            <Color r="0" g="0" b="200"/>
      </FillDesc>
</Circle>
```

3.5.9.3 Contour

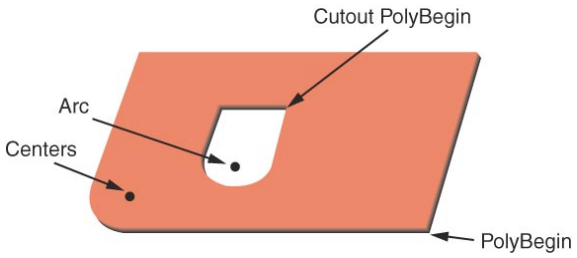
The `Contour` element is a `StandardPrimitive` shape that defines a sequence of connected edges that form a polygon. An edge can be straight or circular. The `polygon` is a closed shape whose edges do not cross. This same characteristic is also true for `cutout`, which represents the absence of material inside the polygon shape. The coordinates of the `polygon`, `cutout`, and subsequent cutouts are defined relative to the local coordinate system of the original `polygon`. The point of origin may be a centroid of the `polygon` or one of the corners that sets the 0/0 coordinate. This is the point used to place the `polygon` or to rotate the image. The `cutout` uses the same coordinates.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Contour	ContourType	A sequence of connected edges that form a polygon. An edge can be straight or circular.	1-1
Polygon	PolygonType	A closed shape whose edges do not cross, the coordinates of which are defined relative to the local coordinate system of the polygon.	1-1
Cutout	CutoutType	A polygon closed shape whose edges do not cross, which adopts the coordinates of the original polygon, however represents the absence of material within the original polygon shape.	0-n



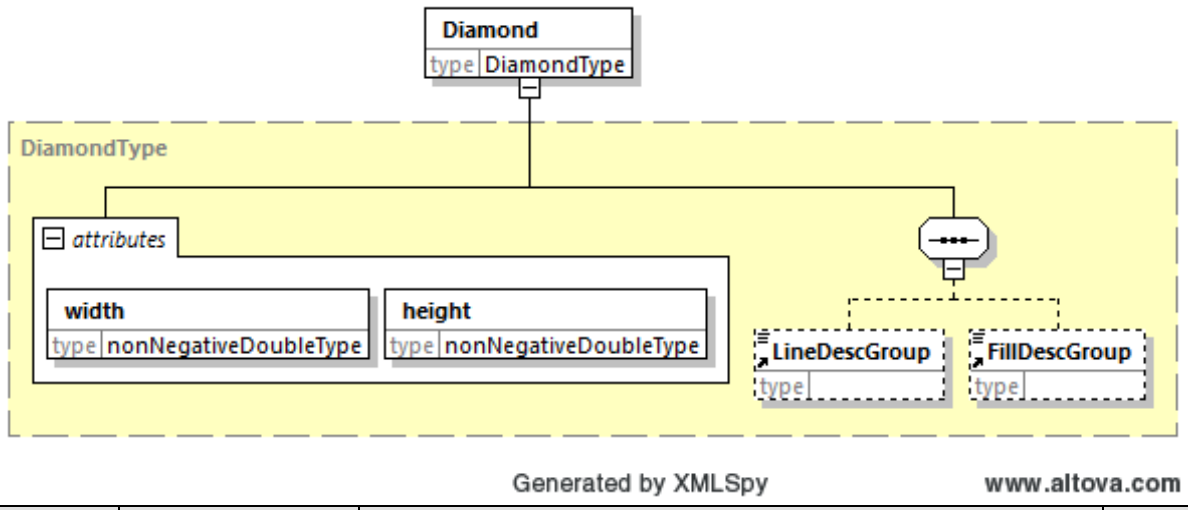

```
<Contour>
  <Polygon>
    <PolyBegin x = "0.00" y = "0.00"/>
    <PolyStepSegment x = "0.00" y = "7.00"/>
    <PolyStepSegment x = "-8.00" y = "7.00"/>
    <PolyStepCurve x = "-15.00" y = "0.00" centerX = "-8.00"
          centerY = "0.00" clockwise = “false”/>
    <PolyStepSegment x = "0.00" y = "0.00"/>
    <FillDesc fillProperty="FILL">
          <Color r="150" g="0" b="0"/>
    </FillDesc>
  </Polygon>
</Contour>
```



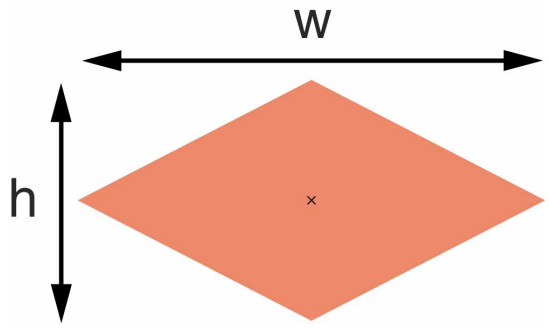

```
<Contour>
  <Polygon>
    <PolyBegin x = "0.00" y = "0.00"/>
    <PolyStepSegment x = "-14.00" y = "0.00"/>
    <PolyStepCurve x = "-3.00" y = "3.00" centerX = "-14.00"
          centerY = "3.00" clockwise = “true”/>
    <PolyStepSegment x = "-17.00" y = "7.00"/>
    <PolyStepSegment x = "0.00" y = "7.00"/>
    <PolyStepSegment x = "0.00" y = "0.00"/>
    <FillDesc fillProperty="FILL">
          <Color r="150" g="0" b="0"/>
    </FillDesc>
  </Polygon>
  <Cutout>
    <PolyBegin x = "-10.00" y = "5.00"/>
    <PolyStepSegment x = "-13.00" y = "5.00"/>
    <PolyStepSegment x = "-13.00" y = "3.00"/>
    <PolyStepCurve x = "-10.00" y = "3.00" centerX = "-11.50"
          centerY = "3.00" clockwise = “false”/>
    <PolyStepSegment x = "-10.00" y = "5.00"/>
    <FillDesc fillProperty="VOID"/>
  </Cutout>
</Contour>
```

3.5.9.4 Diamond

A `Diamond` is a 4-sided `StandardPrimitive` shape. The lengths of the sides of a diamond are always equal. A height and a width dimension specify the diamond. The first line defining the outline of the diamond is drawn between the point that is ½ the height dimension along the positive y-axis and the point that is ½ the width dimension along the x-axis. The same process is used to draw the other three lines of the diamond in each of the remaining quadrants. The `Diamond` is positioned with one of its corners facing the North direction.



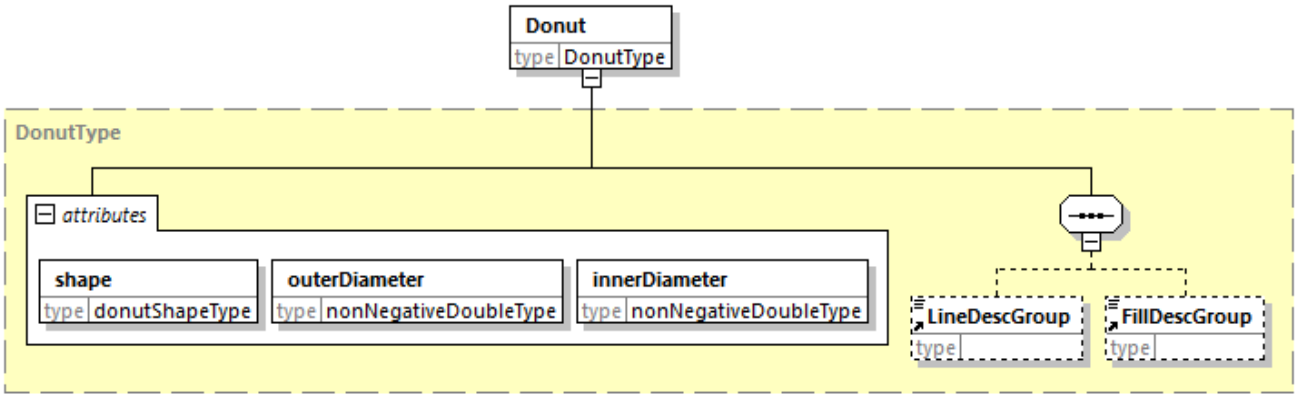
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Diamond	DiamondType	A primitive shape with four equal sides that are extended from its horizontal center to its vertical center. The lines converge into a point both horizontally and vertically. The overall description of the shape is controlled by the width (distance between vertical point) and the height (distance between horizontal point).	1-1
width	nonNegativeDouble Type	The length of the diamond along, and centered on, the x-axis.	1-1
height	nonNegativeDouble Type	The length of the diamond along, and centered on, the y-axis.	1-1
LineDescGroup	LineDescGroupType	A substitution group that specifies either `LineDesc or LineDescRef`. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either `FillDesc or FillDescRef`. Refer to section 3.5.6 FillDesGroup	0-1



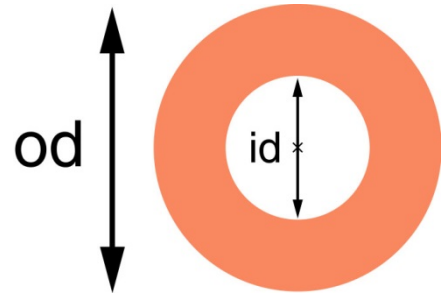
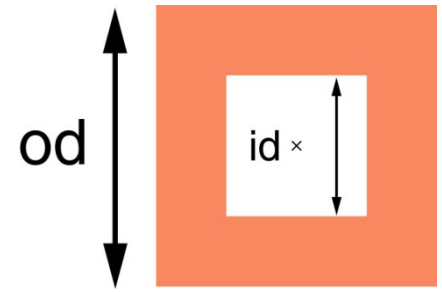
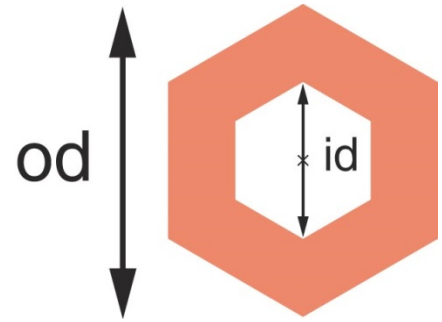
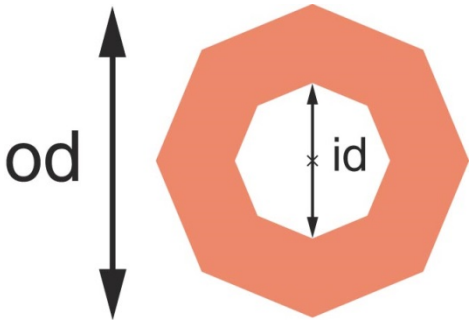

```
<Diamond width = "10.40" height = "6.20">
        <FillDesc fillProperty="FILL">
                <Color r="150" g="0" b="0"/>
        </FillDesc>
</Diamond>
```

3.5.9.5 Donut

A `Donut` is a `StandardPrimitive` shape composed of two concentric identical shapes. The shapes are the same but of different sizes with the outer diameter (OD) being larger than the inner diameter (ID). The shapes must be identical and may be square, round, hexagonal, or octagonal. The center of a `Donut` is also the point of origin of the primitive. The hexagonal and octagonal shapes are defined with a point of the shape facing the North direction.

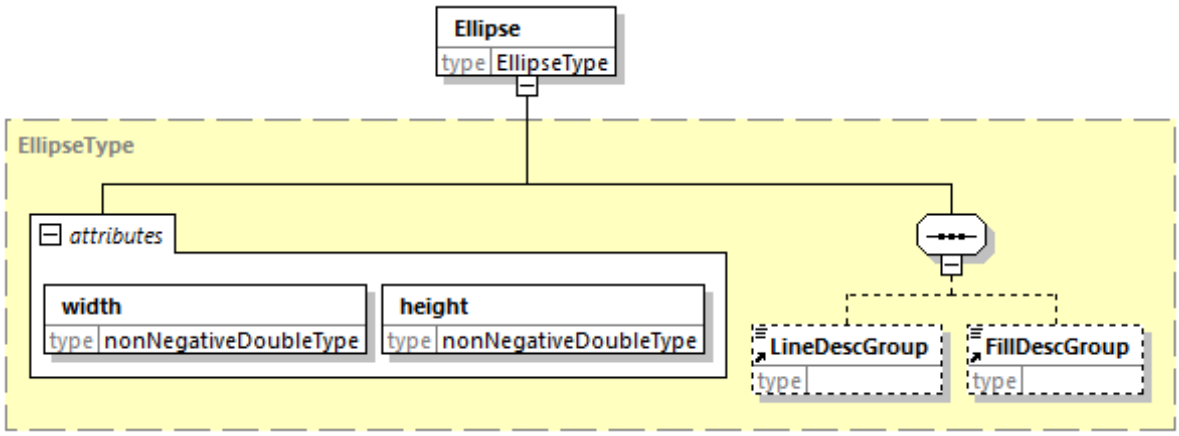


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Donut	DonutType	A round, square, hexagon, or octagon shape consisting of an outer diameter and inner diameter that define physical characteristics of the donut.	1-1
shape	donutShapeType	The inner and outer shapes are one of ROUND, SQUARE , HEXAGON or OCTAGON: ROUND – The inner and outer shapes are like a circle. SQUARE – The inner and outer shapes are like a RectCenter with height and width of each shape being equal. HEXAGON – The inner and outer shapes are like a Hexagon. OCTAGON – The inner and outer shapes are like an Octagon.	1-1
outerDiameter	nonNegativeDouble Type	The outer boundary of the filled region. The meaning based on donutShape: ROUND –The diameter of the circle is the outer boundary of the donut. The center of the circle is at the origin of the donut. SQUARE –The width along the x-axis and the height along the y-axis of a square at the inner boundary of the donut. The center of the square is at the origin. HEXAGON – The point-to-point measurement on the x-axis of the hexagon that forms the outer boundary of the donut. OCTAGON – The point-to-point measurement on the x-axis of the octagon that forms the outer boundary of the donut.	1-1
innerDiameter	nonNegativeDouble Type	The inner boundary of the filled region. The meaning based on donutShape : ROUND – The diameter of the circle is the inner boundary of the donut. The center of the circle is at the origin of the donut. SQUARE – The width along the x-axis and height along the y-axis of a square at the inner boundary of the donut. The center of the square is at the origin. HEXAGON – The point-to-point measurement on the x-axis of the hexagon that forms the inner boundary of the donut. OCTAGON – the point-to-point measurement on the x-axis of the octagon that forms the inner boundary of the donut.	1-1
LineDescGroup	LineDescGroupType	A substitution group that specifies either `LineDesc` or `LineDescRef`. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either `FillDesc` or `FillDescRef`. Refer to section 3.5.6 FillDesGroup	0-1

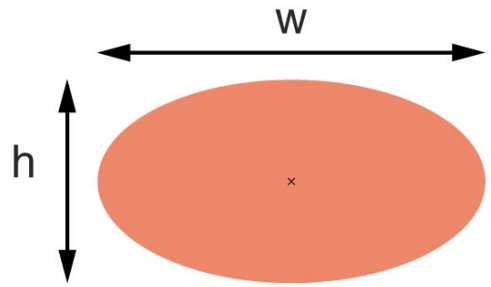
	<Donut shape = "ROUND" outerDiameter = "6.8" innerDiameter = "4.8"> <FillDesc fillProperty="FILL"> <Color r="150" g="0" b="0"/> </FillDesc> </Donut>
	<Donut shape = "SQUARE" outerDiameter = "6.8" innerDiameter = "5.0"> <FillDesc fillProperty="FILL"> <Color r="150" g="0" b="0"/> </FillDesc> </Donut>
	<Donut shape = "HEXAGON" outerDiameter = "12.4" innerDiameter = "10.2"> <FillDesc fillProperty="FILL"> <Color r="150" g="0" b="0"/> </FillDesc> </Donut>
	<Donut shape = "OCTAGON" outerDiameter = "11.60" innerDiameter = "10.00”> <FillDesc fillProperty="FILL"> <Color r="150" g="0" b="0"/> </FillDesc> </Donut>

3.5.9.6 Ellipse

The `Ellipse` is a `StandardPrimitive` shape that is an ellipse with the standard ellipse characteristics. The shape is defined by the width and height dimension. The `Ellipse` is positioned with its point of origin at the center of the `width` and `height` dimensions.



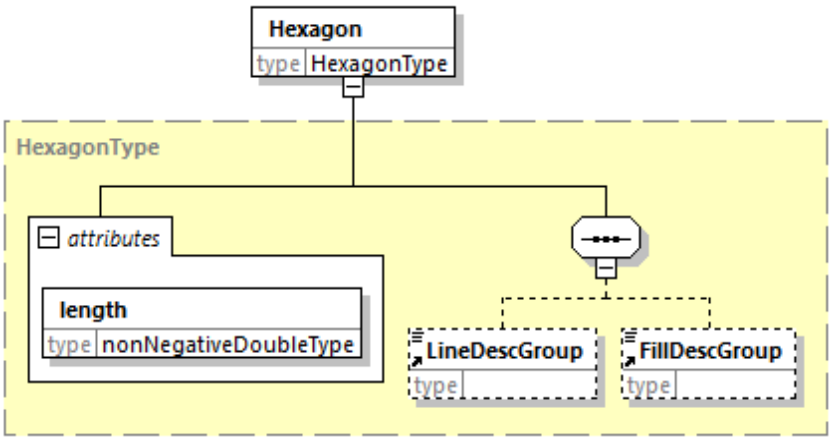
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Ellipse	EllipseType	An elliptical shape that follows the standard ellipse characteristics and is defined by a width and height dimension, establishing the overall limits of the feature.	1-1
width	nonNegativeDoubleType	The height of the ellipse on the y-axis.	1-1
height	nonNegativeDoubleType	The width of the ellipse on the x-axis.	1-1
LineDescGroup	LineDescGroupType	A substitution group that specifies either `LineDesc` or `LineDescRef`. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either `FillDesc` or `FillDescRef`. Refer to section 3.5.6 FillDesGroup	0-1



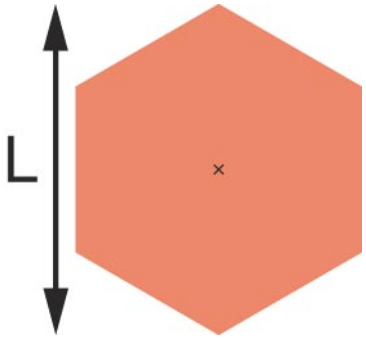

```
<Ellipse width = "12.60" height = "6.20">
        <FillDesc fillProperty="FILL">
                <Color r="150" g="0" b="0"/>
        </FillDesc>
</Ellipse>
```

3.5.9.7 Hexagon

A `Hexagon` is a six-sided `StandardPrimitive` shape with each of the sides being equal in length and with all angles between adjacent sides also being equal. The orientation of the hexagon is in accordance with one of its points facing the North direction. Only one dimension is required and that is the length across the points. `Rotation` is accomplished using `Xform` at the time the hexagon is instantiated.



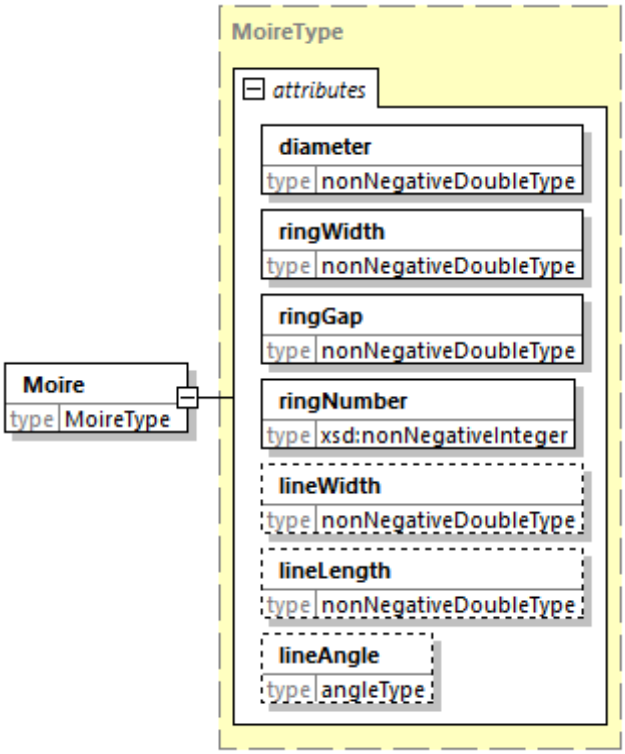
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Hexagon	HexagonType	A six-sided primitive shape with all sides being equal and which is defined by the length (L) across the points. The position of the octagon is in accordance with one of the points facing North.	1-1
length	nonNegativeDouble Type	The length (L) between any two opposing corner points of the hexagon.	1-1
LineDescGroup	LineDescGroupType	A substitution group that specifies either `LineDesc` or `LineDescRef`. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either `FillDesc` or `FillDescRef`. Refer to section 3.5.6 FillDesGroup	0-1



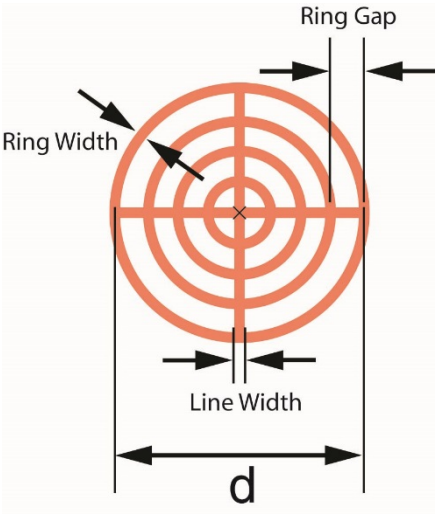

```
<Hexagon length = "12.8">
        <FillDesc fillProperty="FILL">
            <Color r="150" g="0" b="0"/>
        </FillDesc>
</Hexagon>
```

3.5.9.8 Moire

The `Moire` is a primitive shape that consists of a series of circles each with a smaller diameter. The `Moire` is used as an assist in image registration. The `Moire` may be only circles or may also contain a crosshair line to assist in human acknowledgement of moiré alignment. The shape is defined by the number of rings, their center line spacing and the ring line width. The line spacing must be larger than the line width. The crosshair lines can also be described. The `Moire` pattern is positioned using its point of origin which is the center of the ring pattern.



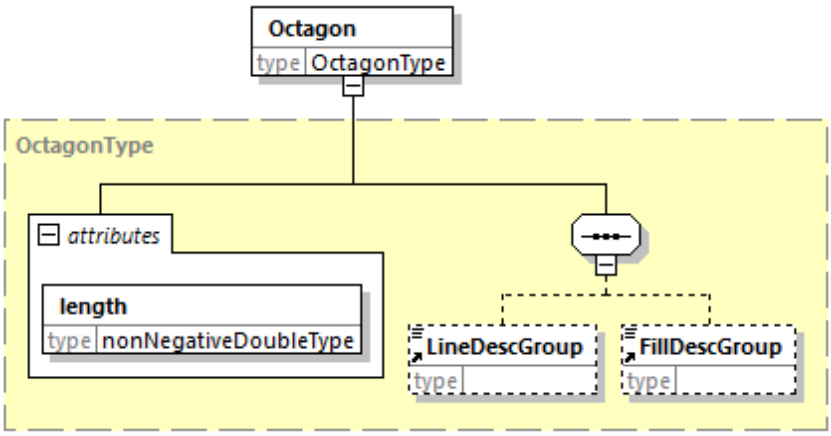
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Moire	MoireType	A series of circles, each consisting of a smaller diameter the shape of which is defined by the number of rings, their center line spacing, and the ring width. The pattern may also contain a crosshair representing its point of origin.	1-1
diameter	nonNegativeDoubleType	The diameter of the center of the outermost circle.	1-1
ringWidth	nonNegativeDoubleType	The width of the line used for each circle.	1-1
ringGap	nonNegativeDoubleType	The gap between circle lines as defined by the dimension between the centerlines of each circle location. The gap between centerlines must be larger than the `ringWidth` so that there is a clearance between individual rings.	1-1
ringNumber	nonNegativeInteger	The number of rings.	1-1
lineWidth	nonNegativeDoubleType	The line width used to develop a cross hair across the moiré. The default is 0.	0-1
lineLength	nonNegativeDoubleType	The length of the line for both the horizontal and vertical cross hair.	0-1
lineAngle	angleType	The angle at which the cross hair may be rotated. Rotation is always counter-clockwise. The default is 0° and can be oriented up to 90°.	0-1



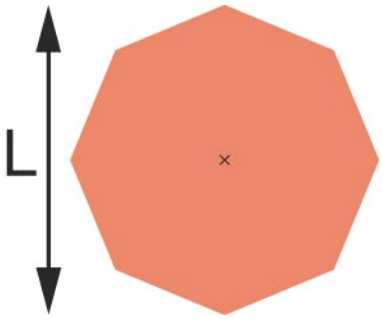

```
<Moire diameter = "8.4" ringWidth = "0.3" ringGap = "0.6" ringNumber = "5"
        lineWidth = "0.3" lineLength = "8.2" lineAngle = "0"/>
</Moire>
```

3.5.9.9 Octagon

An `Octagon` is an eight-sided `StandardPrimitive` shape with each of the sides being equal in length and with all angles between adjacent sides also being equal. The orientation of the `Octagon` is in accordance with one of its points facing the North direction. Only one dimension is required and that is the length across the points. `Rotation` is accomplished using `Xform` at the time the `Octagon` is instanced.



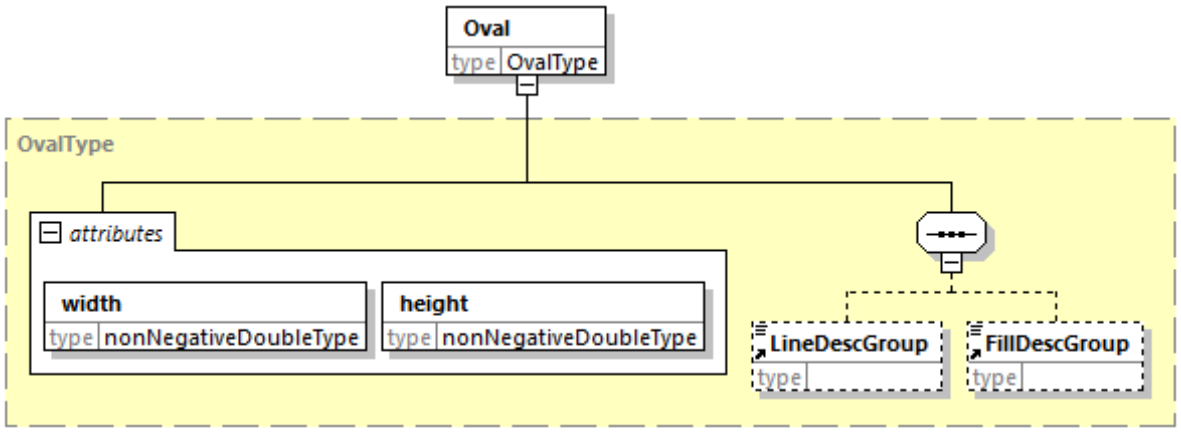
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Octagon	OctagonType	An eight-sided primitive shape with all sides being equal and which is defined by the length (L) across the points. The position of the octagon is in accordance with one of its points facing north.	1-1
length	nonNegativeDoubleType	The length (L) between any two opposing corner points of the octagon.	1-1
LineDescGroup	LineDescGroupType	A substitution group that specifies either `LineDesc` or `LineDescRef`. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either `FillDesc` or `FillDescRef`. Refer to section 3.5.6 FillDesGroup	0-1



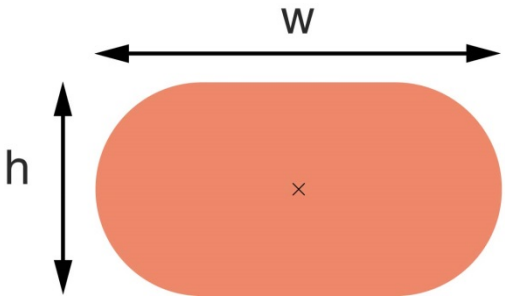

```
<Octagon length = "12.8">
        <FillDesc fillProperty="FILL">
                <Color r="150" g="0" b="0"/>
        </FillDesc>
 </Octagon>
```

3.5.9.10 Oval

An `Oval` is a `StandardPrimitive` shape that defines a rectangle with a complete radius (180 degree arc) at each end. The base rectangle is defined by a width attribute and a height attribute with the center of the rectangle being centered on both the height and width dimensions. The center of the `Oval` rectangle is the point of origin of the shape. The rectangle is defined with edges parallel to the x-axis and y-axis relative to the local coordinate system. Rotation is about the point of origin. The `Oval` is defined with the radius located along the y-axis sides. The radius on the ends of the oval shaped rectangle is always equal to ½ the height.



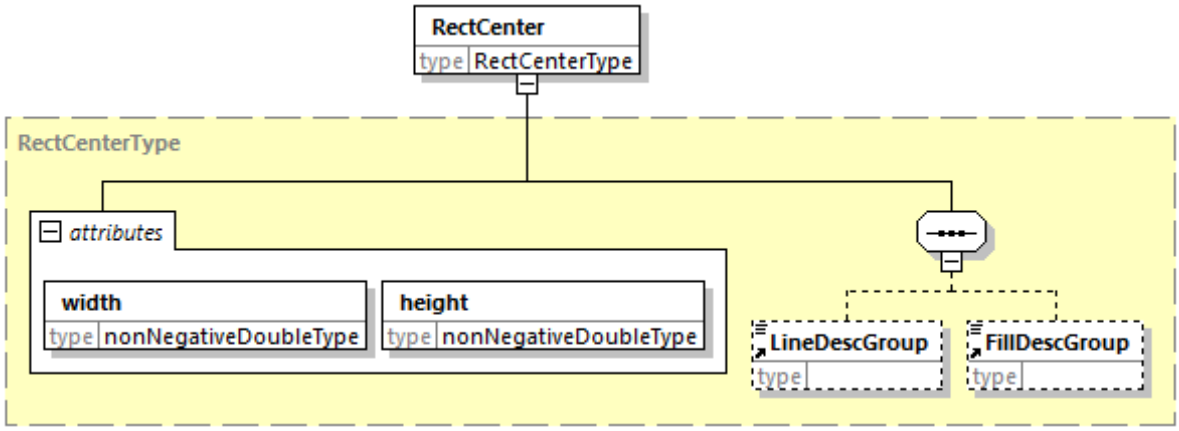
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Oval	OvalType	A rectangle defined by a width and height attribute with the center of the rectangle being centered on both the height and width dimensions.	1-1
width	nonNegativeDoubleType	The length of the rectangle about the x-axis.	1-1
height	nonNegativeDoubleType	The length of the rectangle about the y-axis. It is an error to define a height greater than the width.	1-1
LineDescGroup	LineDescGroupType	A substitution group that specifies either `LineDesc` or `LineDescRef`. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either `FillDesc` or `FillDescRef`. Refer to section 3.5.6 FillDesGroup	0-1



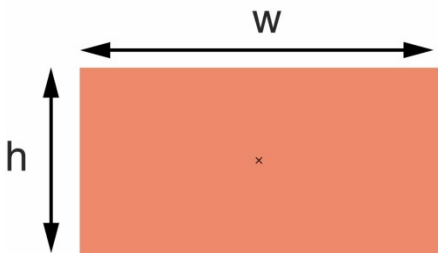

```
<Oval width = "10.6" height = "6.4">
      <FillDesc fillProperty="FILL">
            <Color r="150" g="0" b="0"/>
      </FillDesc>
</Oval>
```

3.5.9.11 RectCenter

A `RectCenter` is a primitive shape that defines a rectangle by a `width` attribute and a `height` attribute with the center of the rectangle being centered on both the height and width dimensions. The center of the `RectCenter` rectangle is the point of origin of the shape. The rectangle is defined with edges parallel to the x-axis and y-axis relative to the local coordinate system. Rotation is about the point of origin. The `RectCenter` is also used to represent a square shape. The characteristics of the square would be to have the width and height equal.



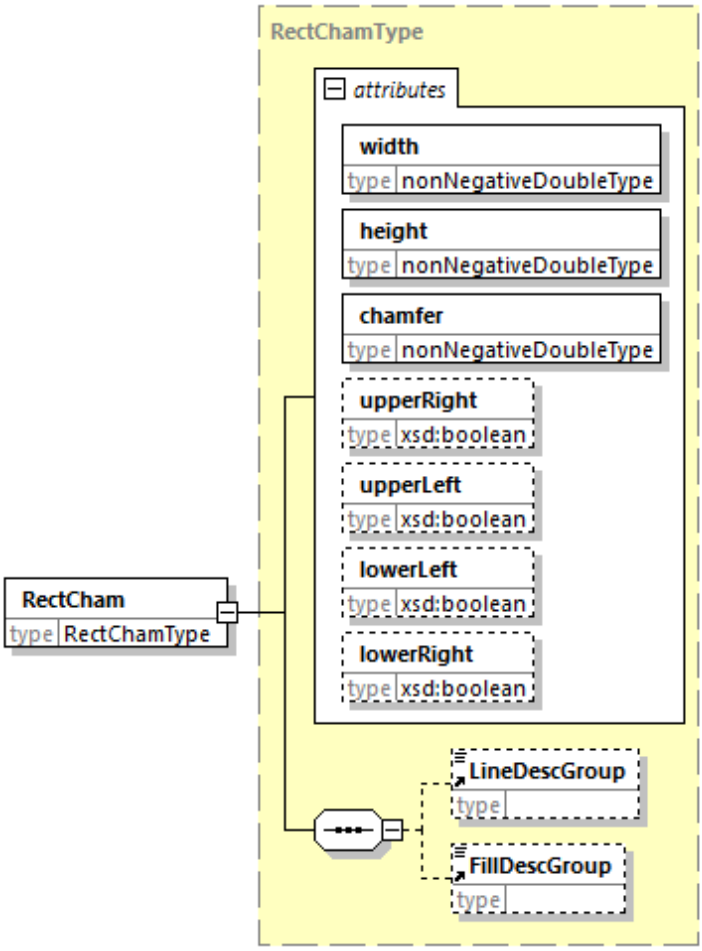
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
RectCenter	RectCenterType	The characteristics of a rectangle defined by a width and height dimension consistent with a horizontal position on the Cartesian coordinate system. The default for FillDesc is FILL without a LineDesc element.	1-1
width	nonNegativeDoubleType	The length of the rectangle about the x-axis.	1-1
height	nonNegativeDoubleType	The length of the rectangle about the y-axis.	1-1
LineDescGroup	LineDescGroupType	A substitution group that specifies either `LineDesc` or `LineDescRef`. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either `FillDesc` or `FillDescRef`. Refer to section 3.5.6 FillDesGroup	0-1




```
<RectCenter width = "16.8" height = "6.4">
        <FillDesc fillProperty="FILL">
                <Color r="150" g="0" b="0"/>
        </FillDesc>
</RectCenter>
```

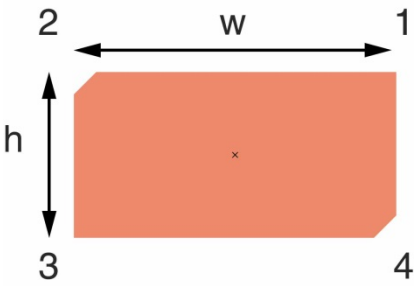
3.5.9.12 RectCham

A `RectCham` is a `StandardPrimitive` shape that defines a rectangle with chamfered corners. The base rectangle is defined by a `width` attribute and a `height` attribute with the center of the rectangle being centered on both the height and width dimensions. The center of the `RectCham` rectangle is the point of origin of the shape. The rectangle is defined with edges parallel to the x-axis and y-axis relative to the local coordinate system. Rotation is about the point of origin. The `RectCham` is also used to represent a square shape with chamfered corners. The characteristics of the square would be to have the width and height equal.

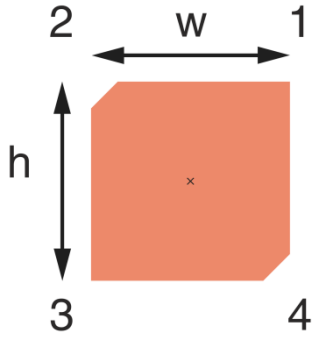


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
RectCham	RectChamType	A rectangle with one or more corners chamfered. The user has the option to define any of the corners as containing the chamfer as well as the chamfered dimensions. A minimum of one (1) chamfered corner must be defined. All chamfers (or opportunities for chamfers) must be identical in size.	1-1
width	nonNegativeDouble Type	The length of the rectangle about the x-axis.	1-1
height	nonNegativeDouble Type	The length of the rectangle about the y-axis.	1-1
chamfer	nonNegativeDouble Type	The length measured from each corner that defines 4 points along the width and 4 points along the height. The corners are clipped between the points at each corner. The resulting chamfers are always cut at 45° relative to the local coordinate system. It is an error to define the value of chamfer to be greater than ½ the height or ½ the width.	1-1

upperRight	Boolean	The upper right corner (1).	0-1
upperLeft	Boolean	The upper left corner (2).	0-1
lowerLeft	Boolean	The lower left corner (3).	0-1
lowerRight	Boolean	The lower right corner (4).	0-1
LineDescGroup	LineDescGroupType	A substitution group that specifies either `LineDesc or LineDescRef`. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either `FillDesc or FillDescRef`. Refer to section 3.5.6 FillDesGroup	0-1



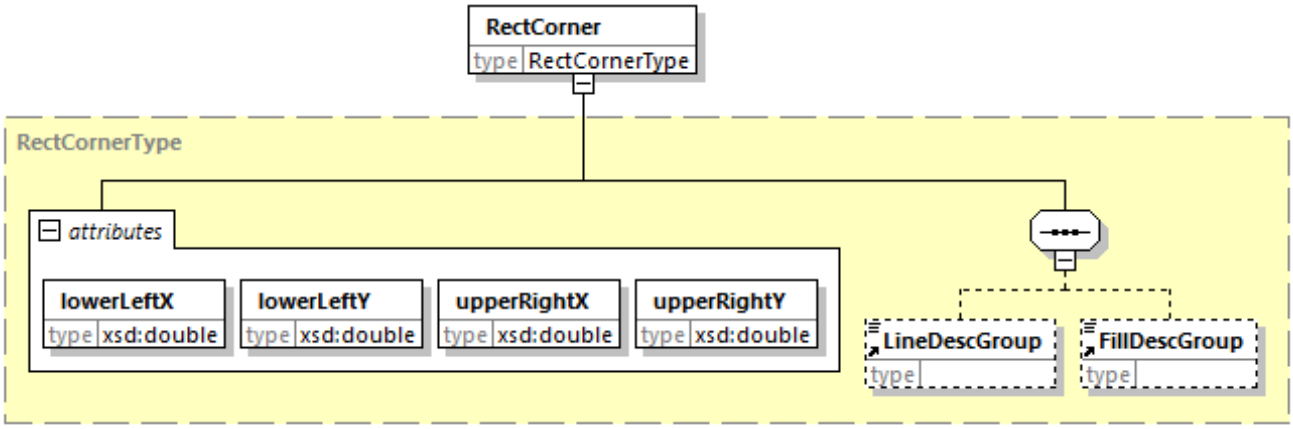

```
<RectCham width = "10.6" height = "6.2" chamfer = "2.0" upperRight= “false”
      upperLeft = “true” lowerLeft = “false” lowerRight = “true”>
        <FillDesc fillProperty="FILL">
              <Color r="150" g="0" b="0"/>
        </FillDesc>
</RectCham>
```




```
<RectCham width = "8.4" height = "8.4" chamfer = "2.0" upperRight = “false”
     upperLeft = “true” lowerLeft = “false” lowerRight = “true” >
        <FillDesc fillProperty="FILL">
              <Color r="150" g="0" b="0"/>
        </FillDesc>
</RectCham>
```

3.5.9.13 RectCorner

A `RectCorner` is a `StandardPrimitive` shape that defines a rectangle. The element describes the lower left and upper right corners of the rectangle. The point of origin of a `RectCorner` rectangle is (0, 0). This can be coincident with attribute `lowerLeftX` and `lowerLeftY`, the lower left corner of the rectangle, but there is no requirement for that location to be at (0, 0). The rectangle is defined with edges parallel to the x-axis and y-axis relative to the local coordinate system. Rotation is about the point of origin, not about the lower left or upper right corners.

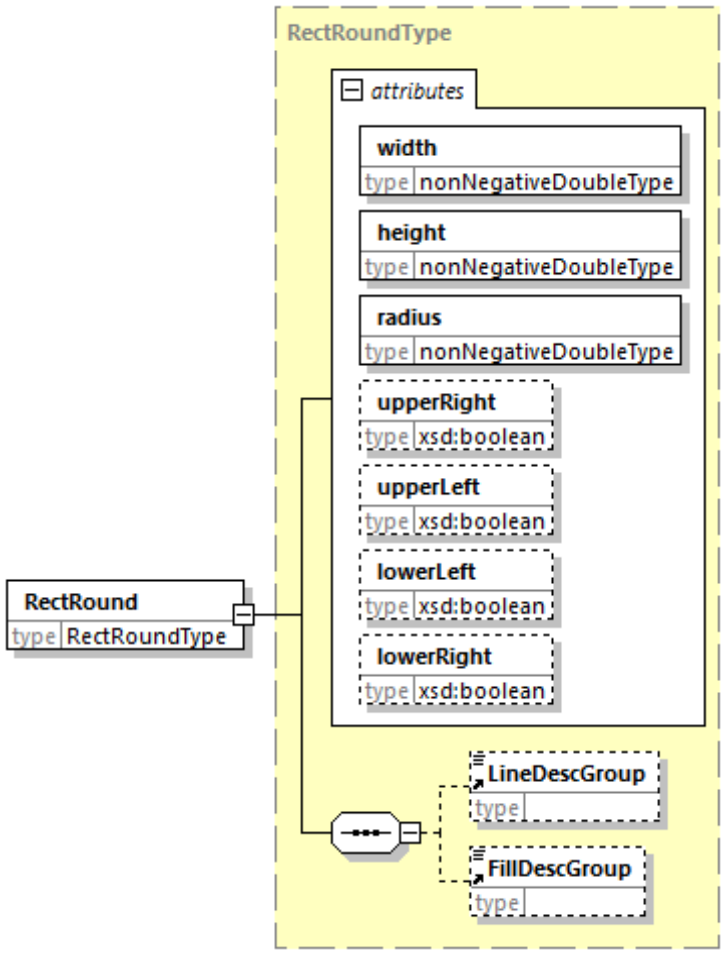


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
RectCorner	RectCornerType	A constraining rectangular area (bounding box) that describes a rectangle consistent with a horizontal position on the Cartesian coordinate system.	1-1
lowerLeftX	double	The lower left hand x dimension of the rectangular area shape.	1-1
lowerLeftY	double	The lower left hand y dimension of the rectangular area shape.	1-1
upperRightX	double	The upper right hand x dimension of the rectangular area shape.	1-1
upperRightY	double	The upper right hand y dimension of the rectangular area shape.	1-1
LineDescGroup	LineDescGroupType	A substitution group that specifies either `LineDesc or LineDescRef`. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either `FillDesc or FillDescRef`. Refer to section 3.5.6 FillDesGroup	0-1


```
<RectCorner lowerLeftX = "0.0" lowerLeftY = "0.0"
   upperRightX = "12.6" upperRightY = "6.8">
      <FillDesc fillProperty="FILL">
            <Color r="150" g="0" b="0"/>
      </FillDesc>
</RectCorner>
```

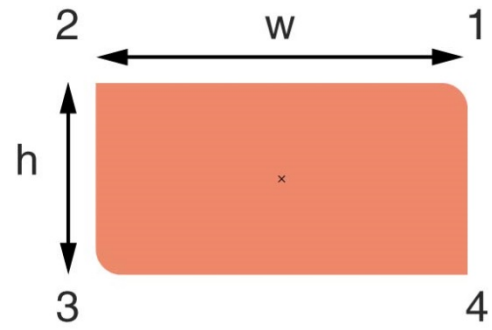
3.5.9.14 RectRound

A `RectRound` is a `StandardPrimitive` shape that defines a rectangle with radius corners. The base rectangle is defined by a width attribute and a height attribute with the center of the rectangle being centered on both the height and width dimensions. The center of the `RectRound` rectangle is the point of origin of the shape. The rectangle is defined with edges parallel to the x-axis and y-axis relative to the local coordinate system. Rotation is about the point of origin. The `RectRound` is also used to represent a square shape with rounded corners. The characteristics of the square would be to have the width and height equal.

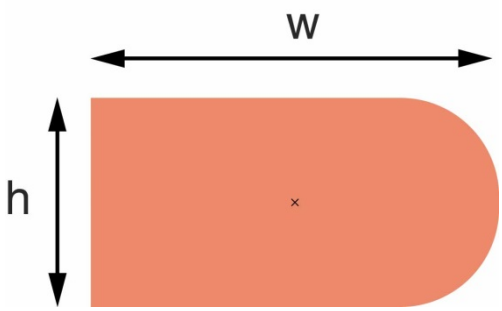


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
RectRound	RectRoundType	A rectangle with one or more corners rounded. The user has the option to define any of the corners as containing the radius as well as the radiused dimensions. A minimum of one (1) rounded corner must be defined.. All corners (or opportunities for corners) must be identical in size. The default for FillDesc is FILL without a LineDesc element.	1-1
width	nonNegativeDoubleType	The length of the rectangle about the x-axis.	1-1
height	nonNegativeDoubleType	The length of the rectangle about the y-axis.	1-1
radius	nonNegativeDoubleType	The radius to be trimmed from the four corners of the rectangle. It is an error to define a radius that is greater than ½ the height value or ½ the width value.	1-1
upperRight	boolean	The upper right corner (1).	0-1
upperLeft	boolean	The upper left corner (2).	0-1
lowerLeft	boolean	The lower left corner (3).	0-1

lowerRight	boolean	The lower right corner (4).	0-1
LineDescGroup	LineDescGroupType	A substitution group that specifies either `LineDesc` or `LineDescRef`. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either `FillDesc` or `FillDescRef`. Refer to section 3.5.6 FillDesGroup	0-1




```
<RectRound width = "10.2" height = "6.4" radius = "3.2"
   upperRight = “true” upperLeft = “false” lowerLeft = “true”
   lowerRight = “false”>
        <FillDesc fillProperty="FILL">
              <Color r="150" g="0" b="0"/>
       </FillDesc>
</RectRound>
```

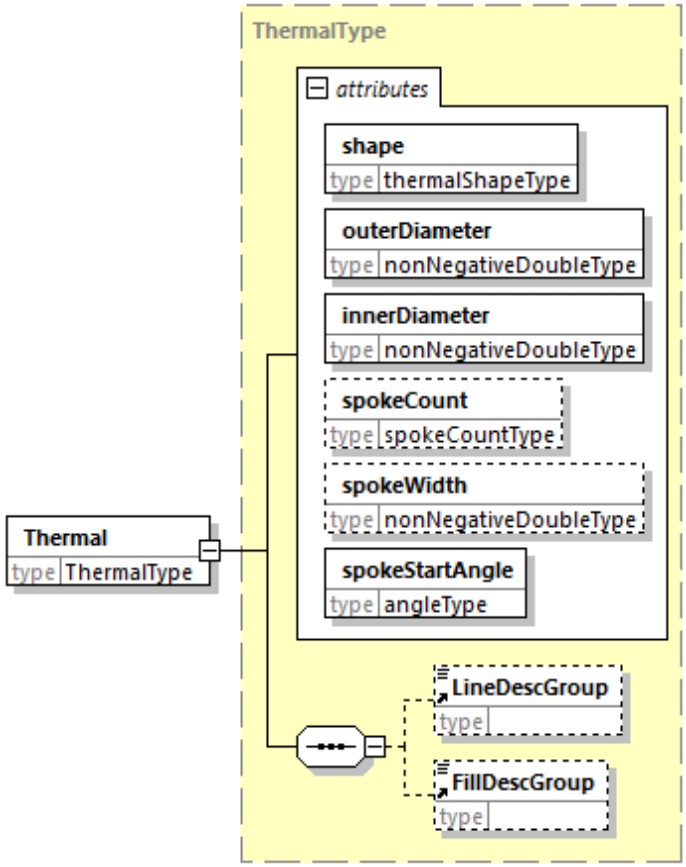



```
<RectRound width = "4.8" height = "4.8" radius = "2.4"
    upperRight = “true” upperLeft  = “false” lowerLeft = “false”
    lowerRight = “true”>
         <FillDesc fillProperty="FILL">
               <Color r="150" g="0" b="0"/>
       </FillDesc>
</RectRound>
```

3.5.9.15 Thermal

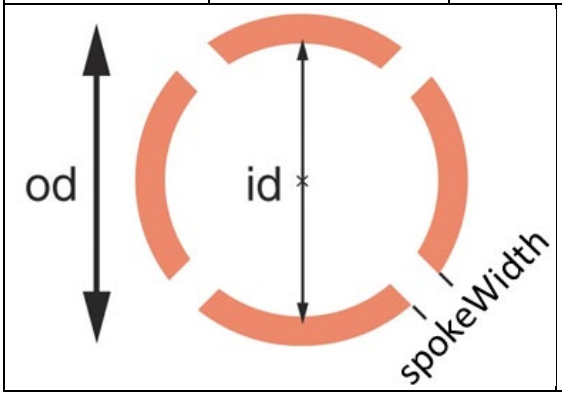
A `Thermal` is a `StandardPrimitive` shape that historically was used to remove material from a plane, conductive filled area or around a plated through hole. The `Thermal` shapes include square, round, hexagonal, or octagonal, and have varying numbers of spokes. The center of a thermal is the point of origin of the primitive.

A spokeless thermal can be used for nonfunctional lands on an innerlayer plane, where the land is not connected to the plane. IPC-2581 defines these using the `Thermal` element with a spoke count of zero. These are similar to a Donut except that they remove material. Many thermal primitive configurations can be generated using different spoke numbers and end types.

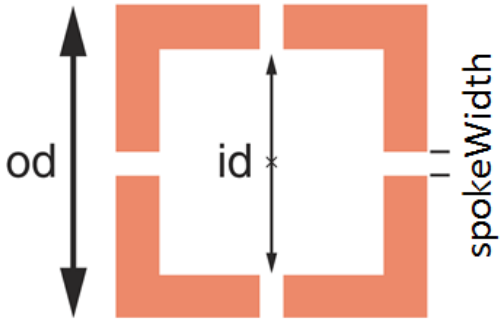
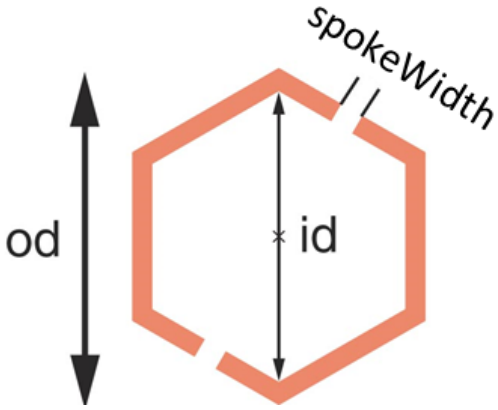
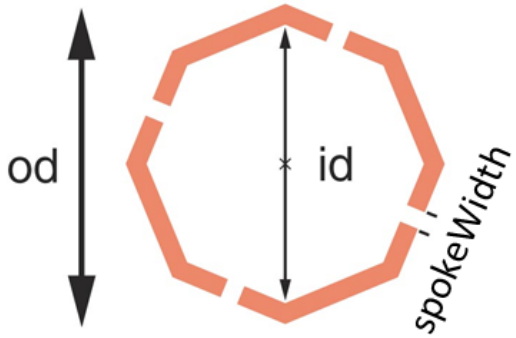
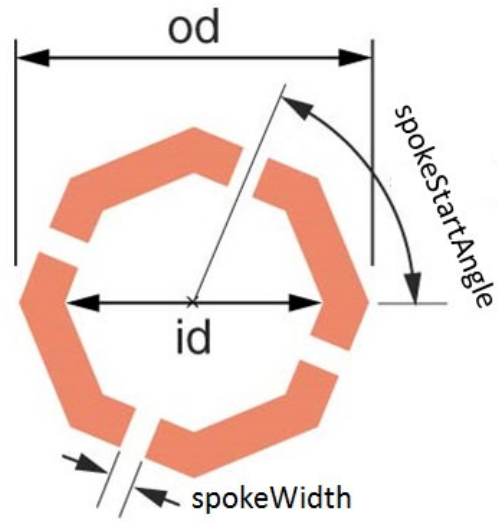


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Thermal	ThermalType	A primitive shape consisting either of ROUND, SQUARE, HEXAGON, or OCTAGON configuration that defines the removal of material from a plane or conductive fill area in accordance to the shape attribute description.	1-1
shape	thermalShapeType	The shape of the thermal. One of ROUND | SQUARE | OCTAGON. ROUND: The inner and outer shapes are like Circle. SQUARE: The inner and outer shapes are like RectCenter with height and width of each shape being equal. HEXAGON: The inner and outer shape are like Hexagon. OCTAGON: The inner and outer shapes are like Octagon.	1-1

outerDiameter	nonNegativeDouble Type	The outer boundary of the filled region. The meaning based on the shape attribute: ROUND: The diameter of the circle is the outer boundary of the thermal. The center of the circle is at the origin of the thermal. SQUARE: The width along the x-axis and the height along the y-axis of a square at the inner boundary of the thermal. The center of the square is at the origin. HEXAGON: The point-to-point measurement (L) on the x-axis of the hexagon that forms the outer boundary of the thermal. OCTAGON: The point-to-point (L) measurement on the x-axis of the octagon that forms the outer boundary of the thermal.	1-1
innerDiameter	nonNegativeDouble Type	The inner boundary of the filled region. The meaning based on the shape attribute: ROUND: The diameter of the circle is the inner boundary of the thermal. The center of the circle is at the origin of the thermal. SQUARE: The width along the x-axis and the height along the y-axis of a square at the inner boundary of the thermal The center of the square is at the origin. HEXAGON: The point-to-point measurement on the x-axis of the hexagon that forms the inner boundary of the thermal. OCTAGON: The point-to-point measurement on the x-axis of the octagon that forms the inner boundary of the thermal.	1-1
spokeCount	spokeCountType	The number of cutouts allowed in the inner and outer shapes. ROUND: must be 0, 2, 3, or 4 (the default is 4) SQUARE: must be 0, 2, or 4 (the default is 4) HEXAGON: must be 0, 2, or 3 (the default is 3) OCTAGON: must be 0, 2, or 4 (the default is 4) If the `spokeCount` is defined as 0 (zero), the other three optional parameters do not apply. The spokeless thermal has a shape like a donut shape.	0-1
spokeWidth	nonNegativeDouble Type	The minimum distance between the sides of a spoke cut. The default value is the `innerDiameter` subtracted from the `outerDiameter`.	0-1
spokeStartAngle	angleType	The angle in counterclockwise direction from the x-axis at which the first spoke is cut. The default angle is 45º counterclockwise from the x-axis.	1-1
LineDescGroup	LineDescGroupType	A substitution group that specifies either `LineDesc or LineDescRef`. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either `FillDesc or FillDescRef`. Refer to section 3.5.6 FillDesGroup	0-1

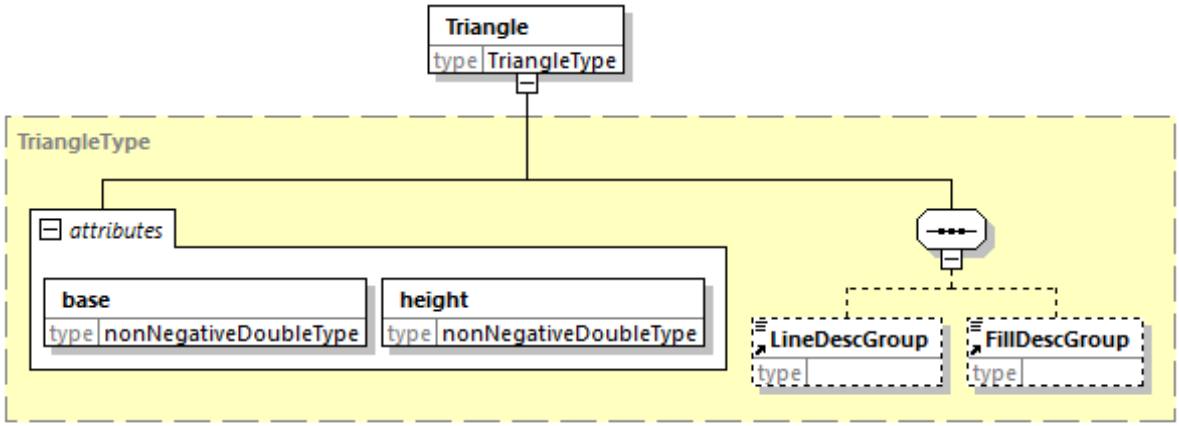



```
<Thermal shape = "ROUND" outerDiameter = "10.6" innerDiameter = "6.0"
  spokeCount = "4" spokeWidth = "2.0" spokeStartAngle = "45.00">
    <FillDesc fillProperty="FILL">
        <Color r="150" g="0" b="0"/>
    </FillDesc>
</Thermal>
```

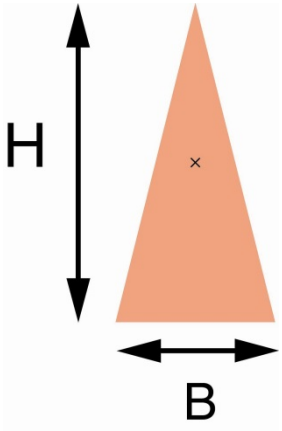
	<Thermal shape = "SQUARE" outerDiameter = "10.8" innerDiameter = "6.0" spokeCount = "4" spokeWidth = "2.4" spokeStartAngle = "0.00"> <FillDesc fillProperty="FILL"> <Color r="150" g="0" b="0"/> </FillDesc> </Thermal>
	<Thermal shape = "HEXAGON" outerDiameter = "10.8" innerDiameter = "6.4" spokeCount = "4" spokeWidth = "2.0" spokeStartAngle = "45.00"> <FillDesc fillProperty="FILL"> <Color r="150" g="0" b="0"/> </FillDesc> </Thermal>
	<Thermal shape = "OCTAGON" outerDiameter = "10.6" innerDiameter = "6.6" spokeCount = "4" spokeWidth = "2.0" spokeStartAngle = "60.00"> <FillDesc fillProperty="FILL"> <Color r="150" g="0" b="0"/> </FillDesc> </Thermal>
	<Thermal shape = "OCTAGON" outerDiameter = "10.6" innerDiameter = "6.0" spokeCount = "4" spokeWidth ="2.0" spokeStartAngle = "60.00"> <FillDesc fillProperty="FILL"> <Color r="150" g="0" b="0"/> </FillDesc> </Thermal>

3.5.9.16 Triangle

The `Triangle` is a `StandardPrimitive` shape that is an isosceles triangle that has two equal sides and a base. The shape is defined by the base and height dimension. The triangle is positioned with its point of origin which is at the center of the base and height dimensions.



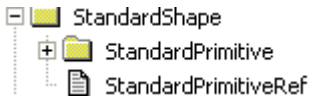
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Triangle	TriangleType	A primitive shape defined by a base and height dimension.	1-1
base	nonNegativeDoubleType	The distance between the two corner points of the base of the triangle with the point of origin at the center of the base and height dimensions.	1-1
height	nonNegativeDoubleType	The triangle height.	1-1
LineDescGroup	LineDescGroupType	A substitution group that specifies either `LineDesc` or `LineDescRef`. Refer to section 3.5.5 LineDesGroup	0-1
FillDescGroup	FillDescGroupType	A substitution group that specifies either `FillDesc` or `FillDescRef`. Refer to section 3.5.6 FillDesGroup	0-1




```
<Triangle base = "4.0" height = "8.0">
        <FillDesc fillProperty="FILL">
            <Color r="150" g="0" b="0"/>
        </FillDesc>
</Triangle>
```

3.5.10 StandardShape

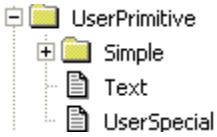
The StandardShape substitution group permits the substitution of any of the StandardPrimitive shapes in accordance with their individual descriptions. A predefined StandardPrimitive may also be instantiated by its unique "id" when the feature is contained in the DictionaryStandard. When a reference is made to the dictionary predefined primitive, the Units must match.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
StandardPrimitive	ABSTRACT	A substitution group that permits the substitution of the StandardPrimitive element with any of the sixteen standard primitive types.	0-n
StandardPrimitiveRef	StandardPrimitive RefType	A reference to a predefined StandardPrimitive, contained in the DictionaryStandard. The reference is by its unique "id". The units of the referenced predefined primitive and the Ecad section where it is to be instantiated must match.	0-n

3.5.11 UserPrimitive

The UserPrimitive substitution group consists of any simple graphic feature (Arc, Line, Outline or Polyline), as well as text or UserSpecial shapes. The UserSpecial element is a collection of Features (which are any of the permitted graphics used in the IPC-2581 file). UserSpecial permits the definition of logos, special targets, drawing formats or other graphics needed by a particular design. UserPrimitives can be predefined, assigned a unique "id" and contained in the DictionaryUser. The DictionaryUser defines the Units used to describe the graphic shapes.

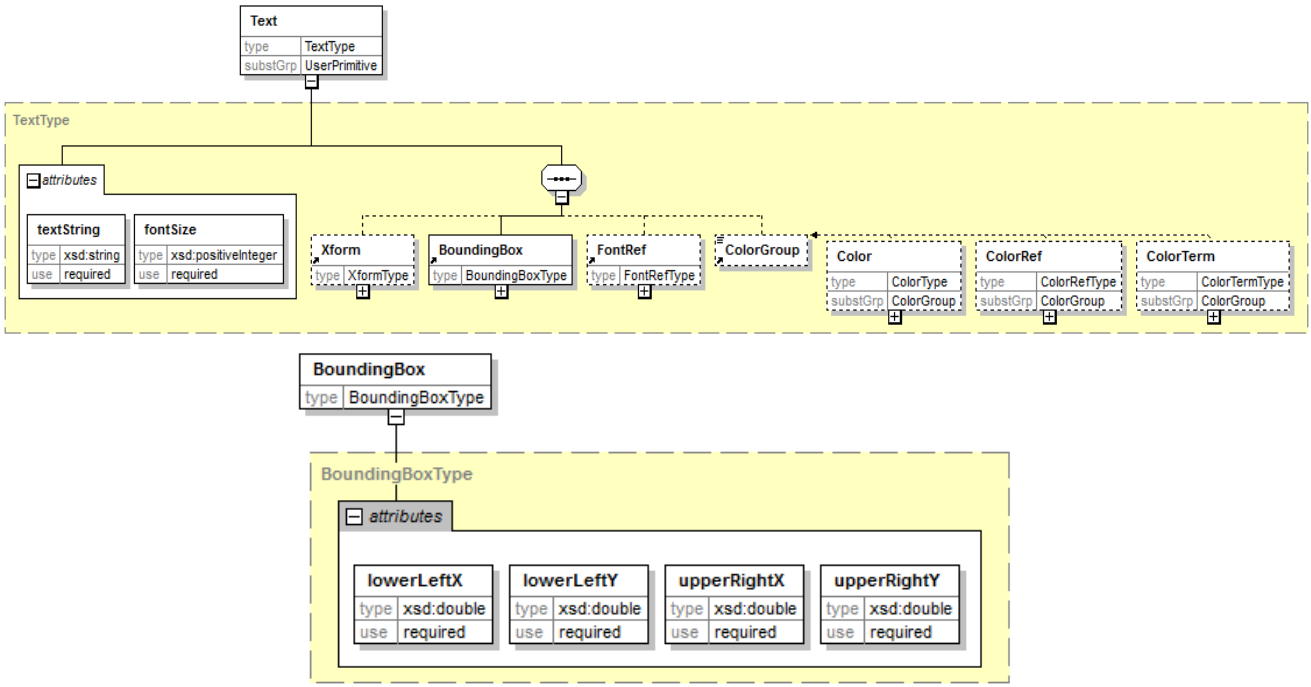


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Simple	ABSTRACT	A substitution consists of defining an Arc, Line, Outline or Polyline. The Simple substitution is called for in the schema in DfxMeasurement, Glyph, and Slot elements and may be defined in any other graphics.	0-n
Text	TextType	The text element defines text strings, fonts, and the bounding box containing the text. Also included are Xform to position, mirror or rotate the text.	0-n
UserSpecial	UserSpecialType	The UserSpecial element has all the capabilities allowed by the standard. The characteristic uses the substitution group "Feature" and may develop any combination of graphical shapes.	0-n

3.5.11.1 Text

When text is to be drawn on a product or a drawing the definition includes a bounding rectangle for the text. The lowerLeftX and lowerLeftY coordinate and the upperRightX and upperRightY coordinate define the BoundingBox rectangle. All portions of the text, including the line width of the strokes of the text, must fit within the BoundingBox rectangle. Any portion of a character exceeding the perimeter of the BoundingBox rectangle will be clipped at the boundaries of the BoundingBox rectangle.

Each Text entry (EntryUser) in the DictionaryUser **shall** have a unique id and consist of the following characteristics:



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Text	TextType	A pre-defined string of text that may be referenced and instantiated within the IPC-2581 file, including specific transformation.	0-n
textString	string	The text phrase (case-sensitive) in accordance with the language element of the `Header` element.	1-1
fontSize	positiveInteger	A dimensional characteristic in terms of an integer that defines the font size.	1-1
Xform	XformType	An element that provides the ability to reset the point of origin of the text box, then scale, mirror image or rotate the text box after the text box origin has been placed at an X and Y location. See 3.3.	0-1
Boundingbox	BoundingBoxType	A constraining rectangular area (bounding box) that encompasses the entire text string including upper and lower case characters.	1-1
lowerLeftX	double	The lower left hand x dimension of the rectangular area encompassing the text.	1-1
lowerLeftY	double	The lower left hand y dimension of the rectangular area encompassing the text.	1-1
UpperRightX	double	The upper right hand x dimension of the rectangular area encompassing the text.	1-1
UpperRightY	double	The upper right hand y dimension of the rectangular area encompassing the text.	1-1
FontRef	FontRefType	An element that is optional to reference a predefined font by its id, if the standard Helvetica font is not being instantiated.	0-1
id	qualifiedNameType	The identification of the `FontDef` stored in the `DictionaryFont`.	1-1
ColorGroup	ABSTRACT	An optional substitution group that permits assigning a particular color through instantiating the three basic colors, or by providing a reference to a predefined `Color` in `DictionaryColor`, or by choosing a basic color term from an enumerated list.	0-1

The following diagram and the requirements describe the general case for how text is to be drawn. There are two variations on the use of text. This makes `Text` an element that is incorporated as a layer feature or in a `Package` description. When used in this manner, all the characteristics of `Xform` and the `BoundingBox` apply. The other

form of text is as a simple `string` attribute. This is where the word `text` is used to add extra information to a particular element and therefore does not require the special features for `location`, `font`, and `Xform`.

When `text` is used as an element, the attribute `textString` should be defined to be enclosed in the textbox as illustrated in Figure 6. This includes upper and lower case letters, as well as all line widths, line descriptions, and line ends. Anything outside the clipping box will be clipped. The clipping boundary is necessary because fonts vary between computer systems and application implementations.

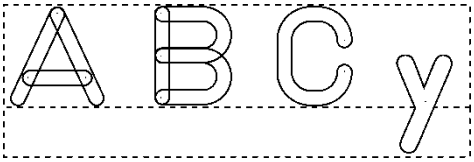


Figure 6 - Bounding rectangle to round end character relationships

3.5.11.1.1 Text restrictions

Text character dimensions are constrained by the bounding rectangle as illustrated in Figure 7. Character height is expressed by the `fontSize` attribute. Incremental units of the `BoundingBox` follow the `Units` element used by the file; this sets the limits (left and right xy coordinates) of the bounding rectangle. Both upper and lower case letters must be inside the `BoundingBox` rectangle. Included in this requirement are the extensions of such descending letters as lower case “g,” “q,” “y,” “j,” and “p.”

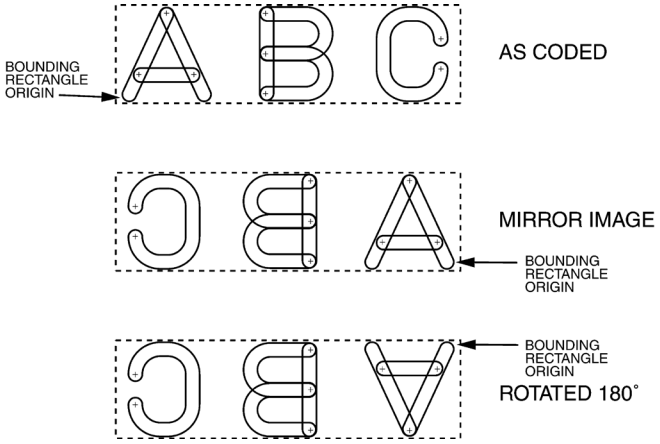


Figure 7 - Text transformation examples

3.5.11.1.2 Text rotation

The bounding rectangle of `Text` is defined relative to the local coordinate system. The `xLocation` and `yLocation` of `Xform` is applied to the bounding rectangle and the text contained within the rectangle to locate the `Text`. The bounding rectangle must be mirrored if required before it is rotated. The text is drawn relative to the bounding rectangle.

The example shown in Figure 8 indicates a `BoundingBox` rectangle that has been rotated 30° about the lower left xy coordinate.

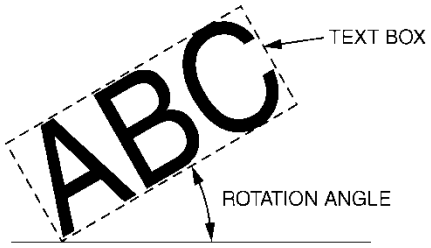
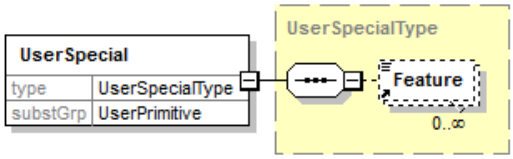


Figure 8 - Rotation Angle

3.5.11.2 UserSpecial

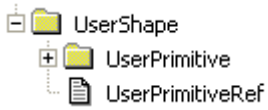
The `UserSpecial` may be any combination of `StandardShapes` or `UserShapes`, and is used to develop logos, targets, drawing formats or other combination of shapes.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
UserSpecial	UserSpecialType	A combination of primitive shapes that may be organized in any orientation needed to represent the user's needs for defining a special graphic shape.	0-n
Feature	ABSTRACT	A substitution group consisting of any graphic allowed by either the `StandardShape` or `UserShape` substitution groups.	0-n

3.5.12 UserShape

The `UserShape` substitution group permits the substitution of any of the `UserPrimitive` shapes in accordance with their individual descriptions. A predefined `UserPrimitive` may also be instantiated by its unique "id" when the feature is contained in the `DictionaryUser`. When a reference is made to the dictionary predefined primitive, the `Units` must match.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
UserPrimitive	ABSTRACT	A substitution group that occurs in the IPC-2581 schema and permits the substitution of the `UserPrimitive` element with any of the user primitive description or types.	0-n
UserPrimitiveRef	UserPrimitiveRefType	A reference to a predefined `UserPrimitive` contained in the `DictionaryUser`. The reference is by its unique "id". The units of the referenced predefined primitive and the Ecad section where it is to be instantiated must match.	0-n

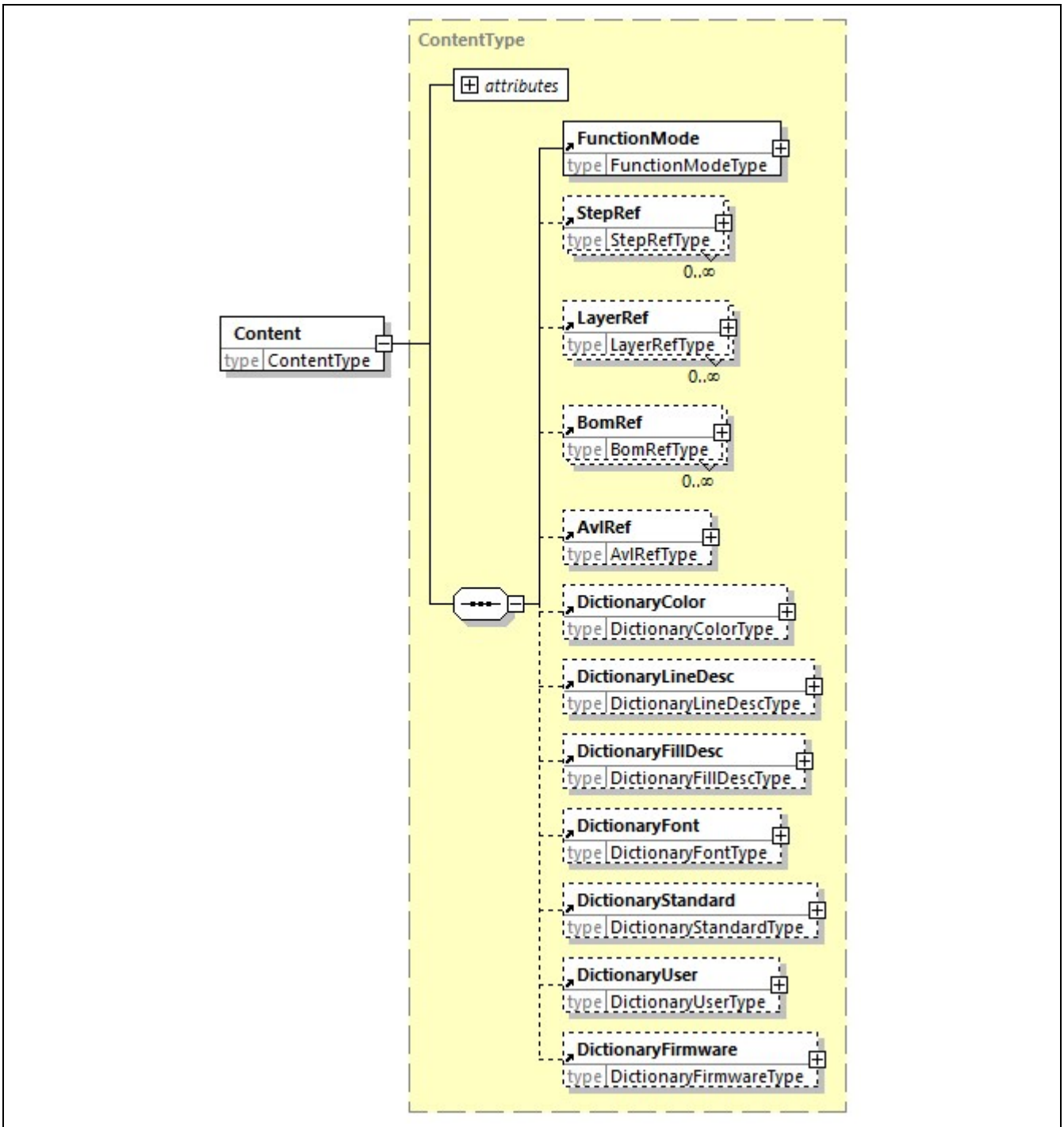
3.5.13 Feature

The `Feature` substitution consists of two major substitution groups. Where `Feature` is called for, an instance must substitute a graphic allowed by either the `StandardShape` or `UserShape` substitution groups.

Attribute / Element Name	Attribute / Element Type	Description	Occurrence
StandardShape	ABSTRACT	A substitution group that occurs in the IPC-2581 schema and permits the substitution of the `StandardShape` element when it is a child of the parent `Pad` element.	0-n
UserShape	ABSTRACT	A substitution group that occurs in the IPC-2581 schema and permits the substitution or classification of a higher level substitution group. The `UserShape` element may be used to further classify `Feature`. In so doing `UserShape` can be substituted by a `UserPrimitive` or `UserPrimitiveRef`.	0-n

4 CONTENT

The `Content` sub-element provides the information about the contents of the IPC-258X file. The `Content` schema identifies the depth and breadth of information in the file. The `Content` sub-elements include references to the `FunctionMode`, `StepRef`, `LayerRef`, `BomRef`, and `AvlRef` included in the file, plus seven Dictionaries: `DictionaryStandard`, `DictionaryUser`, `DictionaryFont`, `DictionaryLineDesc`, `DictionaryFillDesc`, `DictionaryColor` and `DictionaryFirmware`.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence

Content	ContentType	The `Content` element defines the function of the file, and references the major sections of the product description (i.e., Step, Layer, Bom and Avl). In addition, there are six dictionaries indicated in Content that would contain the pre-described information needed for the file details.	1-1
roleRef	qualifiedNameType	A reference to a globally unique name that identifies the role responsibility associated with the specific role at the time the file is transferred or archived.	1-1
FunctionMode	FunctionModeType	The function that the file is intended to perform between trading partners.	1-1
StepRef	StepRefType	The names of all Step elements that are included in the IPC-258X file through the reference of their unique name.	0-n
LayerRef	LayerRefType	The names of all Layer elements that are included in the IPC-258X file through the reference of their unique name.	0-n
BomRef	BomRefType	The names of all Bom elements that are included in the IPC-258X file through the reference of their unique name.	0-n
AvlRef	AvlRefType	A reference to the he `Avl` name established in the file. The Avl element contains the approved vendor list for all items contained in all Bom elements.	0-1
DictionaryColor	DictionaryColorType	An element that contains substitution group information using color description criteria, predefined by the user for reuse in the file.	0-1
DictionaryLineDesc	DictionaryLineDescType	An element that contains substitution group information using line description criteria, predefined by the user for reuse in the file.	0-1
DictionaryFillDesc	DictionaryFillDescType	An element that contains substitution group information using fill description criteria, predefined by the user for reuse in the file.	0-1
DictionaryFont	DictionaryFontType	An element that contains substitution group information regarding font descriptions as predefined Glyphs or references to external URN's for character sets that differ from the Helvetica standard.	0-1
DictionaryStandard	DictionaryStandard Type	An element that contains substitution group information using predefined descriptions of standard primitives identified by the IPC-2581 standard and described by the user for reuse in the file.	0-1
DictionaryUser	DictionaryUserType	An element that contains substitution group information using predefined descriptions of user primitives identified by the IPC-2581 standard and described by the user for reuse in the file.	0-1
DictionaryFirmware	DictionaryFirmware Type	An element that contains substitution group information using firmware description criteria, predefined by the user for reuse in the file.	0-1
The XML schema will have a restriction that requires the reference for `StepRef`, `LayerRef`, `BomRef`, and `AvlRef` be by the globally unique "name." The intent of the Content Element is to act as a table of contents for the IPC-2581 file.			

4.1 FunctionMode

The purpose of the Function Mode is to define which sections in the IPC-2581 schema are included in the IPC-2581 output file. Table 4 shows that the sections (table rows) to be included are determined by choosing one of seven modes (table columns) - UserDef, BOM, Stackup, Fabrication, Assembly,Test, and Stencil. The schema section options are detailed in Section 4.1.1. Descriptions of each mode are provided in Section 4.1.2. Referring to the table, the following are the four possible schema section values:

Y (in a green background) –included section. The section is either mandatory for the requirments of that mode, or mandatory to meet the IPC-2581 schema requirements

N (in a gray background) – excluded section. The section is considered never required for that mode

O (in a white background) – optional section. The section may provide additional data that may or may not be essential.

Table 4 Function Mode Table

Key	Schema Sections	Mode							
		UserDef	BOM	Stackup	Fabrication	Assembly	Test	Stencil	DFX
K	Padstack Definitions	O	N	N	O	O	N	N	N
B	**B**OM & AVL	O	Y	O	O	Y	Y	N	N
C	**C**omponent Packages	O	N	N	N	Y	Y	O	N
A	Component **A**ssembly	O	N	N	N	Y	Y	N	N
S	**S**tackup	O	N	Y	Y	N	N	N	N
U	Profile (O**u**tline)	O	N	O	Y	Y	Y	Y	N
M	Solder **M**ask Layers	O	N	N	Y	N	N	O	N
P	Solder **P**aste Layers	O	N	N	N	O	N	Y	N
L	Silkscreen (**L**egend) Layers	O	N	N	Y	Y	Y	O	N
R	Drilling and **R**outing Layers	O	N	O	Y	Y	Y	O	N
D	**D**ocumentation Layers	O	N	O	O	O	O	O	N
O	**O**uter Copper Layers	O	N	Y	Y	Y	Y	O	N
I	**I**nner Copper Layers	O	N	Y	Y	N	N	N	N
E	Di**e**lectric Layers	O	N	O	O	N	N	N	N
F	Miscellaneous **F**ab Layers	O	N	O	O	N	N	N	N
G	Lo**g**ical Netlist	O	N	N	O	O	O	N	N
Y	Ph**y**sical Netlist	O	N	N	Y	O	Y	N	N
X	DF**X** Measurement	O	O	O	O	O	O	O	Y

Every IPC-2581 export utility must adhere to the FunctionMode table when providing choices to the user, given what is available in the design data. However it is ultimately the user's responibility as to what is avaialable in the data. For example if there is no Miscellaneous Fab data then it can't be included in Fabrication Mode even though the table says it should be.

4.1.1 Schema Sections

Each row in Table 4 is briefly decribed below, including its list of related schema elements. These elements, and relevant child elements, must be included in the IPC-2581 output when the schema section is included in the chosen mode. When the elements refer to a layer, or layers, then only those layers whose layer attributes (shown in brackets) comply with the schema section should be included. For example (referring to Section 4.1.1.7), for the "Solder Mask Layers" section, a Layer element with `name` "SMT" would comply if its `layerFunction` attribute was "SOLDERMASK". So then the `Layer`, `LayerRef` and all `LayerFeature` elements for layer "SMT" would be included. When `LayerFeature` elements of any given layer are included, whether mandatory or optional, then the corresponding `Layer` & `LayerRef` elements **MUST** also be included for that layer.

4.1.1.1 Padstack Definitions

Instances of multilayer structure at a single point site defining the characteristics that exist at a particular point including land description or reference, non-pad description or reference, thermal connections or reference, and holes (through-hole, buried, blind, and microvias). Padstacks are for reference to the construction of the original design.

4.1.1.1.1 Padstack Definition Related Schema Elements

Ecad:CadData:Step:PadstackDef

4.1.1.2 BOM & AVL

The BOM (Bill of Material) is an inventory of items required for board fabrication, assembly, and test. Those items can be components, materials, tools or documents. They are described by attributes such as part number, quantity, and characteristics (value, tolerance, etc.). Note that the `Characteristics` element is a mandatory child of the `BomItem` element.

The AVL (Approved Vendor List) defines a mapping between the OEM part or item number (OEM Design Number) and one or more equivalent manufacturers part numbers (MPN). Each item in the BOM is uniquely identified by its `OEMDesignNumberRef` attribute. Each `OEMDesignNumberRef` value **must** reference an `OEMDesignNumber` value defined in the AVL.

4.1.1.2.1 BOM & AVL Related Schema Elements

Content:BomRef
Content:AvlRef
Bom
Avl

4.1.1.3 Component Packages

Package, silkscreen, assembly outlines, and land patterns of electronic and mechanical components, plus pin numbering and location, and additional attributes, including pin one location and orientation, pickup point, height, etc.

4.1.1.3.1 Component Packages Related Schema Elements

Ecad:CadData:Step:Package

4.1.1.4 Component Assembly Data

Describes the location (both xy and layer), and any rotation or mirroring, of the component instances on the electronic assembly. Each instance references a specific land pattern, and component package style.

4.1.1.4.1 Component Assembly Data Related Schema Elements

Ecad:CadData:Step:Component
Ecad:CadData:Layer (where layerFunction = COMPONENT | COMPONENT_TOP | COMPONENT_BOTTOM | COMPONENT_EMBEDDED | COMPONENT_FORMED | ASSEMBLY)
Ecad:CadData:Step:LayerFeature

4.1.1.5 Profile

Precise coordinates describing the shape of the board, including any cut outs, which can be defined with either primitive shapes or complex polygon elements.

4.1.1.5.1 Profile Related Schema Elements

Ecad:CadData:Step:Profile

4.1.1.6 Stackup

A description of the physical layers that represent the cross section of the board to be manufactured, including layer order, thickness per layer, overall thickness, and tolerance. Can also include layer grouping, such as for laminates, and specifications, to describe characteristics of materials, impedance control, etc.

4.1.1.6.1 Stackup Related Schema Elements

Content:LayerRef
Ecad:CadData:Layer
Ecad:CadData:Stackup

4.1.1.7 Soldermask Layers

Layer definitions and geometric features (artwork) for solder mask on external layers of a printed circuit board (top and bottom).

4.1.1.7.1 Soldermask Layer Related Schema Elements

Content:LayerRef
Ecad:CadData:Layer (where layerFunction = SOLDERMASK)
Ecad:CadData:Step:LayerFeature

4.1.1.8 Solderpaste Artwork

Layer definitions and geometric features (artwork) for solder paste on external layers of a printed circuit board (top and bottom)

4.1.1.8.1 Solderpaste Layer Related Schema Elements

Content:LayerRef
Ecad:CadData:Layer (where layerFunction = SOLDERPASTE)
Ecad:CadData:Step:LayerFeature

4.1.1.9 Silkscreen (Legend) Artwork

Layer definitions and geometric features (artwork) for legend marking to be incorporated on top of solder mask material.

4.1.1.9.1 Silkscreen Layer Related Schema Elements

Content:LayerRef
Ecad:CadData:Layer (where layerFunction = SILKSCREEN | LEGEND)
Ecad:CadData:Step:LayerFeature

4.1.1.10 Drilling & Routing Layers

Information on hole forming, either through mechanical drilling, laser ablation, or chemical mechanism and the relationship of the conductor routing layer.

4.1.1.10.1 Drilling & Routing Related Schema Elements

Content:LayerRef
Ecad:CadData:Layer (where layerFunction = DRILL | ROUT| V_CUT | EDGE_CHAMFER)
Ecad:CadData:Step:LayerFeature

4.1.1.11 Documentation Layers

Any drawing layers such as fabrication or component assembly drawings, board outline, board dimensions etc, that supplement the manufacturing process, but are not artwork that is used directly in the fabrication process

4.1.1.11.1 Documentation Layer Related Schema Elements

Content:LayerRef
Ecad:CadData:Layer (where layerFunction = DOCUMENT | GRAPHIC | BOARD_OUTLINE | REWORK | FIXTURE | PROBE | COURTYARD)
Ecad:CadData:Step:LayerFeature

4.1.1.12 Outer Copper Layers

Layer definitions and, with the exception of the STACKUP function mode, geometric features (artwork) that describe the etched copper on the outer (top & bottom) layers. Mandatory for fabricating the board. May also be useful for assembly and test to identify component and probe placements, and for stencil creation.

4.1.1.12.1 Outer Copper Layer Related Schema Elements

Content:LayerRef
Ecad:CadData:Layer (where layerFunction = CONDUCTOR | CONDFILM | CONDFOIL | PLANE | SIGNAL | MIXED ***and*** side = TOP | BOTTOM)
Ecad:CadData:Step:LayerFeature (not required for STACKUP function mode).

4.1.1.13 Inner Copper Layers

Layer definitions and, with the exception of the STACKUP function mode, geometric features (artwork) that describe the etched copper on all inner layers. Mandatory for fabricating the board.

4.1.1.13.1 Inner Copper Layer Related Schema Elements

Content:LayerRef

Ecad:CadData:Layer (where layerFunction = CONDUCTOR | CONDFILM | CONDFOIL | PLANE | SIGNAL | MIXED ***and*** side = INTERNAL)
Ecad:CadData:Step:LayerFeature (not required for STACKUP function mode).

4.1.1.14 Dielectric Layers

Layer definitions and any geometric features (artwork) that describe the dielectric layers. Unless these layers contain cut outs for embedded components they may not contain any geometric features, but they may reference specifications, describing their material type and properties, that are essential to the fabrication process.

4.1.1.14.1 Dielectric Layer Related Schema Elements

Content:LayerRef
Ecad:CadData:Layer (where layerFunction = DIELBASE | DIELCORE | DIELPREG | DILEADHV | DIELBONDPLY | DIELCOVERLAY))
Ecad:CadData:Step:LayerFeature

4.1.1.15 Miscellaneous Fab Layers

Miscellaneous layer definitions and geometric features (artwork) that are essential to complete the fabrication process, such as plating or coating.

4.1.1.15.1 Miscellaneous Fab Layer Related Schema Elements

Content:LayerRef
Ecad:CadData:Layer (where layerFunction = COATINGCOND | COATINGNONCOND | CONDUCTIVE_ADHESIVE | GLUE | HOLEFILL | SOLDERBUMP | THIEVING_KEEP_INOUT | EDGE_CHAMFER | EDGE_PLATING | STIFFENER | CAPACITIVE | RESISTIVE)
Ecad:CadData:Step:LayerFeature

4.1.1.16 Logical NetList

A list of all nets where each net is described by the component pins connected to it.

4.1.1.16.1 Logical Netlist Related Schema Elements

Ecad:CadData:Step:LogicalNet

4.1.1.17 Physical NetList

A list of all nets where each net is described by the location (co-ordinates and layer) of physical points along its copper etch, as well as information required for bare board electrical testing.

4.1.1.17.1 Physical Netlist Related Schema Elements

Ecad:CadData:Step:PhyNetGroup

4.1.1.18 DFX

A collection of various measurements that are related to a particular characteristic of the product.

4.1.1.18.1 DFX Related Schema Elements

Ecad:CadData:Step:Dfx

4.1.2 Modes

The following details the purpose of each mode available in the function mode table.

4.1.2.1 UserDef

All schema sections are optional. This mode is primarily intended for development or test purposes, to investigate options or requirements that were not thought of at the time of writing this document. It should **not** be used for formal data transfer. Both sender and receiver must first agree to any unofficial transfer. However the mode could also be used internally, say with all options turned on, for storage or pre-processing purposes.

4.1.2.2 BOM

Primarily intended for a preliminary Bill Of Materials, before a schematic or layout exists, e.g. for long lead time part ordering, but could be used at any time during the manufacturing process if only a BOM is required.

4.1.2.3 Stackup

Primarily intended for preliminary stackup analysis before the layout exists, such as for bi-directional transfer between OEM and fabricator (e.g. to agree on materials), or between CAD tool and Signal Integrity, Impedance Calculator, or other tool. However this mode could be used at any time that only a stackup is required, such as from the fabricator for final validation before fabrication occurs. Note that:

- Even for “Stackup only” (no board profile and/or routing exists yet), if the *layerOrGroupRef* attribute of a given *StackupLayer* element refers to a layer (not a group) then that layer **must** be pre-defined in both a *Content:LayerRef* element and a *CadData:Layer* element. That is, it is illegal to have only a Stackup element in an output – all external references from within that element must be included..

4.1.2.4 Fabrication

All the data required to fully fabricate a Printed Circuit Board. **See IPC-2584 and IPC-2588 for sectional data descriptions.**

4.1.2.5 Assembly

All the data required to fully assemble a Printed Circuit Board. **See IPC-2584 and IPC-2588 for sectional data descriptions. See IPC-2586 and IPC-2588 for sectional data descriptions.**

4.1.2.6 Test

All the data required to test a Printed Circuit Board. **See IPC-2584 and IPC-2588 for sectional data descriptions. See IPC-2585 and IPC-2587 for sectional data descriptions.**

4.1.2.7 Stencil

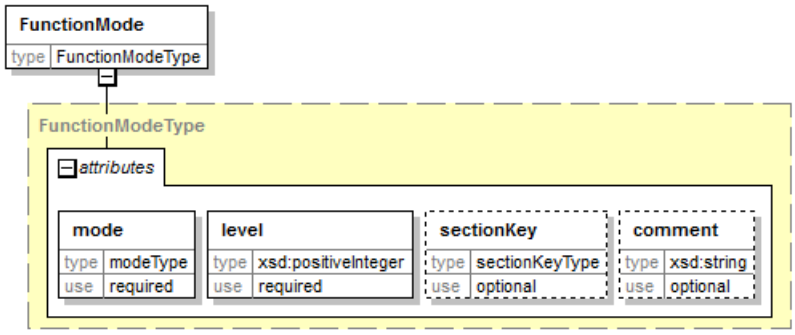
All possible data required to produce solder paste stencils for a Printed Circuit Board.

4.1.2.8 DFX

To allow for only Dfx (measurements vs pass/fail criteria) information to be exchanged without any other design data. Primarily intended for efficient bi-directionasl DFx information exchange, or transfer of DFx information to/from a system that cannot read/write the design data.

4.1.3 FunctionMode Element

The FunctionMode element in the IPC-2581 output file serves to indicate what function the file is intended to perform, by referencing the mode that was used from the Function Mode Table (see Table 4) to populate that file’s contents. The element has 4 attributes - `mode`, `level`, `sectionKey`, and `comment`.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
FunctionMode	FunctionModeType	The function that the file is intended to perform between trading partners.	1-1
mode	modeType	An enumerated string, either USERDEF | BOM | STACKUP | FABRICATION | ASSEMBLY | TEST | STENCIL, that defines the type of mode that the file is intended to serve.	1-1
level	positiveInteger	A numerical value, fixed at 1. For backward compatibility purposes only. This attribute will be removed in revision C	1-1
sectionKey	sectionKeyType	A list of alphabetic characters that is equal to, or a subset of, the enumerated list “ABCDEFGIKLMOPRSUXY”. Each character in the list represents a schema section that is included in the output file. Refer to Table 4 for the key to schema section mapping. So if a schema section is omitted from the output file, to comply with a given function mode, then its corresponding character is removed from the sectionKey list.	0-1
comment	string	Any appropriate comment to help clarify the intended use of the file.	0-1

4.1.3.1 Mode

The `FunctionMode` element has four attributes, the first of which is `mode`. There are seven valid values for the `mode` attribute. These are: USERDEF, BOM, STACKUP, FABRICATION, ASSEMBLY, TEST, and STENCIL. Refer to section 0 for a description of the purpose of each of these modes.

Note that, with the options available, it is possible to include the TEST or STENCIL mode as a subset of the ASSEMBLY mode. This will allow receiving companies who cater for both assembly and test, or assembly and stencil, or all three, to receive a single file. Nevertheless, other companies wish to outsource the test and/or stencil activities separately. Therefore it is important to have separate modes for these activities.

4.1.3.2 Level

In previous revisions of this document each mode in the FunctionMode table was further divided into 3 levels, and so a second attribute, `level` (a positive integer from 1 to 3) was defined for the FunctionMode element to support this. In this revision amendment the levels per mode have been reduced to one, to simplify the table, effectively making the `level` attribute redundant. However the level attribute is retained, for backward compatibility purposes only, and must be fixed at the value ‘1’. It will be removed in the next revision.

4.1.3.3 SectionKey

The `sectionKey` attribute is a string that can precisely indicate which schema sections are included in the output file. So if the indicated mode has options then the receiver will know which of those optionsa are included. It is formed by combining the key characters (as denoted in the first column of Table 4) of each of the included sections into a single string. For example the `sectionKey` value “KSUMLROIEF” represents the FABRICATION mode with additional dielectric layer and miscellaneous FAB layer options.

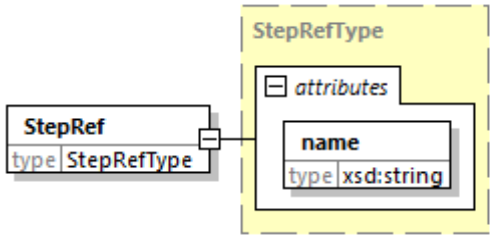
4.1.3.4 Comment

The attribute `comment` is optional string data, intended to provide any clarification as to the intended use of the file.

4.2 StepRef

The reference to the names of all `Step` elements used in the description of the Printed Circuit Assembly (PCA), the unpopulated board or other related information (e.g., documentation). In manufacturing, this basic `Step` is often step and repeated (nested) inside a larger step (called array, or sub-panel). This array step can be further nested into another step (called a production panel). The `Ecad` element always contains at least one `Step`, but may contain several, some basic ones and others nesting previous steps.

The `StepRef` element, as it appears in the `Content` schema, references the job step's names and thus the various steps that are included in the IPC-2581 file. All the graphical data of a IPC-2581 job are located inside steps that can be nested inside each other (PCB/Sub Panel/Panel, etc.). Steps are referenced in the `Content` schema (`StepRef`) as a `string` that relates to the details in the `Ecad` schema.

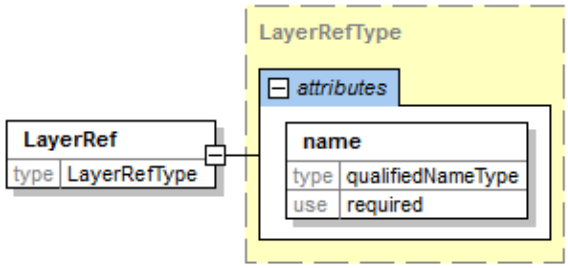


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
StepRef	StepRefType	The reference to all existing `step` elements contained within the file. This feature is a method of checking completeness in file transfer.	1-n
name	string	The unique name of all Step segment descriptions.	1-1

4.3 LayerRef

Layers, as the name implies, are sheets of two-dimensional data that, when laid on top of each other, create the Printed Circuit Assembly (unpopulated PCB and components or other related information). The `Layer` element appears in the IPC-2581 file as a sub-element of the `CadData` element.

The `LayerRef` element, as it appears in the `Content` schema, references all the file `Layer` unique names included in the IPC-2581 file.



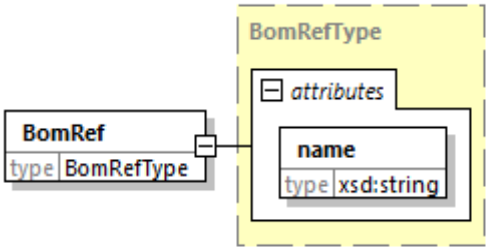
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
LayerRef	LayerRefType	The reference to all existing layer elements contained within the file. This feature is a method of checking completeness in file transfer.	1-n
name	qualifiedNameType	The unique name of all Layer segment descriptions	1-1

4.4 BomRef

The `BOM` section describes the Bill of Materials for the board. A bill of materials is a list of all the different components, materials, mechanical parts, or programmable software used in the electronic product. Components are arranged by the `OEMDesignNumber` or an alternate; materials for board fabrication or component attachment are arranged by their appropriate identifier. Each part number has a list of attributes and is accompanied by a list of the various specific uses or locations on the electronic product, each with its unique name.

The OEMDesignNumber is the part number from the original source OEM, not necessarily the component manufacturer part number.

Each `BomRef` element, as it appears in the `Content` schema, references one of the potentially many Bom categories and the number of items included in each category in the IPC-2581 file.

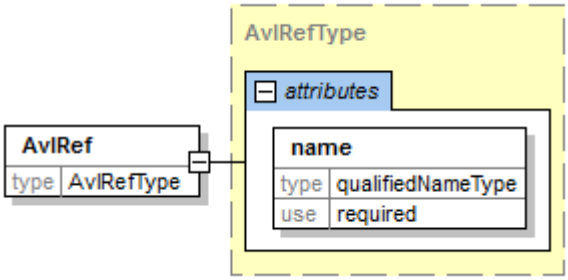


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
BomRef	BomRefType	The reference to all BOM elements that are arranged by the category description contained as an attribute of each BomItem, e.g. ELECTRICAL | PROGRAMMABLE | MECHANICAL | MATERIAL | DOCUMENT. This feature identifies the specific number of BomItems as well as the category to which they pertain, and is a method of checking completeness in file transfer.	0-n
name	string	The unique name of all Bom segment descriptions.	1-1

4.5 AvlRef

The `Avl` section describes the Approved Vendor Lists for the materials used to fabricate the board and the assembly. The BOM (bill of material) lists include all the different components to be used on the board, arranged by their appropriate part number, and material used to fabricate the board arranged by the part number of the material. There are also BOMs for the material used that are consumed by the fabrication and assembly processes. Each BOM has a corresponding list of approved vendors if the customer wishes to restrict the components and materials used for the electronic assembly to a specific supplier(s). There **shall** be only one `Avl` section in a IPC-2581 file. It **shall** provide the names of each of the approved suppliers and **shall** correlate them with the BOM that contains the material/component descriptions.

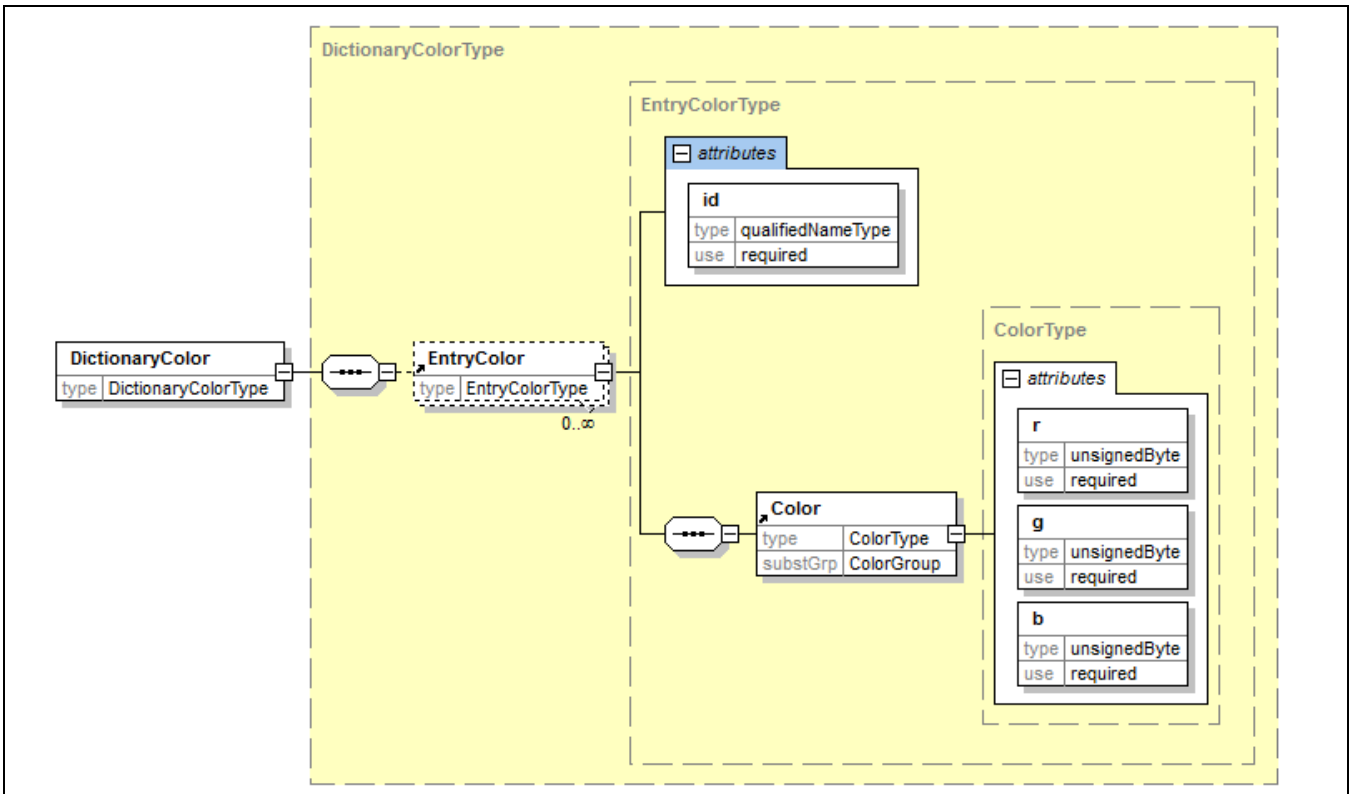
The Avl is used by the customer, the fabricator and the assembler to coordinate the relationship with the bills of materials described in the IPC-2581 file.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
AvlRef	AvlRefType	A reference to the single Approved Vendor List for all the suppliers identified and the total number in the IPC-2581 file. This feature is a method of checking completeness in file transfer.	0-1
name	qualifiedNameType	The unique name of the single Approved Vender List segment descriptions.	1-1

4.6 DictionaryColor

The `DictionaryColor` is intended to provide lookup information on predefined `Color descriptions`. The `DictionaryColor` is maintained as part of a substitution group schema. The intent is to have color descriptions available that are identified by their three color hues and intensity characteristics and a specific name (id). They may be reused throughout the file as appropriate. The name (id) of a `color` must be unique within the `DictionaryColor`.

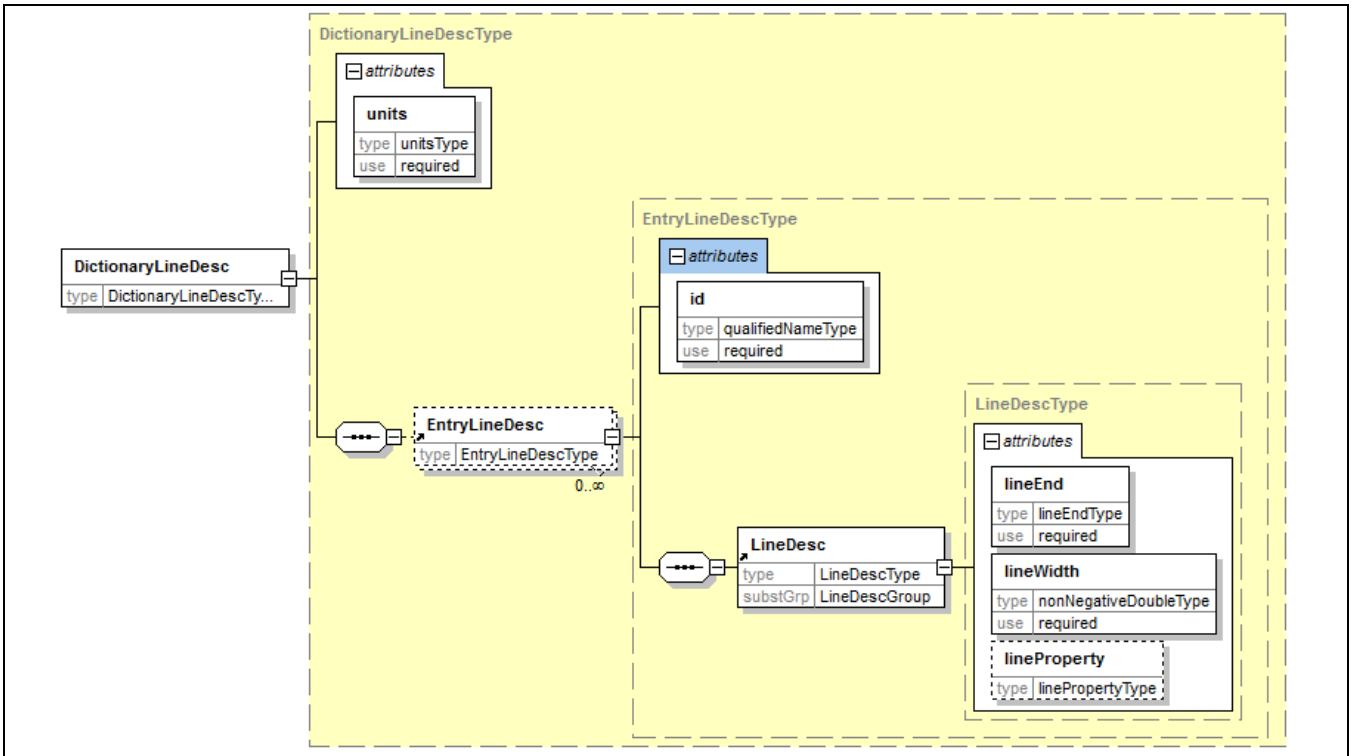


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
DictionaryColor	DictionaryColorType	An element that contains substitution group information using color description criteria, predefined by the user for reuse in the file.	0-1
EntryColor	EntryColorType	An element that establishes the individual characteristic associated with a color substitution that has been identified by the user in the IPC-2581 file.	0-n
id	qualifiedNameType	The qualified description name assigned as an id standard for XML schema color substitution.	1-1
Color	ColorType	A specific color identified through the instantiation of the three color spectrum as a part of the schema within the IPC-2581 file.	1-n

The organization of the `DictionaryColor` is accomplished in accordance with the substitution group description criteria. The `Color` description may be any combination of the three color hues (red, green and blue) at the appropriate intensity according to the values of the r, g and b attributes. Refer to section 3.5.1.1.

4.7 DictionaryLineDesc

The `DictionarylineDesc` is intended to provide lookup information on predefined line descriptions. The `DictionarylineDesc` is maintained as part of a substitution group schema. The intent is to have line descriptions available that are identified by their characteristics and a specific name (id). They may be reused throughout the file as appropriate. The name (id) of a `LineDesc` must be unique within the `DictionaryLineDesc`.



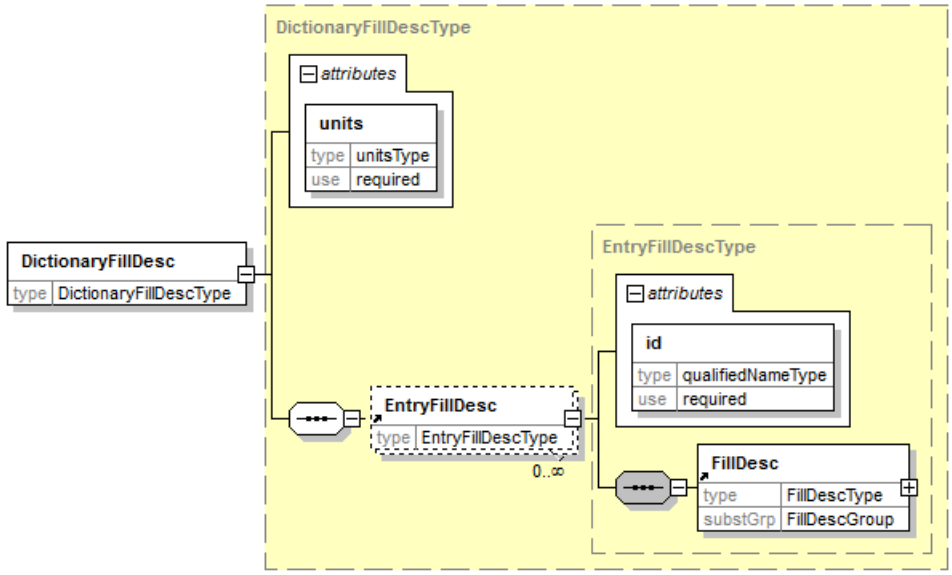
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
DictionaryLineDesc	DictionaryLineDesc Type	An element that contains substitution group information using line description criteria, predefined by the user for reuse in the file.	0-1
units	unitsType	An enumerated string that may be one of the following: MILLIMETER | MICRON | INCH.	1-1
EntryLineDesc	EntryLineDescType	An element that establishes the individual characteristic associated with a line description substitution group that has been identified by the user of the IPC-2581 file.	0-n
id	qualifiedNameType	The unique qualified name description assigned as an id for any `LineDesc` for XML schema substitution.	1-1
LineDesc	LineDescType	An element that defines the `LineEnd, LineWidth,` and `LineProperty` characteristics to become part of the substitution group for defining line descriptions.	1-n

```
<DictionaryLineDesc units = "MILLIMETER">
    <EntryLineDesc id = "FineLine">
        <LineDesc lineEnd = "ROUND" lineWidth = "0.05"/>
    </EntryLineDesc>
    <EntryLineDesc id = "MediumLine">
        <LineDesc lineEnd = "NONE" lineWidth = "0.15"/>
    </EntryLineDesc>
        <EntryLineDesc id = "HeavyLine">
        <LineDesc lineEnd = "SQUARE" lineWidth = "0.30"/>
    </EntryLineDesc>
 </DictionaryLineDesc>
```

The organization of the DictionaryLineDesc is accomplished in accordance with the substitution group description criteria. The lineDesc description defines the LineEnd and LineWidth according to the specific characteristics defined in section 3.5.5.1. The LineDescRef function is used in the body of the IPC-2581 file when a specific LineDesc has been predefined, assigned a name, and the unique "id" is referenced in the file. This feature permits the use of either a predefined LineDesc, or defining the details of a LineDesc within the file. The description in the file must contain all the features of a line description under the rules of the LineDesc definition.

4.8 DictionaryFillDesc

The DictionaryFillDesc is intended to provide lookup information on predefined fill descriptions. The DictionaryFillDesc is maintained as part of a substitution group schema. The intent is to have fill descriptions available that are identified by their characteristics and a specific name (id). They may be reused throughout the file as appropriate. The name (id) of a FillDesc must be unique within the DictionaryFillDesc.

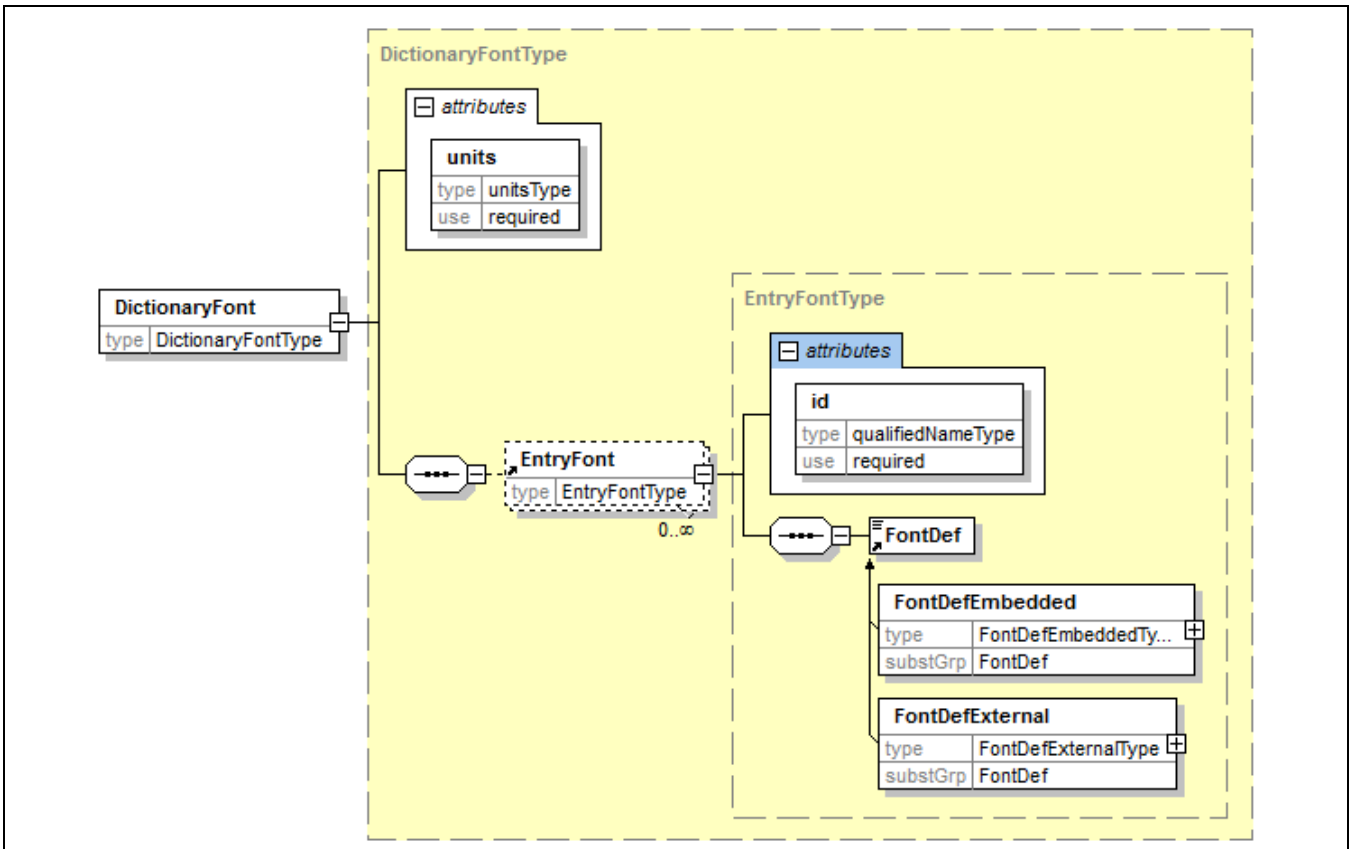


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
DictionaryFillDesc	DictionaryFillDesc Type	An element that contains substitution group information using fill description criteria, predefined by the user for reuse in the file.	0-1
units	unitsType	An enumerated string that may be one of the following: MILLIMETER | MICRON | INCH.	1-1
EntryFillDesc	EntryFillDescType	An element that establishes the individual characteristic associated with a fill description substitution group that has been identified by the user of the IPC-2581 file.	0-n
id	qualifiedNameType	The unique qualified name description assigned as an id for any FillDesc for XML schema substitution.	1-1
FillDesc	FillDescType	An element that defines the FillProperty and Color characteristics to become part of the substitution group for defining Fill descriptions.	1-n

The organization of the DictionaryFillDesc is accomplished in accordance with the substitution group description criteria. The FillDesc description defines the FillProperty and Color according to the specific characteristics defined in section 3.5.6.1. The FillDescRef function is used in the body of the IPC-2581 file when a specific FillDesc has been predefined, assigned a name, and the unique "id" is referenced in the file. This feature permits the use of either a predefined FillDesc, or defining the details of a FillDesc within the file. The description in the file must contain all the features of a Fill description under the rules of the FillDesc definition.

4.9 DictionaryFont

The DictionaryFont is intended to provide lookup information on predefined font descriptions when the standard Helvetica font is not used. The DictionaryFont is maintained as part of a substitution group schema. The intent is to have font descriptions available that are identified by their characteristics and a specific name (id). The reference is to individual Glyph characters or to a known font through reference to a URN. Font descriptions may be reused throughout the file as appropriate. The name (id) of a FontDef must be unique within the DictionaryFont.



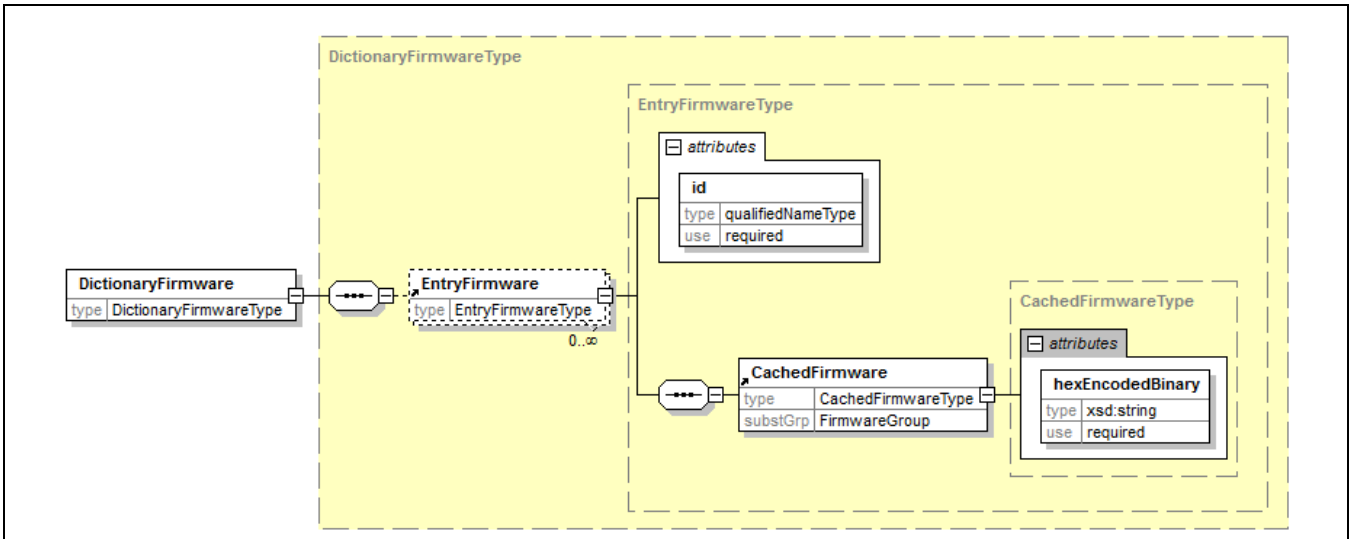
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
DictionaryFont	dictionaryFontType	An element that consists of all the named font descriptions within the IPC-2581 file.	0-n
units	unitsType	An enumerated string that may be one of the following: MILLIMETER | MICRON | INCH.	1-1
EntryFont	EntryFontType	An element that establishes the individual characteristic associated with a font substitution character or characters that have been identified by the user in the IPC-2581 file.	0-n
id	qualifiedNameType	The unique qualified name description assigned as an id for any EntryFont for XML schema substitution.	1-1
FontDef	ABSTRACT	A part of the substitution group that permits the substitution of individual Glyphs (EmbeddedFontDef) or known font types through reference of a URN (ExternalFontDef).	1-n

The organization of the DictionaryFont is accomplished in accordance with the substitution group description criteria. The FontDef description may be any character represented as a Glyph according to the specific characteristics identified in the following paragraphs. FontDef may also be a know font through reference of a URN. The FontRef function is used in the body of the IPC-2581 file when a specific font has been predefined, assigned a name, and the unique “id” is referenced in the file. This feature permits the substitution of the standard Helvetica font;

Font descriptions are only contained in the `DictionaryFont` and are not instantiated in the body of the IPC-2581 file.

4.10 DictionaryFirmware

The `DictionaryFirmware` is intended to provide lookup information on predefined `CachedFirmware`. The `DictionaryFirmware` is maintained as part of a substitution group schema. The intent is to have firmware descriptions available that are identified by their characteristics and a specific name (id). They may be reused throughout the file as appropriate. The name (id) of a `CachedFirmware` must be unique within the `DictionaryFirmware`.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
DictionaryFirmware	DictionaryFirmwareType	An element that consists of all the named `CachedFirmware` within the IPC-2581 file.	0-n
EntryFirmware	EntryFirmwareType	An element that establishes the individual characteristic associated with a Firmware entry substitution that has been identified by the user in the IPC-2581 file.	0-n
id	qualifiedNameType	The qualified description name assigned as an `id` for XML schema substitution of firmware.	1-1
CachedFirmware	CachedFirmwareType	Any CachedFirmware identified and contained in the `DictionaryFirmware` as a part of the schema within the IPC-2581 file.	1-n

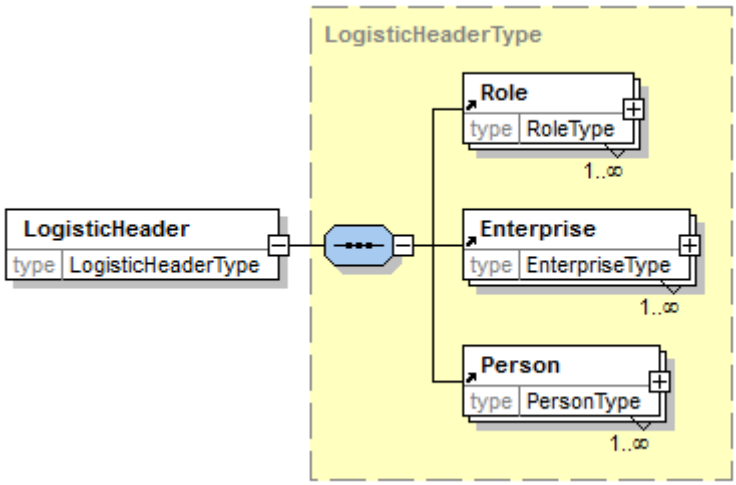
The organization of the `DictionaryFirmware` is accomplished in accordance with the substitution group description criteria. The `CachedFirmware` description may be any `hexEncodedBinary` string according to the specific characteristics identified in the following paragraphs. The `FirmwareRef` function is used in the body of the IPC-2581 file when a specific `CachedFirmware` has been predefined, assigned a name, and the unique “id” is referenced in the file. This feature permits the use of either a predefined `CachedFirmware`, or defining the details of the `Firmware` associated with a particular `Component` identified by reference designator in the `Step` section within the file. The description in the file must contain all the features of a particular `Firmware` under the rules of the particular encoded definition.

5 LOGISTIC HEADER

The `LogisticHeader` element consists of information about the owner of the IPC-2581 file. It can be used for configuration management or contact information. The enterprise is also linked to the Bill of Material and the Approved Vendor List.

5.1 LogisticHeader

The `LogisticHeader` describes information pertaining to ordering and delivery. This includes the role played by the individual providing ordering and delivery information, the title of the person responsible and the address and particulars of the enterprise.

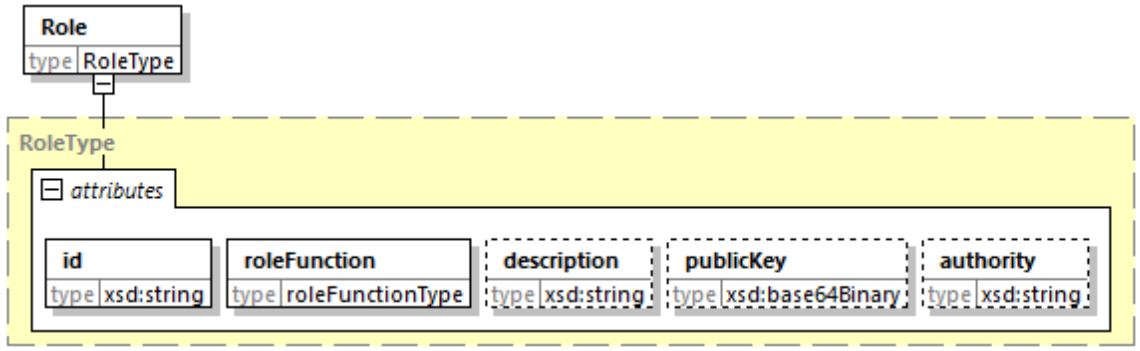


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
LogisticHeader	LogisticHeaderType	The `LogisticHeader` describes information pertaining to ordering and delivery.	1-1
Role	RoleType	Defines the type of activity within the enterprise.	1-n
Enterprise	EnterpriseType	Provides information about the company identified in the BOM or Avl schema.	1-n
Person	PersonType	Identifies the person involved in sending receiving or having anything to do in the trading partner relationship.	1-n

```
<LogisticHeader>
   <Role name = "OWNER" description = "IPC Staff" publicKey = "x6d8rf7xd90mJHR13" authority = "25XX standard development"/>
   <Enterprise id = "IPC" name = "IPC Incorporated - Association Connecting Electronics Industries" code = "57834" codeType = "CAGE"
               address1 = "3000 Lakeside Drive " city = "Bannockburn" stateProvince = "Illinois" country = "US" postalCode = "60015" phone
                = "+1-847-597-2839" fax = "+1-847-597-2839" email = "Dieterbergman@ipc.org" url = "www.ipc.org"/>
   <Person name = "Dieter W. Bergman" enterpriseRef = "IPC" title = "Director Technology Transfer" email = "Bergdi@ipc.org" phone =
               "847-597-2839" fax = "847-615-7105" mailstop = "309S " roleRef = "Owner"/>
</LogisticHeader>
```

5.2 Role

A `Role` element declares a type of activity within an `Enterprise`. The attribute values of the `Role` based on the requirements of the activities performed by the role.

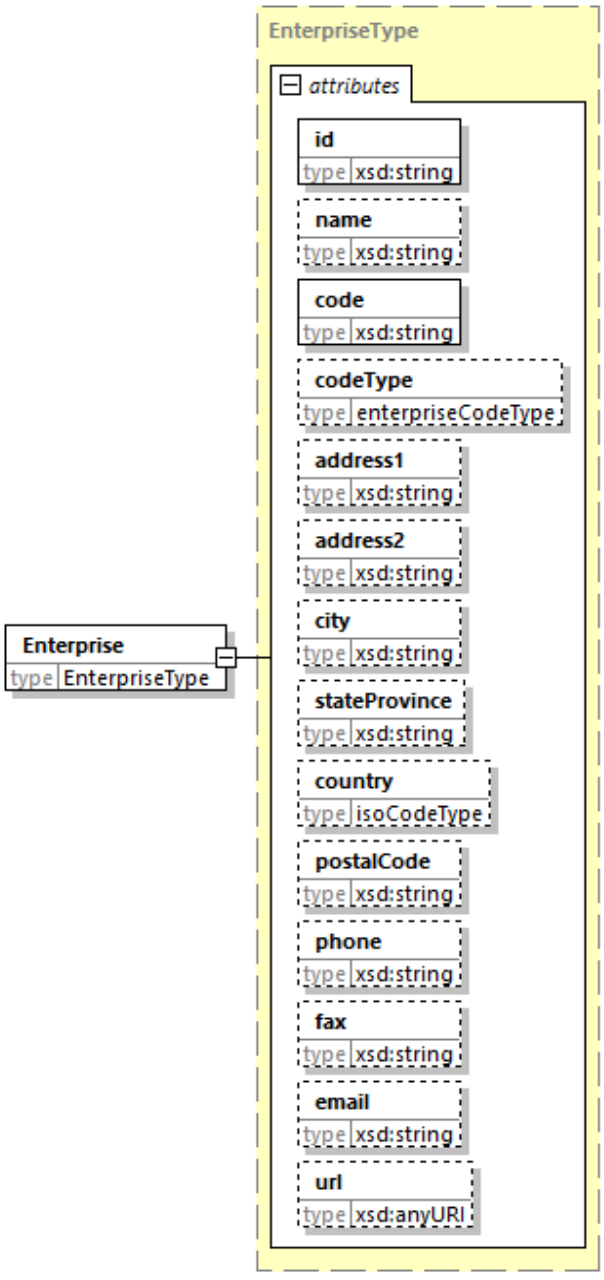


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Role	RoleType	Defines the type of activity within the enterprise.	1-n
id	string	A string that must be consistent throughout the IPC-2581 file that provides the identification of the role. A globally unique name that identifies the specific role responsibility associated with the general role descriptions.	1-1
roleFunction	roleFunctionType	The attribute that defines a globally unique identification of the role within an enterprise The description uniquely identifies a role type used by the enterprise. The name is a `roleType` (a restricted xsd:string) that must be unique within the global (top-level) namespace of the IPC-2581 file. The standard IPC role types are defined as follows: SENDER: Identifies the person sending out the IPC-2581 file. OWNER: Identifies the person who maintains the configuration management of the IPC-2581 file and has the right to increment the file history number of the IPC-2581 file. RECEIVER: Identifies the person receiving the IPC-2581 file. DESIGNER: Identifies the designer of the product described in the IPC-2581 file. ENGINEER: Identifies the engineer who is responsible for the product described in the IPC-2581 file. BUYER: Identifies the person who is responsible for payment. CUSTOMERSERVICE: Identifies the customer service representative who is responsible for the account. DELIVERTO: Identifies the person in the receiving department who takes possession of the shipment in the name of the enterprise. BILLTO: Identifies the person in the billing or purchasing department to whom the billing should be addressed. OTHER: Any other name however completing the description attribute is recommended	1-1
description	string	The `description` attribute further defines a role within an enterprise in order to highlight the specific capabilities of the `roleFunction` in harmony with the `FunctionMode` of the file. (The `description` is optional if the IPC definition is to be used, but useful in order to differentiate between several ENGINEER functions.)	0-1
publicKey	base64Binary	The `publicKey` attribute of a role holds the public encryption key if one exists for the role. The key is base64 encoded. (See IETF *RFC 1421* for the base64 algorithm) If a role `publicKey` is present it can be used instead of a `Person`/`publicKey` to encrypt data. The role's `publicKey` is used to encrypt data so only that someone with access to the role's private key can access the data.	0-1
authority	string	The access level associated with this role as defined by the system referenced by `externalConfigurationControlEntryPoint`	0-1

```
<LogisticHeader>
    <Role id = “CircuitDesign” function = “ENGINEER” description = “In charge of Impedance Control”/>
    <Role id = “LayoutPerson” function = “ENGINEER”/>
    <Role id = “PurchasingNo1” function = “BUYER” description = “To be informed of cost modification”/>
    <Role id = “LayoutEngineer” function = “DESIGNER”/>
</LogisticHeader>
```

5.3 Enterprise

The `Enterprise` element provides information about an enterprise that will be referenced within the IPC-2581 file. The attributes of the `Enterprise` element are defined as follows:



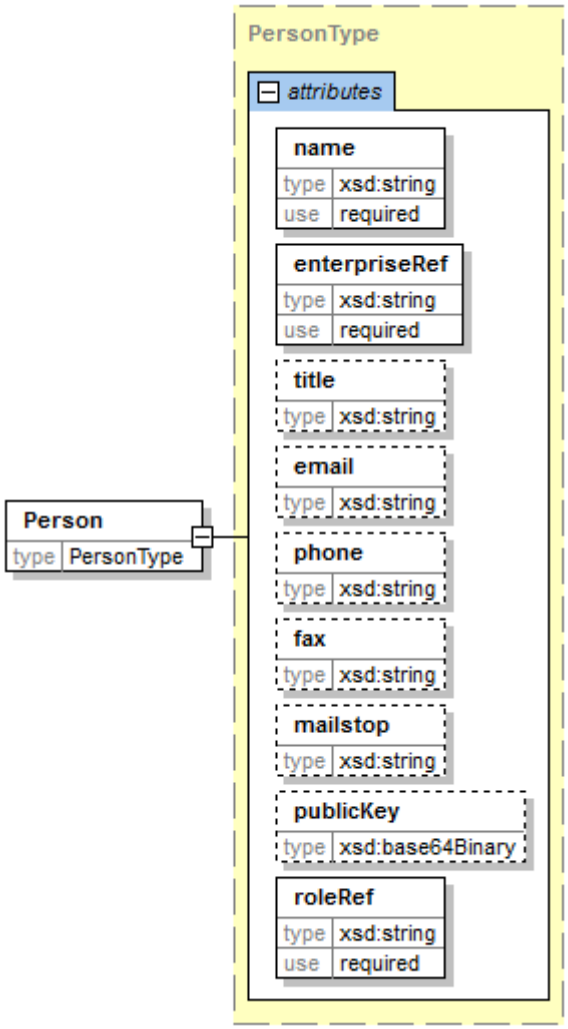
Attribute / Element Name	Attribute / Element Type	Description	Occurrence

Enterprise	EnterpriseType	Provides information about the company identified in the `Bom` or `Avl` schema.	1-n
id	string	The id uniquely identifies an enterprise throughout the IPC-2581 file. The id is a `string` data type that must be unique within the global (top-level) namespace of the IPC-2581 file. (Suggest “XYZ”, “ACME”…).	1-1
name	string	The full name of the enterprise.	0-1
code	string	Value of a CAGE or DUNS code. If no CAGE or DUNS code is available use “NONE” as the value of the code attribute.	1-1
codeType	enterpriseCodeType	One of DUNS or CAGE. The default is DUNS. If the DUNS codeType is selected, then the `code` attribute of `Enterprise` is the D-U-N-S Number of the enterprise. (See the reference to D&B D-U-N-S Number at http://www.dnb.com/) If the CAGE `codeType` is used then the CAGE code of the `Enterprise` is in the code attribute of Enterprise. (see http://www.dscc.dla.mil/offices/sourcedev/cage.html).	0-1
address1	string	The street address of the `Enterprise`.	0-1
address2	string	Additional address information for the `Enterprise`.	0-1
city	string	The `city`.	0-1
stateProvince	string	The state or province.	0-1
country	isoCodeType	The two-letter ISO country code from the ISO 3166 standard. (See ftp://info.ripe.net/iso3166-countrycodes). The default country is “US.”	0-1
postalCode	string	The postal code.	0-1
phone	string	The general phone number for the `Enterprise`.	0-1
fax	string	The phone number of the `Enterprise` fax machine.	0-1
email	string	The email address for the `Enterprise`.	0-1
url	anyURI	The Internet HTTP Web address of the `Enterprise`.	0-1

<LogisticHeader>
<Role name = “ENGINEER” description = “responsible for data in file”/>
<Enterprise id = “Acme” name = “Acme Tool and Die Company Inc.” code = “1433” codeType = “DUNNS” address1 = “7347 Concorde Ave.” address2 = “suite 42” city = “Camden” stateProvince = “NJ” country = “US” postalCode = “08780” phone = “609-458-5943” fax = “609-458-5900” email = “AcmeCorp@mindspring.com” url = “www.Acmeproducts.com”/>
<Enterprise Id = “Masters” name = “Master Spring Manufacturer” code = “NONE” address1 = “3793 Varembe Ave.” address2 = “Room 412” city = “Geneva” stateProvince = “Switzerland” country = “CH” phone = “+ 49-22-47 64 84” email = “masters@swisscom.ch”/>
</LogisticHeader>

5.4 Person

The `Person` element provides information about a person who will be referenced within the IPC-2581 file. The attributes of a `Person` element are defined as follows:



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Person	PersonType	Identifies the person involved in sending receiving or having anything to do in the trading partner relationship.	1-n
name	string	A string that uniquely identifies the person throughout the IPC-2581 file. The Name must be unique within the global (top-level) namespace of the IPC-2581 file. It may be the full legal name or a known abbreviation.	1-1
enterpriseRef	string	The name of the person's company or enterprise. If no enterprise exists, the term "SELF" should be used.	1-1
title	string	The job title of the person.	0-1
email	string	The email address of the person.	0-1
phone	string	The phone number of the person.	0-1
fax	string	The fax machine phone number of the person.	0-1
mailstop	string	The mail stop within the Enterprise, however this may be an alternate address from the Enterprise should the mail e directed somewhere else. In that event, the Enterprise shall be named, but contain no address or contact information.	0-1

publicKey	base64Binary	The publicKey attribute of a person holds the public encryption key if one exists for the person. The key is base64 encoded. (see IETF RFC 1421 for the base64 algorithm) The person's publicKey is used to encrypt data so only that person can access the data.	0-1
roleRef	string	A reference to a globally unique name (the Role "id" attribute) that identifies the specific role responsibility associated with the general "roleFunction" descriptions.	1-1

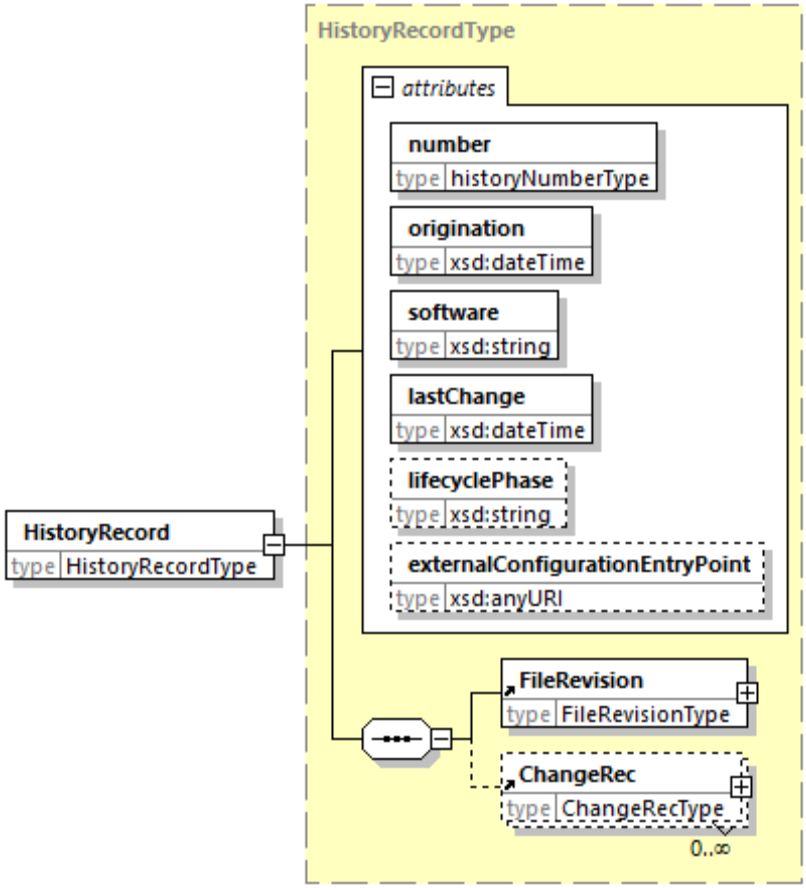
<LogisticHeader>
<Person name = "Dilbert" enterpriseRef = "Acme" email = "jdilbert@acme.com" phone = "(301) 555-1212" roleRef = "Purchasing Manager"/>
<Person name = "John Jones" enterpriseRef = "Philco Corp" title = "Consultant" email = jones@aol.com phone = "(301) 555-1212" mailstop = "37 Stringer Rd., Overland, OH, 56432" roleRef = "JJ Engineer" />
</LogisticHeader>

6 HISTORY RECORD

The History Record element consists of changes performed on the file throughout its history. Several attributes are defined as part of the `History` as well as two elements. These are file revision and change records elements.

6.1 HistoryRecord

The `HistoryRecord` element provides a sequential change number for the IPC-2581 file. The number is changed every time the controlled version of the IPC-2581 file is modified. Only the file owner is allowed to change the value of `HistoryRecord/number`. The attributes of a `HistoryRecord` element are defined as follows:



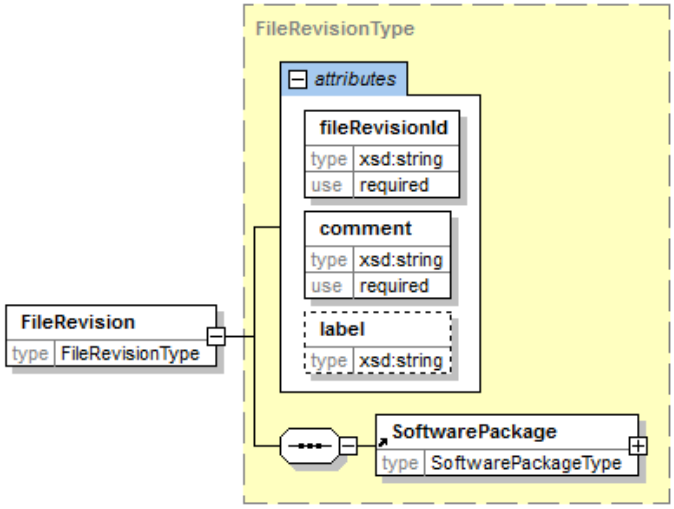
Attribute / Element Name	Attribute / Element Type	Description	Occurrence

HistoryRecord	HistoryRecordType	The `HistoryRecord` element provides a sequential change number for the IPC-2581 file. The number is changed every time the controlled version of the IPC-2581 file is modified. Also identified are the change approval conditions.	1-1
number	historyNumberType	The revision number of the IPC-2581 file. The content of this number is defined and controlled by the file owner.	1-1
origination	dateTime	The timestamp recorded when the IPC-2581 file was first created.	1-1
software	string	The name of the software tool used to create the original file.	1-1
lastChange	dateTime	The timestamp recorded when the History number was last incremented.	1-1
lifecyclePhase	string	A string indicating the lifecycle phase of the data. The states are user defined per the user's lifecycle policy, e.g. DFX review, stackup review, prototype, preproduction, production, etc.	0-1
externalConfiguration EntryPoint	anyURI	A URI referencing a configuration control system that "owns" the IPC-2581 file contents.	0-1
FileRevision	FileRevisionType	An element that tracks the changes that have been made to an IPC-2581 file. The revision identifier does not necessarily track the revision of the product but does establish the sequence and software tools used to make the changes.	1-1
ChangeRec	ChangeRecType	An element that is required to manage the configuration of the changes made to the product during its development phases and its final configuration in the field.	0-n

```
<HistoryRecord number = "Example1" origination = "2004-02-11T12:53" software = "ECAD System" lastChange =
                "2004-02-13T13:24:00">
   <FileRevision fileRevisionId = "Example1" comment = "Primitive layout positioning">
      <SoftwarePackage name = "Manual Interpretation" vendor = "IPC" revision = "none">
         <Certification certificationStatus = "ALPHA" certificationCategory = "DETAILEDDRAWING"/>
      </SoftwarePackage>
   </FileRevision>
</HistoryRecord>
```

6.2 FileRevision

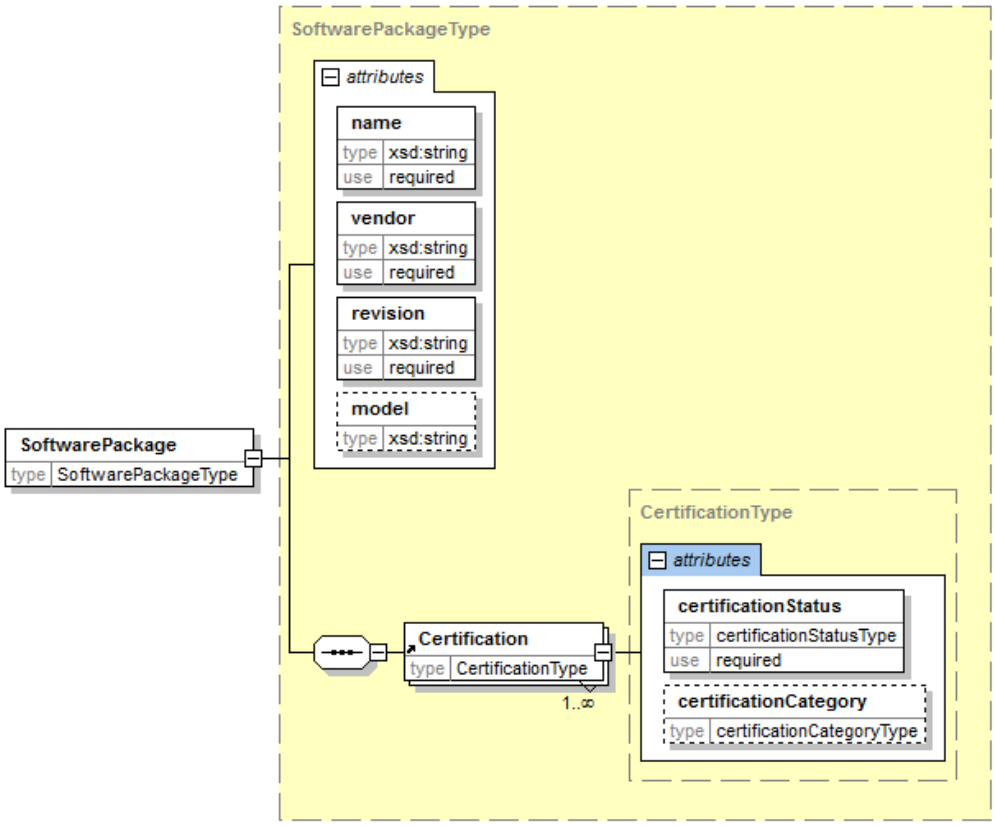
The `FileRevision` element tracks changes to the IPC-2581 file. The revision identifier does not necessarily track the revision of the product. The purpose of the `FileRevision` is to track which software tools were used to make changes to the file and the sequence in which the changes were made.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
FileRevision	FileRevisionType	An element that tracks the changes that have been made to an IPC-2581 file. The revision identifier does not necessarily track the revision of the product but does establish the sequence and software tools used to make the changes.	1-1
fileRevisionId	string	An identifier for the revision. This value may be supplied by a revision control system such as RCS, CVS, or SCCS.	1-1
comment	string	A short description of the revision, such as a changes statement entered by RCS or SCCS.	1-1
label	string	A label that can be applied to a branch head. The label can be used to associate a file revision of special significance.	0-1
SoftwarePackage	SoftwarePackage Type	A nested element, the software package that wrote this revision of the file.	1-1

6.3 SoftwarePackage

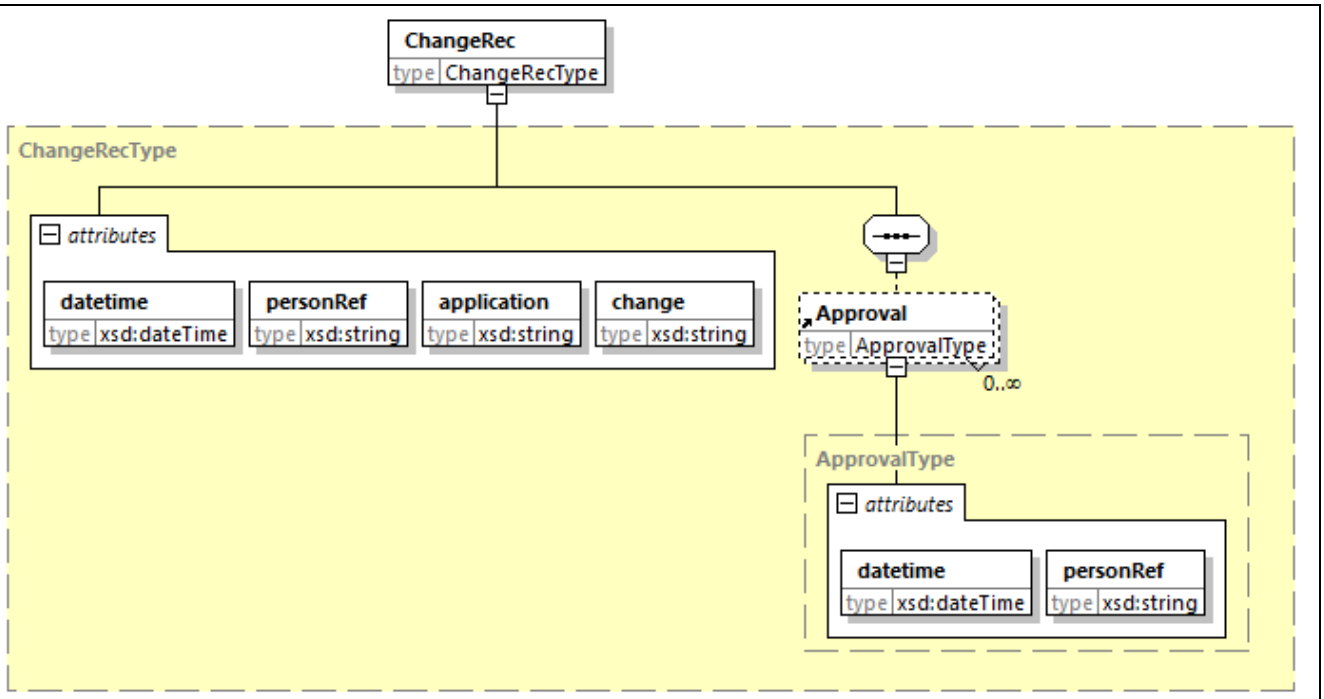
The `SoftwarePackage` element is the description of the software package that was used to create the revision to the file. This includes the revision of the software that wrote the file as well as the vendor name and platform model. Also added to the `SoftwarePackage` schema is any certification that has occurred of the software's ability to meet the requirements of the IPC-2581 standard.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
SoftwarePackage	SoftwarePackageType	A nested element, the software package that wrote this revision of the file.	1-1
name	string	The name of the software package that wrote the revision to the file.	1-1
vendor	string	The tool providers name both system and software package.	1-1
revision	string	The revision of the software that wrote the file.	1-1
model	string	The model of the software or release number.	0-1
Certification	CertificationType	The different certifications that the tool or software package has acquired.	1-n
certificationStatus	certificationStatusType	An enumerated string that defines the status as one of four types. ALPHA | BETA | CERTIFIED | SELFTEST.	1-1
certificationCategory	certificationCategory Type	The various categories that exist for certification of the type of activities related to building electronic assemblies. An enumerated string consisting of: ASSEMBLYDRAWING | ASSEMBLYFIXTUREGENERATION | ASSEMBLYPANEL | ASSEMBLYPREPTOOLS | ASSEMBLYTESTFIXTUREGENERATION | ASSEMBLYTESTGENERATION | BOARDFABRICATION | BOARDFIXTUREGENERATION | BOARDPANEL | BOARDTESTGENERATION | COMPONENTPLACEMENT | DETAILEDDRAWING | FABRICATIONDRAWING | GENERALASSEMBLY | GLUEDOT | MECHANICALHARDWARE | MULTIBOARDPARTLIST | PHOTOTOOLS | SCHEMATICDRAWINGS | SINGLEBOARDPARTLIST | SOLDERSTENCILPASTE | SPECSOURCECONTROLDRAWING | EMBEDDEDCOMPONENT | OTHER	0-1

6.4 ChangeRec

The `ChangeRec` element is the information needed for configuration management of the changes made to the product that the data file represents. The characteristics are stored by the `datecode` that the change record was executed. The information can also be used to obtain approval of a suggested change.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
ChangeRec	ChangeRecType	An element that is required to manage the configuration of the changes made to the product during its development phases and its final configuration in the field.	0-n
datetime	dateTime	The timestamp recorded when the change was made to the file.	1-n
personRef	qualifiedNameType	The pointer to the 'name' of the person listed in the `LogisticHeader` who initiated the change.	1-1
application	string	The effectivity of the change indicating when it becomes active, such as after so many completed units.	1-1
change	string	A short description of the change.	1-1
Approval	ApprovalType	A nested element that signifies who approved the suggested change submitted by the design, fabrication, assembly or test operation.	0-n
datetime	dateTime	The timestamp recorded when the change made to the file was approved.	0-n 1-1
personRef	qualifiedNameType	The pointer to the 'name' of the person listed in the `LogisticHeader` who approved the change.	1-1
<ChangeRec datetime = "2010-06-12T13:20:00" personRef = "Harry Jones" application="The change is to be implemented immediately with all produced items reworked" change="Replace the chip resistors with new part number 34-67-95, sixteen places"/> <Approval datetime = "2010-06-13T10:32:30" personRef = "John Smithy"/>			

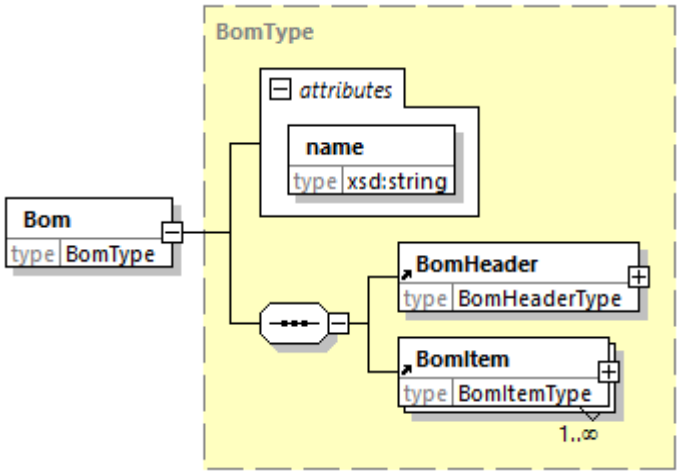
7 BOM (Material List)

This section describes the Bill of Materials for the printed board and printed board assembly. A bill of materials is a list of all the different materials and components to be used in the manufacture of the electronic assembly. The information is arranged by a specific category of material or components and then by the OEM Design Number (ODN). This is the number assigned by the owner of the file. Each ODN has a list of attributes and is accompanied by a list of the various specific uses of the materials or components on the electronic assembly, each with its private name or reference designator.

The `BOM` dataset represents the list of materials or components found on a particular board, keyed by the OEM Design Number (ODN). The original `BOM` is delivered by the owner of the file (OEM, EMS, etc.) in the early stages of the design. The Bom element is composed of the `BomHeader` and the `BomItem`

For example, ODN **348324-001** can be of package **pqfp100**, has an Internal Part Number (IPN) **30020A** and may have four occurrences on the board, labeled **U14**, **U15**, **U75**, **U76**. Each occurrence is called a Reference Designator (RefDes for short).

The IPC-2581 file can contain several `BOM` elements. Each one has a `BomHeader` sub-element with board and date/time information. The main data resides in the sub-element `BomItem`.

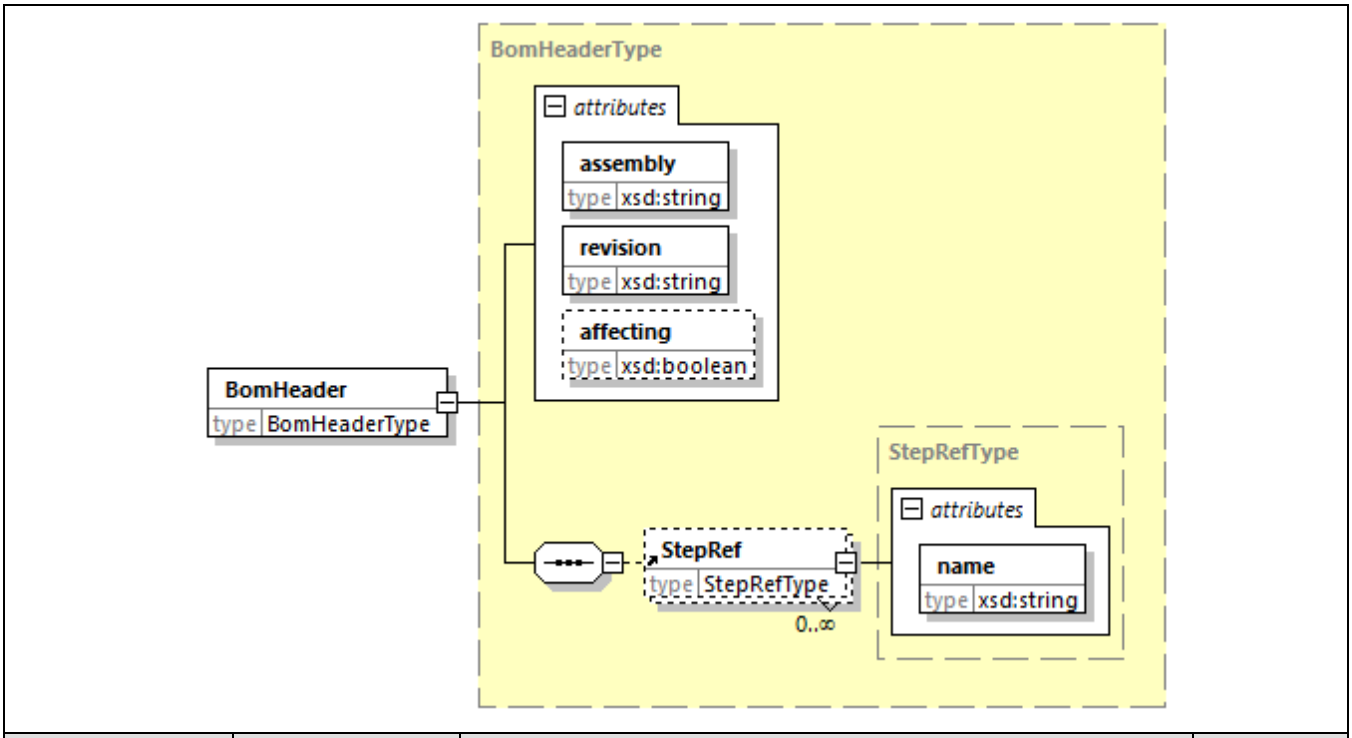


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Bom	BomType	The `Bom` element describes the Bill of Materials for the printed board and printed board assembly. A bill of materials is a list of all the different materials and components to be used in the manufacture of the electronic assembly.	1-1
name	string	The name string that identifies the specific `Bom` section. This name is referenced in the `AvlDataHeader` element.	1-1
BomHeader	BomHeaderType	A nested element containing identification and logistical information about the `Bom`.	1-1
BomItem	BomItemType	The individual elements that define the details of each of the items in the `Bom`.	1-n

```
<Bom name = "TestBoard1">
<BomHeader assembly = "Karens Design" revision = "Prototype" stepListRef = "KarensBoard"/>
      <BomItem OEMDesignNumberRef = "Fabricated" quantity = "1" pinCount = "4" category = "ELECTRICAL"     description = "Card Edge
Connector">
         <RefDes name = "J1" populate = “false”/>
         <Characteristics category = "ELECTRICAL"/>
      </BomItem>
      <BomItem OEMDesignNumberRef = "Sample1234" quantity = "1" pinCount = "8" category = "ELECTRICAL" internalPartNumber =
                  "Molex 354892" description = "Bifurcated Thru-hole connector"/>
         <RefDes name = "J2"  packageRef =”Connector1” populate = “true”/>          <Characteristics category = "ELECTRICAL"/>
      </BomItem>
      <BomItem OEMDesignNumberRef = "SOIC129867" quantity = "1" pinCount = "8" category = "ELECTRICAL" internalPartNumber =
                  "Phillips IC2436" description = "SOIC 1.27 pitch">   /
         <RefDes name = "U1" packageRef = “SOIC12” populate = “true”/>          <Characteristics category = "ELECTRICAL">
            <Textual definitionSource = "Pretested Logic" textualCharacteristicName = "Per Supplier Data Sheet"/>
         </Characteristics>
      </BomItem>
      <BomItem OEMDesignNumberRef = "CAP 24A1846" quantity = "1" pinCount = "2" category = "ELECTRICAL" internalPartNumber =
        "Phillips Cap1235" description = "3225 Surface Mount Capacitor">
         <RefDes name = "C1" packageRef = “Capacitor1”populate = “true” />         <Characteristics category = "ELECTRICAL">
            <Measured measuredCharacteristicName = "Capacitance" measuredCharacteristicValue = "20"
                engineeringUnitOfMeasure = "Microfarads" engineeringNegativeTolerance = "3 " engineering
               PositiveTolerance = "3"/>
```

7.1 BOM Header

Each `Bom` in the IPC-2581 file has a `BomHeader` element. This is a mandatory requirement as a part of the `Bom` element. The following characteristics are necessary to properly describe a `Bom`.

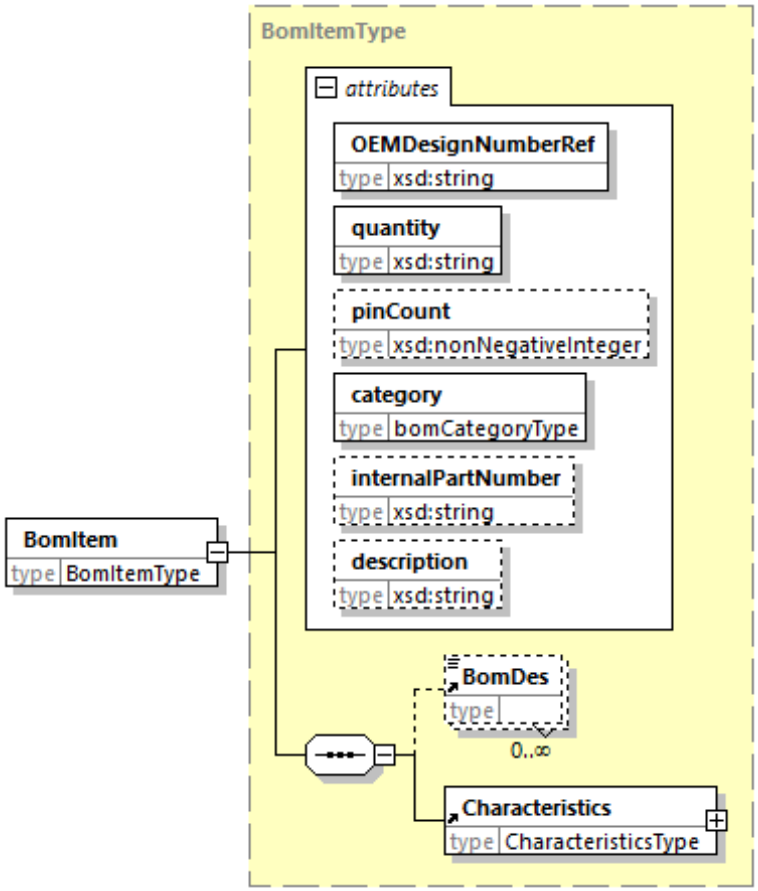


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
BomHeader	BomHeaderType	A nested element containing identification and logistical information about the `Bom`.	1-1
assembly	string	Electronic assembly name as parsed from the `Bom` file.	1-1
revision	string	Revision as parsed from the `Bom` file.	1-1
affecting	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lower case). True equals that the current BOM was used in the assembly merge process in the job and therefore was the affecting one. This is due to the fact that there can be multiple BOMs in a job, but only one has been used to affect the current top and bottom component placements sections.	0-1
stepRef	string	Identification of specific steps used to help describe the BomItem within the category of materials	0-n

7.2 BomItem

Each `BomItem` is a part of the `BomItem` list. A `BomItem` consists of a variety of attributes. `BomItem` contains the reference to the OEM Design Number (ODN), the line item of the ODN, a quantity of parts or material required, an optional pin count for component mounting, a required category of the `BomItem` an optional `internalPartNumber` (IPN), and an optional description of the `BomItem` The `BomItem` also contains two additional elements that include the list of the bill of material designators (`BomDes`) associated with the `BomItem`, and the characteristics (`Characteristics`) associated with describing how the bom item is measured, ranged, enumerated, or identified in textual information. `Multiple BomDes` lists may be maintained since there may be several designator file locations.

The attributes are shown in the following table and are a part of the `Bom` section of the IPC-2581 file.



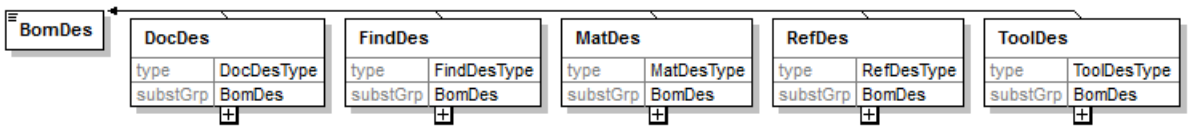
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
BomItem	BomItemType	The individual elements that define the details of each of the items in the Bom.	1-n
OEMDesignNumberRef	string	A qualified name referencing the OEM part number data.	1-1
quantity	string	The amount of the instances of this item in the board or assembly. This may be a length in the units of the IPC-2581 file, a nonNegativeInteger or an As Required (AR) designation to signify as needed quantity.	1-1
pinCount	nonNegativeInteger	The number of physical pin terminations on the package shown as a lineItem. The number includes power and ground and thermal pads whether connected or not connected to a net.	0-n
category	bomCategoryType	The category of the bomItem as an enumerated string being either ELECTRICAL | PROGRAMMABLE | MECHANICAL | MATERIAL | DOCUMENT	1-1
internalPartNumber	string	Internal or warehouse stock part identifier.	1-1
description	string	The description of the BomItem.	0-1
BomDes	ABSTRACT	A substitution group that permits the designation of Bom Items as they relate to the category identified by the LayerFunction. The substitution includes Reference designators (for components), Material designators (for Material call-outs), Documentation designators (for drawings, graphics or specifications) and Tool Designators (for tools, fixtures or templates) required to be identified in the Bill of Material. .	0-n
Characteristics	CharacteristicsType	A nested element containing descriptive strings that can be linked together and also a reference to a describing line in an external file.	1-1

SpecRef	SpecRefType	A reference to a unique `Spec` pertaining to the BomItem. Examples are a purchase specification or assembly instructions for `RefDes` or `FindDes` children, a material definition for `MatDes` children, or document details for `DocDes` children	0-n

```
<Bom name = "TestBoard1">
    <BomHeader assembly = "Karens Design" revision = "Prototype"/>
     <BomItem OEMDesignNumberRef = "Sample1234" quantity = "1" pinCount = "8" category = "ELECTRICAL"     internalPartNumber =
                "Molex 354892" description = "Bifurcated Thru-hole connector">
        <RefDes name = "J2" packageRef = Connector1" populate = "true"/>
        <Characteristics category = "ELECTRICAL"/>
     </BomItem>
     <BomItem OEMDesignNumberRef = "SOIC129867" quantity = "1" pinCount = "8" category = "ELECTRICAL" internalPartNumber =
                 "Phillips IC2436" description = "SOIC 1.27 pitch">
        <RefDes name = "U1" packageRef = SOIC12" populate = "true"/>
        <Characteristics category = "ELECTRICAL">
           <Textual definitionSource = "Pretested Logic" textualCharacteristicName = "Per Supplier Data Sheet"/>
        </Characteristics>
     </BomItem>
     <BomItem OEMDesignNumberRef = "CAP 24A1846" quantity = "1" category = "ELECTRICAL" internalPartNumber =
                  "Phillips Cap1235" description = "3225 Surface Mount Capacitor">
        <RefDes name = "C1" packageRef = "Capacitor1" populate = "true"/>
        <Characteristics category = "ELECTRICAL">
           <Measured measuredCharacteristicName = "Capacitance" measuredCharacteristicValue = "20"
                        engineeringUnitOfMeasure = "Microfarads" engineeringNegativeTolerance = "3"
              engineeringPositiveTolerance = "3"/>
        </Characteristics>
     </BomItem>
```

7.2.1 BomDes

The Bill of Material Designations (BomDes) are intended to identify items that may be contained in the Bom and need a specific designation based on their characteristics. There are five groups of designations which include documents, mechanical components, materials, electrical components, and tools/fixtures.

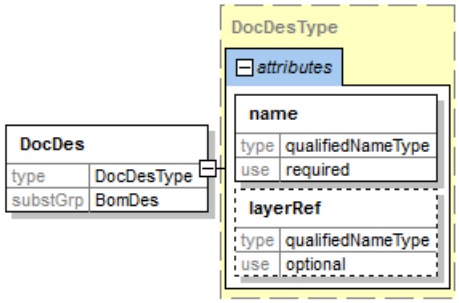


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
BomDes	BomDescType	A substitution group that permits the designation of Bom Items as they relate to the category identified by the LayerFunction.	0-n
DocDes	DocDesType	A Bom item designation used to identify "**Documents**" that are contained in the Bill of Material and may be referenced throughout the IPC-2581 file.	0-n
FindDes	FindDesType	A Bom item designation used to identify Find Numbers associated with "**Mechanical Parts**" that are contained in the Bill of Material but have no electrical connectivity to the PCB. Unlike "Reference Designators" that are unique instances of a part number (1:1), Find Numbers are often collections of the same part number (1:many), and in the Mechanical World are often tied to balloon notes and work instructions.	0-n
MatDes	MatDesType	A Bom item designation used to identify "**Materials**" that are contained in the Bill of Material and may be referenced throughout the IPC-2581 file.	0-n
RefDes	RefDesType	A Bom item designation used to identify "**Components**" that are contained in the Bill of Material and may be referenced throughout the IPC-2581 file. Components are defined as any parts that have electrical connectivity via one or more pins/pads, even though their category may be "mechanical" e.g. a grounded connector shield.	0-n

ToolDes	ToolDesType	A Bom item designation used to identify "**Tools & Fixtures**" that are contained in the Bill of Material and may be referenced throughout the IPC-2581 file.	

7.2.1.1 DocDes

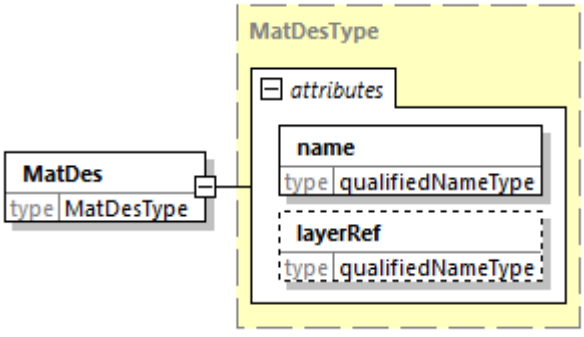
A Bom item designation used to identify "**Documents**" that are contained in the Bill of Material and may be referenced throughout the IPC-2581 file.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
DocDes	BomDesType	A Bom item designation used to identify the properties of specific documents contained in a IPC-2581 file.	0-n
Name	qualifiedNameType	A unique name assigned as a designation for documents consisting of a combination of letter(s) and number(s)	1-1
layerRef	qualifiedNametype	A reference to the identification of the unique name assigned to a specific layer with a DOCUMENT layerFunction that contains graphical data that is considered documentation	0-1

7.2.1.2 MatDes

A Bom item designation used to identify "**Materials**" that are contained in the Bill of Material and may be referenced throughout the IPC-2581 file.

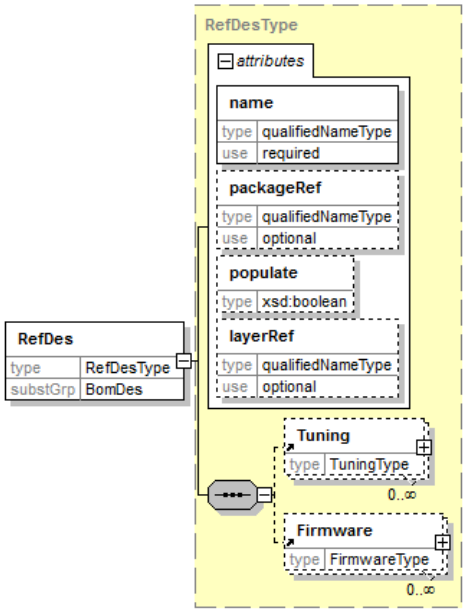


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
MatDes	BomDesType	A Bom item designation used to identify the properties of specific materials contained in a IPC-2581 file.	0-n
name	qualifiedNameType	A unique name assigned as a designation for materials consisting of a combination of letter(s) and number(s)	1-1
layerRef	qualifiedNameType	A reference to the identification of the unique name assigned to a specific layer which identifies a layerFunction describing the materials being identified in the CadData.	0-1

7.2.1.3 RefDes

A Bom item designation used to identify "**Components**" that are contained in the Bill of Material and may be referenced throughout the IPC-2581 file.

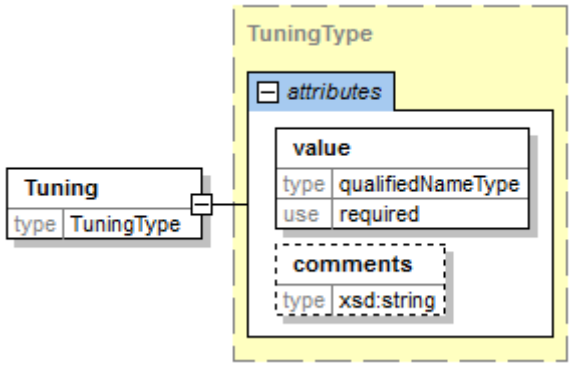
The `RefDes` is an element that represents the specific reference designator associated with a component that becomes a part of the electronic assembly. This is a mandatory requirement for all `BomItems` that have a reference designator associated with their ELECTRICAL descriptions. In this instance the standard set of reference designator letters **shall** be used. i.e., R = Resistor, C = Capacitor, CR = Diode etc. The prefix letter M **shall** be used for all MECHANICAL parts, (e.g., terminals, latches, heatsinks, etc.), and S for all Software PROGRAMMABLE `BomItems`.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
RefDes	RefDesType	A nested element containing the reference designator strings for the individual parts identified in the file for a one to one relationship with the quantity listed for the `BomItem`.	0-1
name	qualifiedNameType	The qualifiedNameType that identifies the reference designator used as the attribute `refDes` of the `Component` element in `Step`.	1-1
packageRef	qualifiedNameType	The name assigned to the package describing the physical outlines, documentation, and land patterns features related to package pin assignment.	0-1
populate	boolean	An enumerated string is either "true" or "false" (part of the 3WC standard –must be lower case). True equals that the `RefDes` was populated; False indicates that it was not. True is the default.	0-1
layerRef	qualifiedNameType	A reference to the identification of the unique name assigned to a specific layer which identifies a layerFunction describing the components being identified in the CadData mounted on the top, bottom or internal to the printed board.	0-1
Tuning	TuningType	A number of conditions that determine any adjustment that is needed for a particular `BomItem`.	0-n
Firmware	FirmwareType	A nested element containing descriptive strings that can be linked together to describe the software implementation for an individual `BomItem`.	0-n

7.2.1.3.1 Tuning

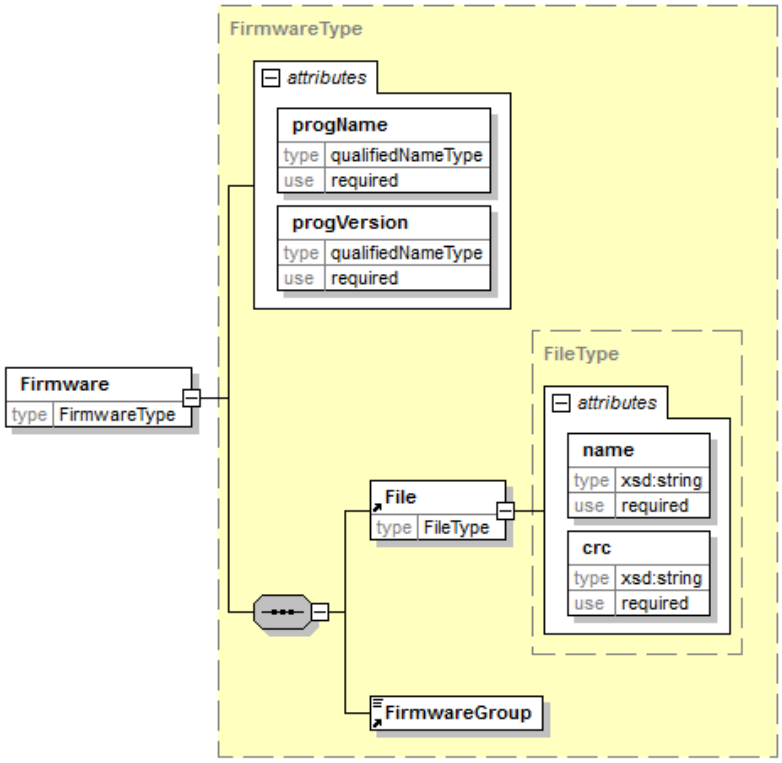
The `Tuning` element represents conditions that determine any adjustment that is needed for a particular `BomItem`.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Tuning	TuningType	A number of conditions that determine any adjustment that is needed for a particular `BomItem`.	0-n
value	qualifiedNameType	The value of the tuning characteristics and its relationship to the other prescribed values.	1-1
comments	string	Any instructions or comments needed to prescribe the proper tuning.	0-n

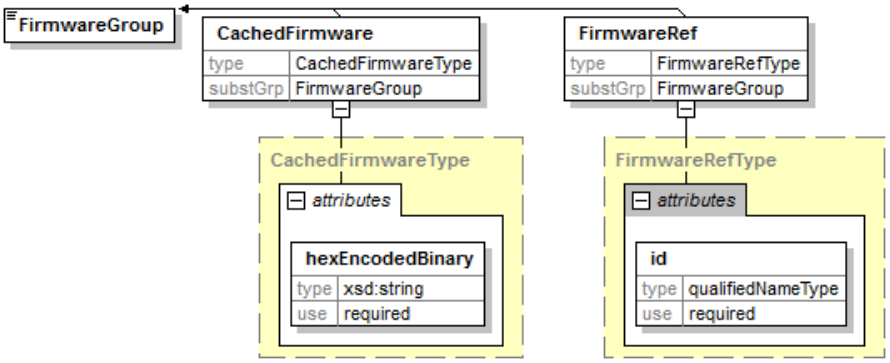
7.2.1.3.2 Firmware

A nested element containing descriptive strings that can be linked together to describe the software implementation for an individual `BomItem` and associates the characteristics of the specific reference designators to which the programmable information is to be included.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Firmware	FirmwareType	A nested element containing descriptive strings that can be linked together to describe the software implementation for an individual `BomItem`.	0-n
progName	qualifiedNameType	Name of the program to be embedded in the BomItem.	0-1
progVersion	qualifiedNameType	Version of program or revision status.	0-1
File	FileType	Computer file containing the program code to be inserted.	1-1
name	string	Name of the file.	1-1
crc	string	Certification status to verify correct program elements.	1-1
FirmwareGroup	ABSTRACT	A substitution group that specifies the `CachedFirmware` which has been stored by the user in the `DictionaryFirmware`.	0-1

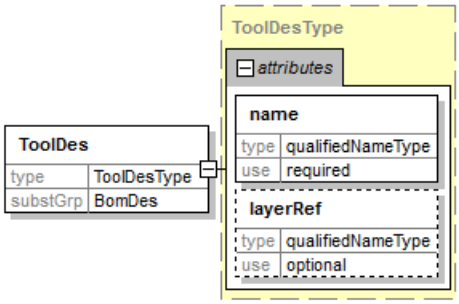
7.2.1.3.3 Firmware Group



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
FirmwareGroup	Abstract	A substitution group that specifies the `CachedFirmware` which has been stored by the user in the `DictionaryFirmware`.	0-1
CachedFirmware	CashedFirmwareType	The firmware description needed by a particular component that becomes part of the predefined firmware in the `DictionaryFirmware`.	1-1
hexEncodedBinary	string	An attribute defining the binary code that **shall** be added to a particular component and which is contained in the `DictionaryFirmware`.	1-1
FirmwareRef	FirmwareRefType	The specific reference to firmware previously defined and contained in the `DictionaryFirmware`.	1-1
id	qualifiedNameType	The id uniquely identifies a firmware that has been stored in the firmware dictionary and is used throughout the IPC-2581 file. The id is a `qualifiedNameType` data type (a restricted xsd: string) that must be unique within the global (top-level) namespace of the IPC-2581 file.	1-1

7.2.1.4 ToolDes

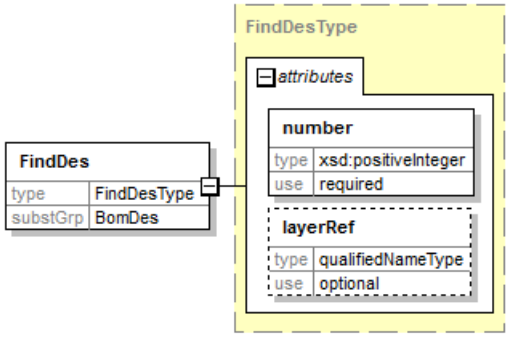
A Bom item designation used to identify "**Tools or Fixtures** " that are contained in the Bill of Material and may be referenced throughout the IPC-2581 file.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
ToolDes	ToolDesType	A Bom item designation used to identify the properties of specific tools or fixtures contained in a IPC-2581 file.	0-n
name	qualifiedNameType	A unique name assigned as a designation for tools or fixtures consisting of a combination of letter(s) and number(s)	1-1
layerRef	qualifiedNameType	A reference to the identification of the unique name assigned to a specific layer which identifies a layerFunction describing the tools or fixtures being identified in the CadData.	0-1

7.2.1.5 FindDes

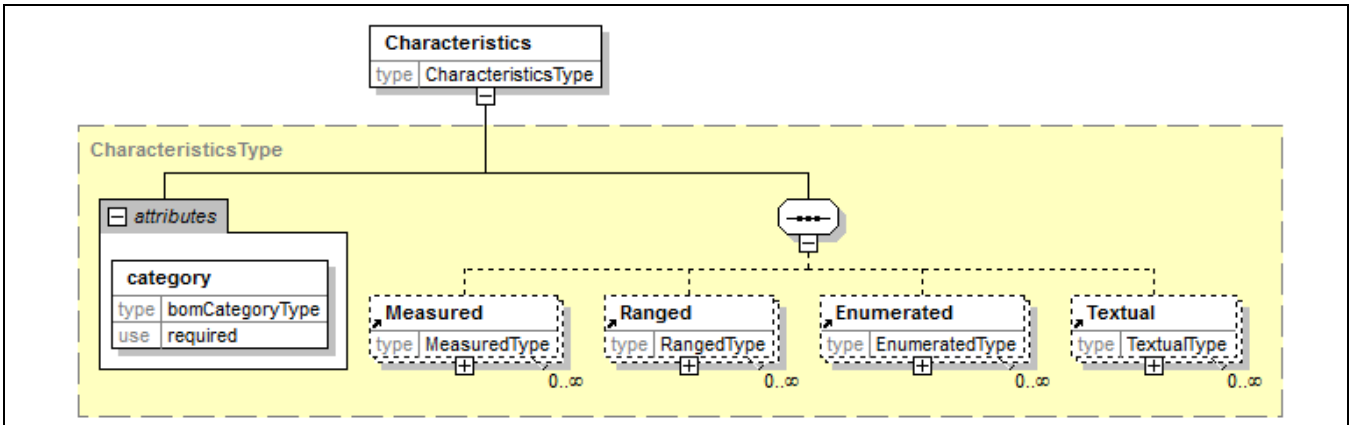
A Bom item designation used to identify "**Mechanical Parts**" that are contained in the Bill of Material and may be referenced throughout the IPC-2581 file.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
FindDes	FindDesType	A Bom item designation used to identify Find Numbers associated with "**Mechanical Parts**" that are contained in the Bill of Material but have no electrical connectivity to the PCB.	0-n
number	positiveInteger	A unique number assigned as a Find Number, being a positive integer. Unlike "Reference Designators" that are unique instances of a part number (1:1), Find Numbers are often collections of the same part number (1:many), and in the Mechanical World are often tied to balloon notes and work instructions.	1-1
layerRef	qualifiedNametype	A reference to the identification of the unique name assigned to a specific layer which identifies a layerFunction describing the mechanical parts being identified in the CadData.	0-1

7.2.2 Characteristics

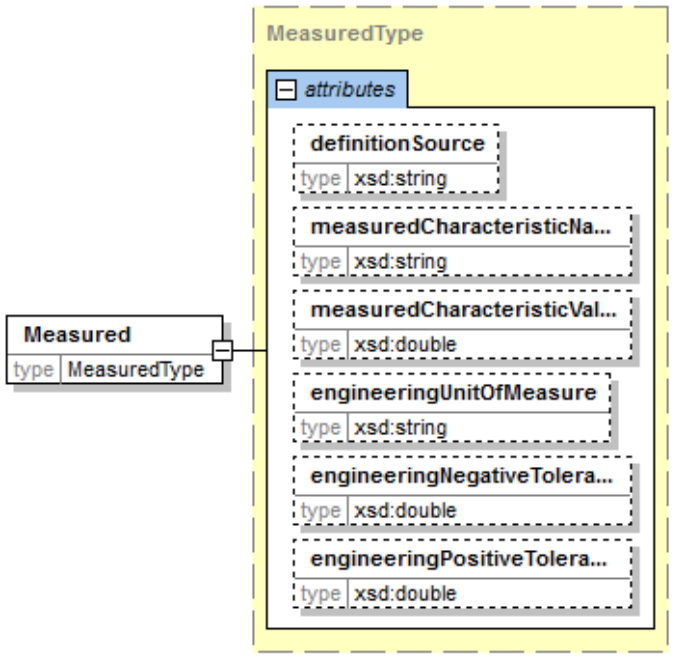
A group of specific characteristics applicable to a particular `BomItem`; they all relate to one of the categories to which the `BomItem` belongs. Each characteristic has its own level of requirements and are defined under the major element `Characteristics`.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Characteristics	CharacteristicsType	A nested element containing descriptive strings that can be linked together and also a reference to a describing line in an external file.	1-1
category	bomCategoryType	Defines the type of material or component category to which the `BomItem` belongs. The bomCatogoryType enumerations may be either ELECTRICAL | PROGRAMMABLE | MECHANICAL | MATERIAL | DOCUMENT	1-1
Measured	MeasuredType	A nested element containing descriptive strings that can be linked together to describe the measurable characteristics for an individual `BomItem`.	0-n
Ranged	RangedType	A nested element containing descriptive strings that can be linked together to describe the tolerances or ranges of variation that can be applied to the characteristics for an individual `BomItem`.	0-n
Enumerated	EnumeratedType	A nested element containing descriptive strings that can be linked together to describe those values that define the numerical characteristics and the source for that information for an individual `BomItem`.	0-n
Textual	TextualType	A nested element containing descriptive strings that can be linked together to describe any textual characteristics and their source for an individual `BomItem`.	0-n

7.2.2.1 Measured

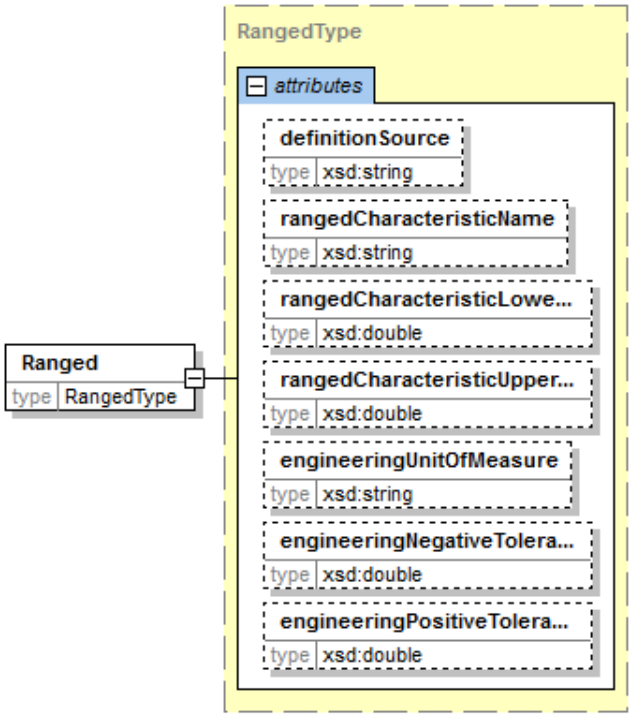
The `Measured` elements are those properties that when linked together describe the measurable characteristics for an individual `BomItem`. These characteristics provide the nominal value and also include the tolerances on the measurement.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Measured	MeasuredType	A nested element containing descriptive strings that can be linked together to describe the measurable characteristics for an individual `BomItem`.	0-n
definitionSource	string	The source of the information about the measurements.	0-1
measuredCharacteristicName	string	A unique name applied to the characteristic (e.g., capacitance)	0-1
measuredCharacteristicValue	double	The value of the measured property at its nominal or target value.	0-1
engineeringUnitOfMeasure	string	The engineering unit of measure.	0-1
engineeringNegativeTolerance	double	The negative tolerance on the value identified.	0-1
engineeringPositiveTolerance	double	The positive tolerance on the value identified.	0-1

7.2.2.2 Ranged

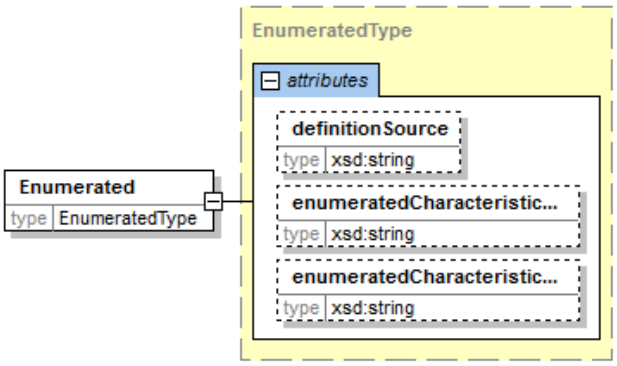
The `Ranged` elements are those properties that when linked together describe the ranges that a `BomItem` must meet. These range characteristics include the upper and lower limit of the range as well as the tolerances on the measurement. These values are compared against those that have been measured to ascertain that the `BomItem` is within specifications.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Ranged	RangedType	A nested element containing descriptive strings that can be linked together to describe the tolerances or ranges of variation that can be applied to the characteristics for an individual `BomItem`.	0-n
definitionSource	string	The source of the information about the measurement ranges.	0-1
rangedCharacteristicName	string	A unique name applied to the characteristic.	0-1
rangedCharacteristicLowerValue	double	The lower value of a ranged characteristic.	0-1
rangedCharacteristicUpperValue	double	The upper value of a ranged characteristic.	0-1
engineeringUnitOfMeasure	string	The engineering unit of measure.	0-1
engineeringNegativeTolerance	double	Any negative tolerance that can may be applied to the range in order to affix the target value of the `BomItem`.	0-1
engineeringPositiveTolerance	double	Any positive tolerance that can may be applied to the range in order to affix the target value of the `BomItem`.	0-1

7.2.2.3 Enumerated

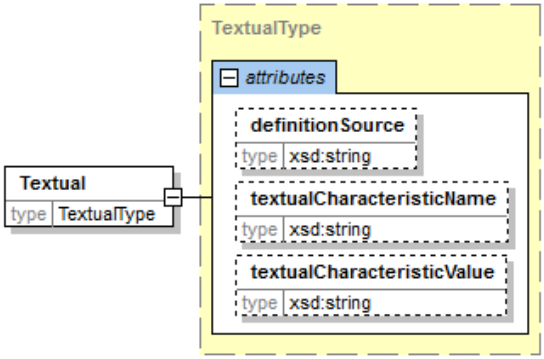
The `Enumerated` elements are those properties that, when linked together, describe the enumerated value of a `BomItem` as well as the source of that information.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Enumerated	EnumeratedType	A nested element containing descriptive strings that can be linked together to describe those values that define the numerical characteristics and the source for that information for an individual `BomItem`.	0-n
definitionSource	string	The source of the information about the enumerated value.	0-1
enumeratedCharacteristicName	string	A unique name applied to the characteristic.	0-1
enumeratedCharacteristicValue	string	The enumerated value identified for the `BomItem`.	0-1

7.2.2.4 Textual

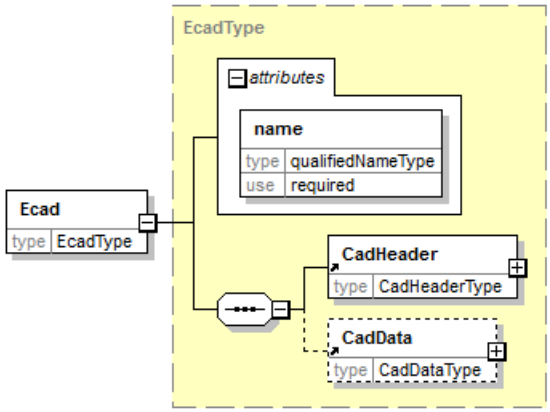
The `Textual` elements are those properties that when linked together describe the textual value of a `BomItem` as well as the source of that information.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Textual	TextualType	A nested element containing descriptive strings that can be linked together to describe any textual characteristics and their source for an individual `BomItem`.	0-n
definitionSource	string	The source of the information about the textual value.	0-1
textualCharacteristicName	string	A unique name applied to the characteristic.	0-1
textualCharacteristicValue	string	The textual value identified for the `BomItem`.	0-1

8 ELECTRONIC COMPUTER AIDED DESIGN (ECAD)

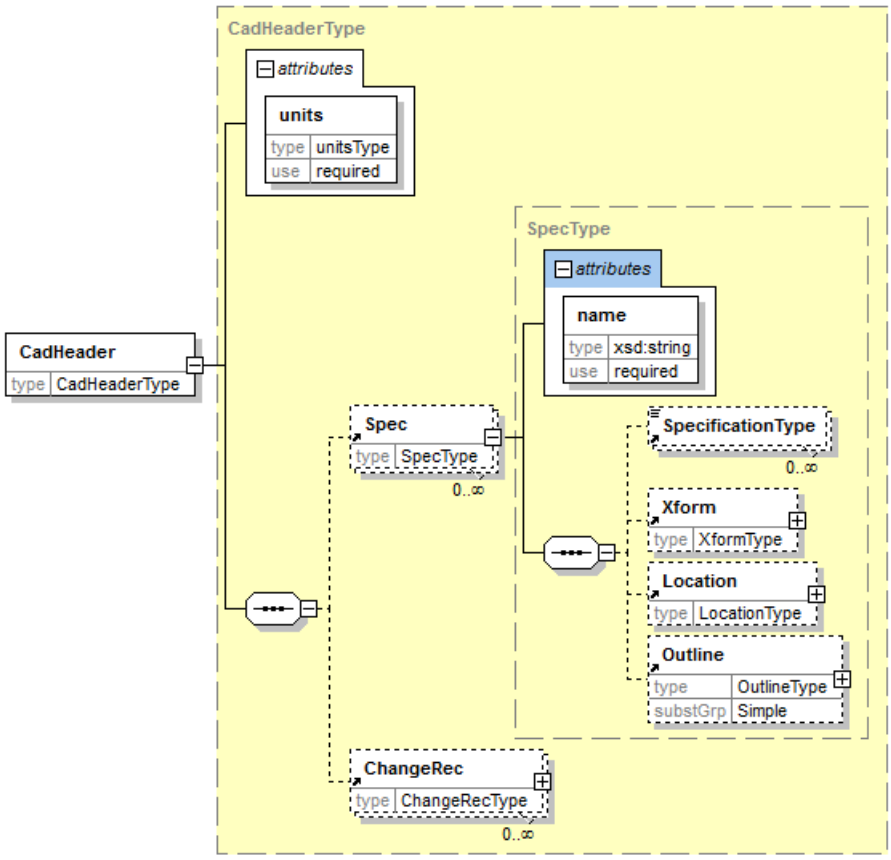
The `Ecad` section describes the Computer Aided Design data of the job, including all the graphical description of the layers, component location, panel design, etc. In most cases, the `Ecad` section is by far the largest body of data inside the IPC-2581 file. To understand how the `Ecad` section is organized, it is important to be familiar with the `Layers` and `Step` elements.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Ecad	EcadType	The `Ecad` section describes the Computer Aided Design data of the job, including all the graphical description of the layers, component location, panel design, etc.	1-1
name	qualifiedNameType	The qualified name related to identify the specific `Ecad` information based on the electronic assembly(s) in the IPC-2581 file.	1-1
CadHeader	CadHeaderType	A nested element containing identification and information about the description of the data in the file.	1-1
CadData	CadDataType	A nested element containing the actual Cad data describing the printed board and printed board assembly characteristics.	0-1

8.1 CadHeader

The `CadHeader` element is mandatory. Inside the `CadHeader` there are general attributes that describe the printed board, characteristics of the assembly, tooling, or documentation as defined by the `CadData`. The `Spec` element helps to define special instructions, tolerances, location where these apply, or any other comments that relate to the `CadData`. Dimensions are defined by the units attribute and may be in imperial (inch) or metric (mm or µm) units. Once defined the unit descriptions apply to all ecad data; however the units may be modified by the incorporation of a specification such that the specification may apply to the information or instructions provided by the Spec. This condition has no impact on the Units established for the IPC-2581 file and only applies to the specification details. Any feature imported from any dictionary file **shall** be configured in the same units as defined by the `Ecad` units attribute.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
CadHeader	CadHeaderType	A nested element containing identification and information about the description of the data in the file.	1-1
units	unitsType	An enumerated string that may be one of the following: MILLIMETER | MICRON | INCH.	1-1
Spec	SpecType	Contains information related to the characteristics of the printed board, the assembly, material or any other layer function and may include the values for conductor width and spacing, compliance requirements, dielectric information, thermal characteristics or other attributes and instructions. The specification is instantiated where it is required through the unique name of the specification which establishes the SpecType and the detailed requirements	0-n
ChangeRec	ChangeRecType	A nested element containing identification and information about the description of the data in the Ecad section of the file.	0-n

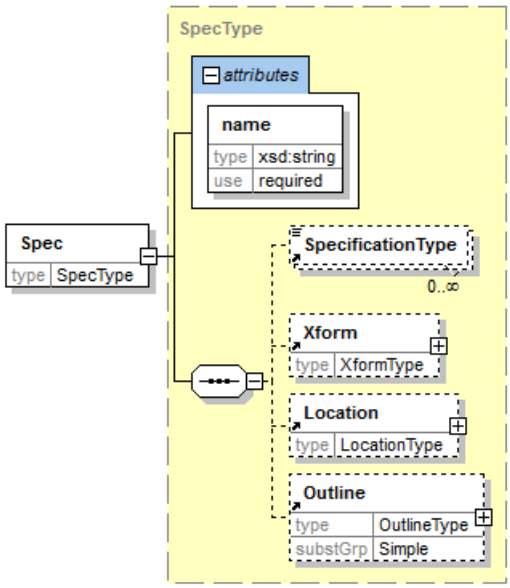
8.1.1 Spec

The `Spec` element contains various types of information related to the characteristics of the board or assembly. The concepts may include values for:

- width, spacing, component spacing, component to edge, etc.
- impedance, capacitance and resistance values.
- dimensions between edge/feature/hole to edge/feature/hole, V Cut or backdrill dimensions.

The instance of invoking a specification is provided through the option of including the elements as a part of a characteristic describing the board, assembly or a feature of their manufacturing such as the material or a physical condition of the final product. The Spec provides that information that clearly identifies the goal conditions of the final product.

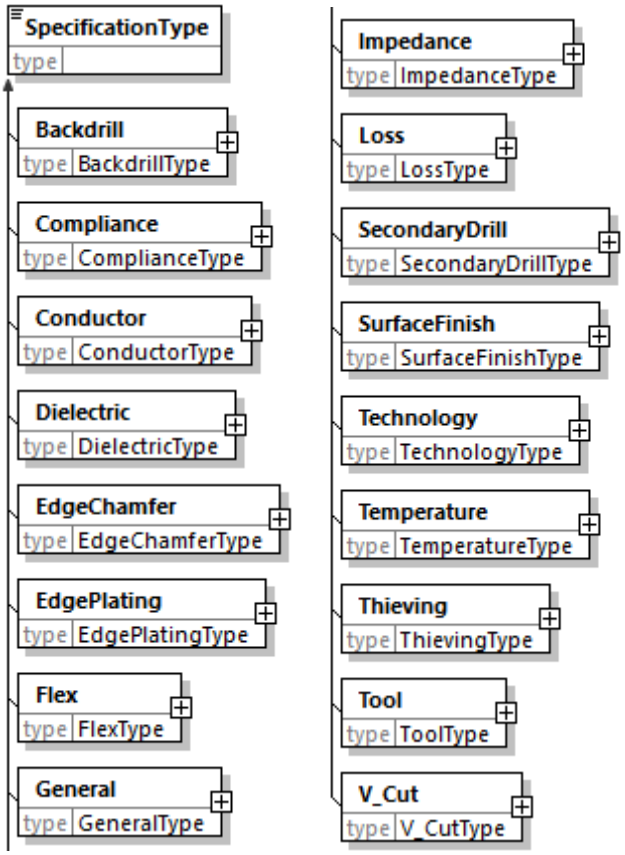
Up to two sub-elements may describe the specific `Location` or `Outline` where the spec is to be applied. The xform sub-element permits changing of the original point of origin, scaling of features, rotation or mirror imaging of the spec requirements.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Spec	SpecType	Contains information that applies to the board or assembly fabrication and relates to the end product capability of the product being produced. Instructions may include various values for conductor width and spacing, component spacing, component to board edge conditions, impedance values, capacitance values, resistance values, compliance to regulations and other requirement dimensions.	0-n
name	qualifiedNameType	The qualified name of the specification. The name must be unique and should indicating to which product in the file the specification pertains.	1-1
SpecificationType	ABSTRACT	A substitution group that permits the substitution of the SpecificationType with the specific elements and their characteristics that identify the purpose of the specification in order to focus the application of the descriptions to where they should apply. Examples of the specificationtypes include: Backdrill, Compliance, Conductor, Dielectric, General, Impedance, Technology, Temperature, Tool, and V_Cut.	0-n
Xform	XformType	An element that provides the ability to reset the point of origin of a predefined area that can be scaled, mirror imaged or rotated. See 3.3	0-1
Location	LocationType	The image defined by `Outline` or a pre-defined image is located to identify where the specification applies. The image may have been reorientated by the Xform.	0-1
Outline	OutlineType	A nested element containing a specific area(s) to which the specification(s) apply. The `Outline` is a closed polygon configuration	0-n

8.1.1.1 SpecificationType

The SpecificationType is an Abstract substitution feature that permits various specification functions to be defined within the concept of the Spec element.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
SpecificationType	ABSTRACT	A substitution group that permits the substitution of the SpecificationType with any specific elements and their characteristics that identify the purpose of the specification in order to focus the application of the descriptions to where they should apply. Possible specificationtypes are Backdrill, Compliance, Conductor, Dielectric, EdgeChamfer, EdegPlating, Flex, General, Impedance, Loss, SecondaryDrill, SurfaceFinish, Technology, Temperature, Thieving, Tool, and V_Cut.	0-n
Backdrill	BackdrillType	The identification of a SpecificationType related to the backdrill of plated-through holes or vias in order to remove some of the plating without disturbing the interconnection structure.	0-n
Compliance	ComplianceType	The identification of a SpecificationType related to compliance to regulations that are being imposed on the materials or the final product of the board, or the assembly. The requirements may be regional or customer based and may include reclamation or disposal.	0-n
Conductor	ConductorType	The identification of a SpecificationType related to the conductor used to carry signals throughout the printed board assembly describing the characteristics, surface finish or other properties related to the conductor shape or material.	0-n
Dielectric	DielectricType	The identification of a SpecificationType related to the dielectric properties of the material which includes their electrical characteristics as well as their physical makeup such as resin or reinforcement styles and also the processing temperature that they can withstand.	0-n
EdgeChamfer	EdgeChamferType	The identification of a SpecificationType related to the board edges where a chamfer is required across the depth (cross section) of the board.	0-n
EdgePlating	EdgePlatingType	The identification of a SpecificationType related to the board edges where plating is required across the depth (cross section) of the board.	0-n

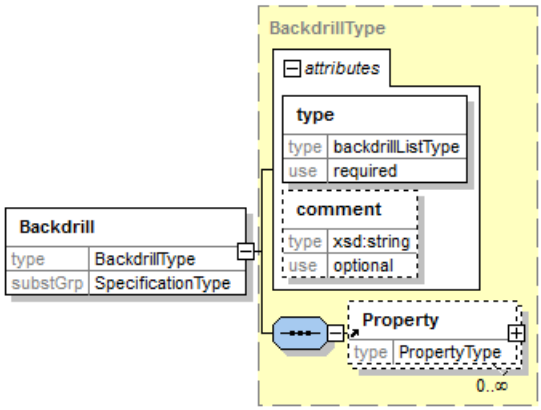
Flex	FlexType	The identification of a SpecificationType related to properties associated with a flex zone.	0-n
General	GeneralType	The identification of a SpecificationType related to any general requirements that may be necessary to impose on any feature or product identified through the layer function and which allows instantiation of a spec onto the requirements	0-n
Impedance	ImpedanceType	The identification of a SpecificationType related to the impedance requirements of a stackup construction or an printed board assembly in order to define the characteristics or the relationship to material separation or layering	0-n
Loss	LossType	The identification of a SpecificationType related to the loss requirements of a stackup construction or a printed board assembly.	0-n
SecondaryDrill	SecondaryDrillType	The identification of a SpecificationType related to the creation of a countersink or counter bore of a hole	0-n
SurfaceFinish	SurfaceFinishType	The identification of a SpecificationType related to the surface finish material of a layer or object	0-n
Technology	TechnologyType	The identification of a SpecificationType related to the technology being used in the construction of the printed board or the assembly or any process concepts that help identify the process control sensitive areas	0-n
Temperature	TemperatureType	The identification of a SpecificationType related to the thermal capabilities or requirements that are necessary in order to meet the coefficient of expansion of material or parts in an effort to establish reliability conditions	0-n
Thieving	ThievingType	The identification of a SpecificationType related to the thieving keep in or keep out area in the manufacture of a printed board, printed board panel, or a printed board pallet.	0-n
Tool	ToolType	The identification of a SpecificationType related to the tooling used in the manufacture of a printed board, printed board panel, printed board assembly or a printed board pallet and may be a single cutter or a template used for ensuring process consistency	0-n
V_Cut	V_CutType	The identification of a SpecificationType related to the scoring of printed board panels or pallets in order to allow the removal of individual board or coupon products identifying the cut size or material remaining.	0-n

Those Specification types that may not be associated with a single physical layer, such as Backdrill, Compliance, General or Temperture, can be referenced by `SpecRef` elements from the parent `Stackup` element, to help make the recipient aware of their existence. For example:

```
<Stackup name="ECAD_CROSS_SECTION" overallThickness="1.623" tolPlus="0" tolMinus="0" whereMeasured="OTHER">
  <SpecRef id=”Backdrills”/>
  <SpecRef id=”Compliances”/>
  <StackupGroup name="RootGroup" thickness="1.623" tolPlus="0" tolMinus="0">
    <StackupLayer layerOrGroupRef="TOP" thickness="0.0559" tolPlus="0" tolMinus="0" sequence="1">
  :
  etc
```

8.1.1.2 Backdrill

The Backdrill specification feature is intended to define the dimensions and characteristics of backdrilling some of the plated through holes.

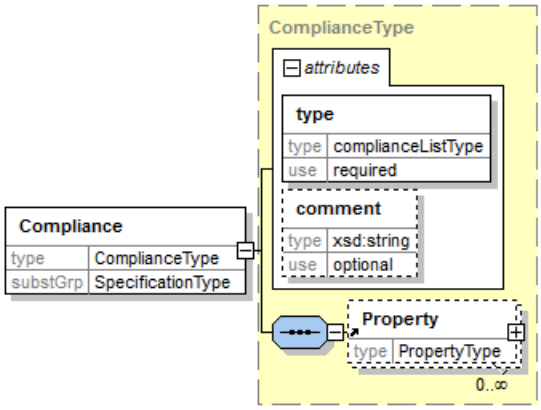


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Backdrill	BackdrillType	The identification of a SpecificationType related to the backdrill of plated-through holes or vias in order to remove some of the plating without disturbing the interconnection structure.	0-n
type	backdrillListType	A set of enumerated string descriptions that identify the condition of the backdrill requirement. They consist of START_LAYER | MUST_NOT_CUT_LAYER | MAX_STUB_LENGTH | OTHER. If OTHER is noted a comment attribute is required.	1-1
comment	string	A description of the requirements for backdrilling that are different or supplement the backdrillListType enumerations	0-1
Property	PropertyType	A nested element containing numerical attribute descriptions that relate to the backdrill requirements and permit the references to alternate conditions that may apply. Unit descriptions apply only to the specificationType and do not impact the requirements of the overall IPC-2581 file Unit descriptions. Refer to Section 3.4.10 for the definition of `Property`.	0-n

```
<Spec name=”backdrill_1”>
    <Backdrill type=”START_LAYER”>
            <Property layerOrGroupRef=”L12”/>
    </Backdrill>
    <Backdrill type=”MUST_NOT_CUT_LAYER”>
            <Property layerOrGroupRef=”L8”/>
    </Backdrill>
    <Backdrill type=”MAX_STUB_LENGTH”>
            <Property value=”0.5” unit=”MM” plusTol=”0” minusTol=”5” tolPercent=“true”/>
    </Backdrill>
</Spec>
```

8.1.1.3 Compliance

The Compliance specification feature is intended to define the legal requirements imposed under certain conditions and to define the characteristics that need to be adhered to in developing or providing the product defined in the IPC-2581 file.

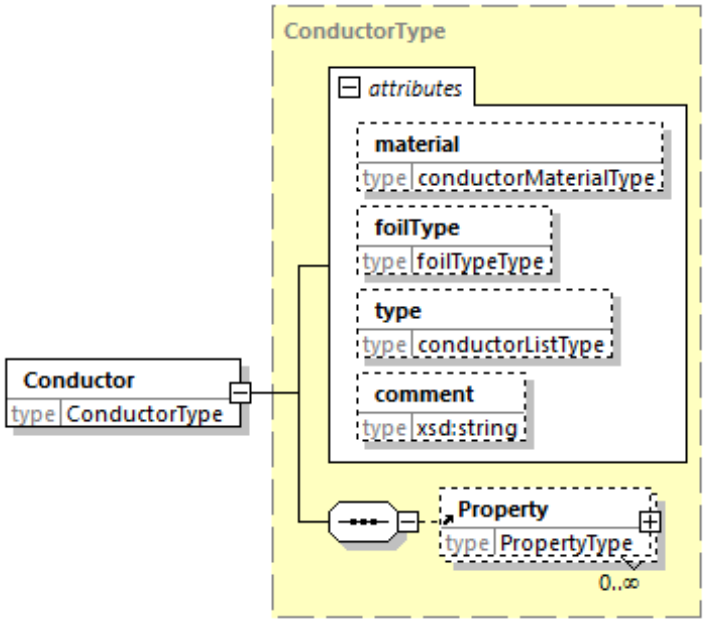


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Compliance	ComplianceType	The identification of a SpecificationType related to the backdrill of plated through holes or vias in order to remove some of the plating without disturbing the interconnection structure.	0-n
type	complianceListType	A set of enumerated string descriptions that identify the condition of the compliance requirement. They consist of ROHS | CONFLICT_MINERALS | WEEE | REACH | HALOGEN_FREE | OTHER. If OTHER is noted, a comment attribute is required.	1-1
comment	string	A description of the requirements for meeting compliance regulations that are different or supplement the complianceIListType enumerations	0-1
Property	PropertyType	A nested element containing numerical attribute descriptions that relate to the compliance requirements and permit the references to alternate conditions that may apply. Unit descriptions apply only to the specificationType and do not impact the requirements of the overall IPC-2581 file Unit descriptions. Refer to Section 3.4.10 for the definition of `Property`.	0-n

```
<Spec name=”compliances”>
    <Compliance type=”ROHS”>
    </Compliance>
    <Compliance type=”HALOGEN_FREE”>
    </Compliance>
</Spec>
```

8.1.1.4 Conductor

The Conductor specification feature is intended to define the specific characteristics of conductors, or conductor surfaces that need to be provided in the product defined in the IPC-2581 file.



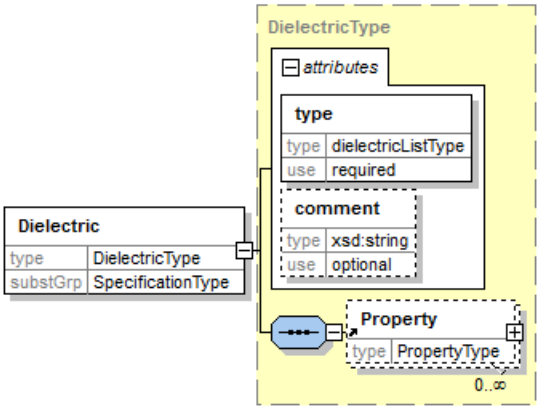
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Conductor	ConductorType	The identification of a SpecificationType related to the conductor characteristics such as dimensions, surface conditions and electrical properties.	0-n
material	conductorMaterialType	A set of enumerated string descriptions that identify the conductor material. Possible values are COPPER | SILVER_INK | CARBON_INK | CONDUCTIVE_INK | OTHER	0-1
foilType	foilTypeType	A set of enumerated string descriptions that identify a copper foil type in accordance with IPC-4652. Possible values are CU-E1 |CU-E2 |CU-E3 | CU-W5 | CU-W6 | CU-W7 |CU-W8 |NI/E1 | CU-E10 | CU-E11 | OTHER	0-1
type	conductorListType	A set of enumerated string descriptions that identify the condition of the conductor requirement. They consist of CONDUCTIVITY | PRODUCT_NAME | SURFACE_ROUGHNESS_UPFACING | SURFACE_ROUGHNESS_DOWNFACING | SURFACE _ROUGHNESS_TREATED | ETCH_FACTOR | FINISHED_HEIGHT | WEIGHT | OTHER. If OTHER is noted a comment attribute is required.	0-1
comment	string	A description of the requirements for conductors that are different or supplement the conductorListType enumerations	0-1
Property	PropertyType	A nested element containing numerical attribute descriptions that relate to the `type` attribute, and permit the references to alternate conditions that may apply. Unit descriptions apply only to the specificationType and do not impact the requirements of the overall IPC-2581 file Unit descriptions. Refer to Section 3.4.10 for the definition of `Property`.	0-n

```
<Spec name=”copper_foil2”>
    <Conductor type=”WEIGHT”>
        <Property value=”0.5”  unit=”OZ”/>
    </Conductor>
    <Conductor foilType=”CU-E1”/>
    <Conductor type=”SURFACE_ROUGHNESS_UPFACING”>
        <Property value=”3”  unit=”MICRON” refUnit=”RMS”/>
    </Conductor
</Spec>
```

```
<Spec name=”coin_material”>
    <Conductor material=”COPPER”/>
</Spec>

<Spec name=”conductive_ink”>
    <Conductor material=”SILVER_INK”/>
    <Conductor type=”PRODUCT_NAME”>
        <Property text=”PE410”/>
    </Conductor>
</Spec>
```

8.1.1.5 Dielectric

The Dielectric specification feature is intended to define the specific characteristics of dielectric material, or characteristics that need to be provided in the product defined in the IPC-2581 file.



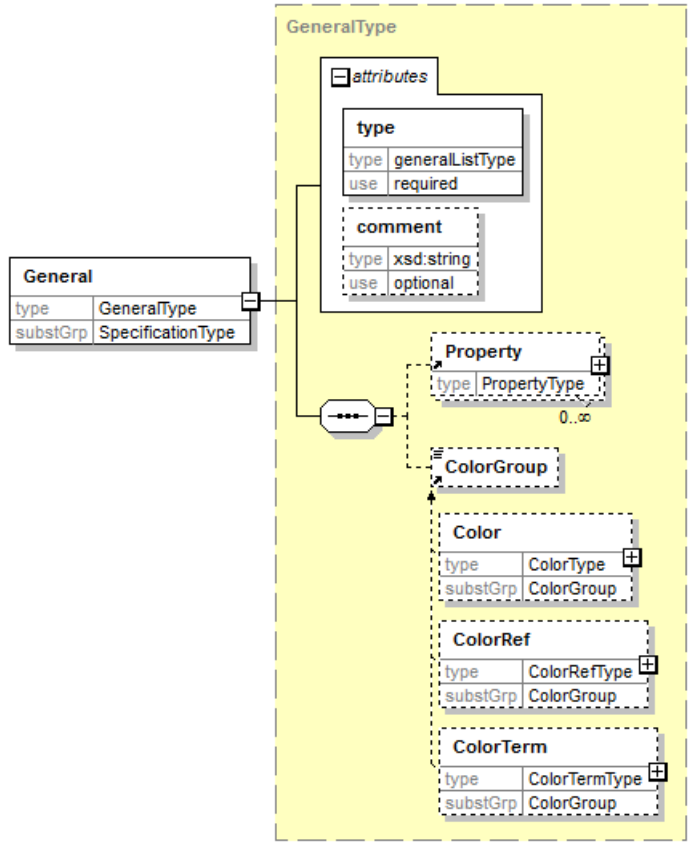
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Dielectric	DielectricType	The identification of a SpecificationType related to the dielectric characteristics such as electrical properties, physical makeup, and process temperature restrictions.	0-n
type	dielectricListType	A set of enumerated string descriptions that identify the condition of the conductor requirement. They consist of DIELECTRIC_CONSTANT | LOSS_TANGENT | GLASS_STYLE | RESIN_CONTENT | Tg_DSC | Tg_DMA | Tg_TMA | Td | SLASH_ IPC4101 | SLASH _IPC4103 | PRODUCT_NAME | OTHER. If OTHER is noted a comment attribute is required.	1-1
comment	string	A description of the requirements for dielectric material that are different or supplement the dielectricListType enumerations	0-1
Property	PropertyType	A nested element containing numerical attribute descriptions that relate to the dielectric requirements and permit the references to alternate conditions that may apply. Unit descriptions apply only to the specificationType and do not impact the requirements of the overall IPC-2581 file Unit descriptions. Refer to Section 3.4.10 for the definition of `Property`.	0-n

```
<Spec name="dielec4_15">
  <Dielectric type="DIELECTRIC_CONSTANT">
        <Property value="4.150" refValue="10.0e9" refUnit="Hz"/>
        <Property value="4.148" refValue="15.0e9" refUnit="Hz"/>
  </Dielectric>
  <Dielectric type="LOSS_TANGENT">
        <Property value="0.035" refValue="10.0e9" refUnit="Hz"/>
        <Property value="0.034" refValue="15.0e9" refUnit="Hz"/>
  </Dielectric>
  <Dielectric type="RESIN_CONTENT">
        <Property value="50"  unit="PERCENT"/>
  </Dielectric>
  <Dielectric type="GLASS_STYLE">
        <Property value="1080"/>
  </Dielectric>
</Spec>
```

```
<Spec name="dielectric_layer5">
  <Dielectric type="PRODUCT_NAME">
        <Property text="I-Speed"/>
  </Dielectric>
  <Dielectric type="DIELECTRIC_CONSTANT">
        <Property value="3.64"/>
  </Dielectric>
  <Dielectric type="LOSS_TANGENT">
        <Property value="0.006"/>
  </Dielectric>
  <Dielectric type="Tg_DSC">
        <Property value="180" unit="CELCIUS" refUnit="MAX"/>
  </Dielectric>
  <Dielectric type="Td">
        <Property value="360" unit="CELCIUS"/>
  </Dielectric>
  <Dielectric type="SLASH _IPC4101">
        <Property value="98"/>
        <Property value="99"/>
        <Property value="101"/>
        <Property value="126"/>
  </Dielectric>
</Spec>
```

8.1.1.6 General

The General specification feature is intended to define any general characteristics related to the final product and may be grouped as electrical, thermal, material requirements or identify instructions or standards that influence the product defined in the IPC-2581 file.

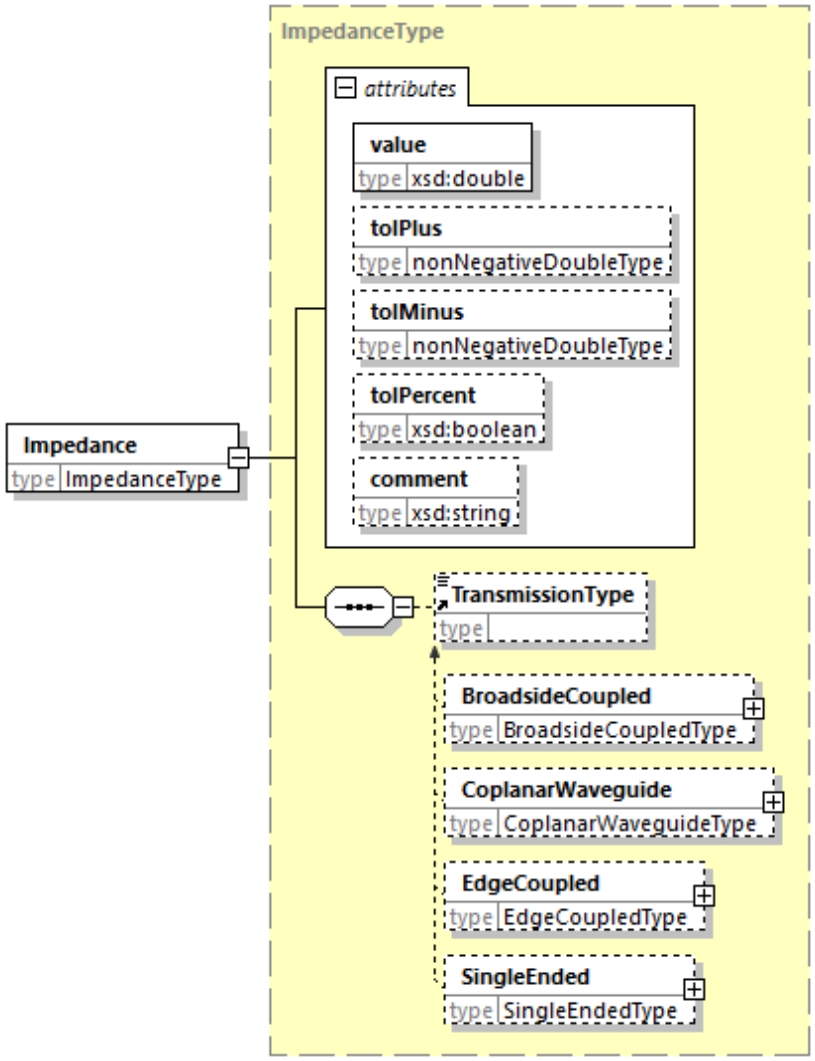


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
General	GeneralType	The identification of a SpecificationType related to any general characteristics such as electrical, thermal or material properties, as well as special instructions or standards that the product must meet.	0-n
type	generalListType	A set of enumerated string descriptions that identify any general requirements. The general groupings consist of ELECTRICAL | THERMAL | MATERIAL | INSTRUCTION | STANDARD | CONFIGURATION | OTHER. If OTHER is noted a comment attribute is required.	1-1
comment	string	A description of any general requirements that are a different category or supplement the generalListType enumerations	0-1
Property	PropertyType	A nested element containing numerical attribute descriptions that relate to the general requirements and permit the references to alternate conditions that may apply. Unit descriptions apply only to the specificationType and do not impact the requirements of the overall IPC-2581 file Unit descriptions. Refer to Section 3.4.10 for the definition of `Property`.	0-n
ColorGroup	ABSTRACT	An optional substitution group that permits assigning a color through instantiating the three basic colors, or by providing a reference to a predefined `Color` in `DictionaryColor`, or choosing a basic color term from an enumerated list.	0-1

```
<Spec name="Standards">
    <General type="STANDARD">
            <Property text="Telcordia Standard XYZ" refUnit="CLASS" refValue="3" />
    </General>
    <General type="STANDARD">
            <Property text="IPC-4101" refUnit="SECTION" refText="8.2.1.13" />
    </General>
</Spec>
<Spec name="Soldermask">
    <General type="MATERIAL">
              <Property text="SMOBC-LPI"/>
              <ColorTerm name="GREEN"/>
     </General>
</Spec>
<Spec name="Build Notes">
    <General type="INSTRUCTION">
              <Property text="Line neckdowns are intentional, build as is." refUnit="ITEM" refValue="1" />
              <Property  text="100 ohm differential pair traces that violate the specified separation are
                        intentional, build as is." refUnit="ITEM" refValue="2" />
     </General>
</Spec>
```

8.1.1.7 Impedance

The Impedance specification feature is intended to define any impedance characteristics related to the final product and includes the transmission and structure characteristics that the product defined in the IPC-2581 file must meet.

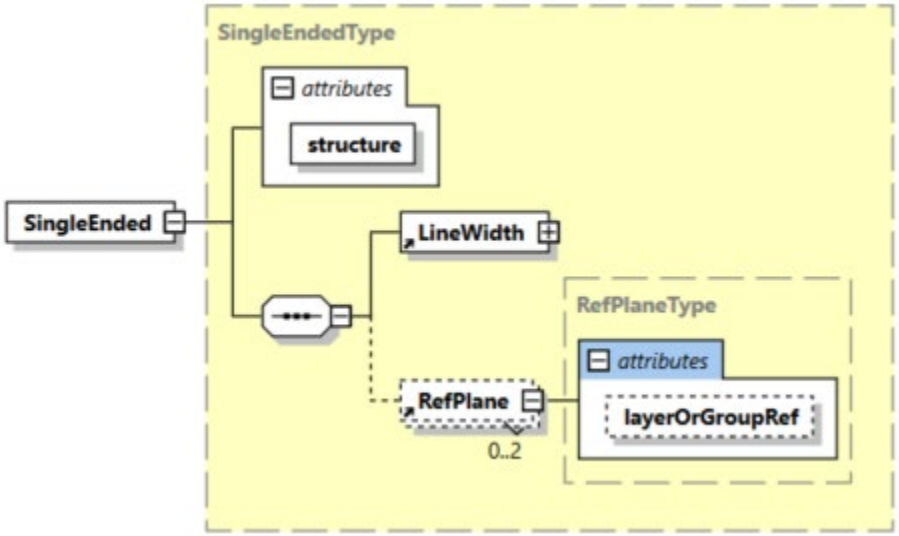


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Impedance	ImpedanceType	The identification of a `SpecificationType` related to the impedance characteristics and structure such as the electrical requirements, relationship to surrounding plane and specific structure properties that the product must meet.	0-n
value	double	The value of impedance, in ohms. For example 10e3 = 10 Kilohms	1-1
tolPlus	nonNegativeDouble	The plus tolerance that may be applied to the nominal number defined in the Value attribute setting the upper control limit.	0-1
tolMinus	nonNegativeDouble	The minus tolerance that may be applied to the nominal number defined in the Value attribute setting the lower control limit.	0-1
tolPercent	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lower case). True means that the attribute(s) `tolPlus` and `tolMinus` are a percentage if present. False indicates that they represent an absolute value (nonNegativeDouble). The Default is “false”.	0-1

TransmissionType	ABSTRACT	A substitution group that permits the substitution of one of the following elements that contain physical properties of the `Impedance`: `EdgeCoupled`, `SingleEnded`, `BroadsideCoupled`, `CoplanarWaveguide`. The `TransmissionType` shall be omitted if, and only if, this `Spec` is referenced from a `LogicalNet`, if the `LogicalNet` spans multiple layers and the physical properties change per layer.	0-1
comment	string	A description of any impedance requirements that are different or supplemental	0-1

8.1.1.7.1 SingleEnded

A permitted substitution of `TransmissionType` as a child of the `Impedance` element, to identify a single ended transmission type of impedance specification.



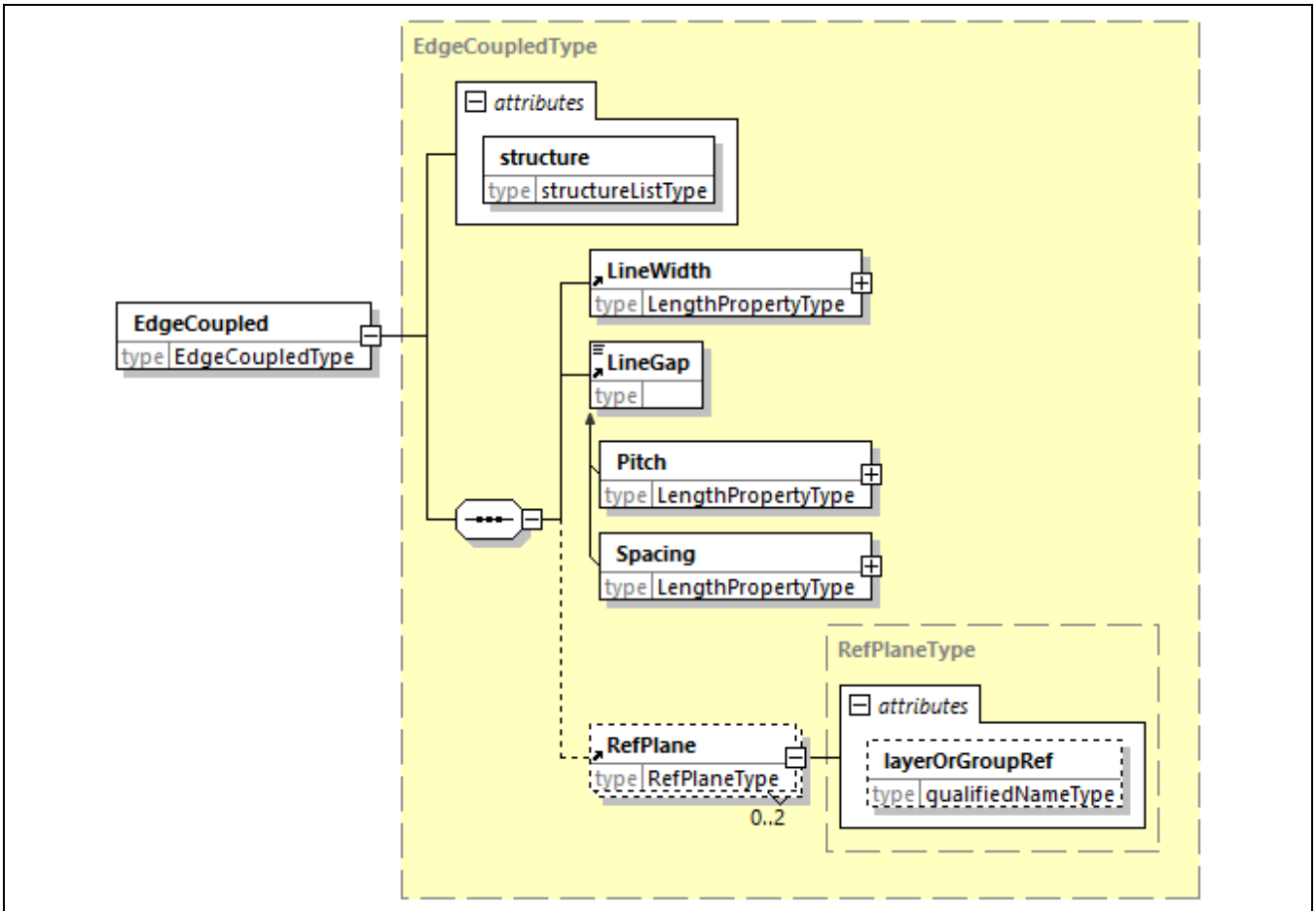
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
SIngleEnded	SingleEndedType	The identification of a an single ended transmission type of impedance specification	0-1
structure	structureListType	A set of enumerated string descriptions that identify the impedance structure. They consist of STRIPLINE_SYMMETRIC | STRIPLINE_ASYMMETRIC |STRIPLINE _PLANE_LESS | MICROSTRIP_EMBEDDED | MICROSTRIP_NO_MASK | MICROSTRIP_MASK_COVERED | MICROSTRIP_DUAL_MASK_COVERED |	1-1
LineWidth	LengthPropertyType	A child element that defines the trace width, unit of measure, and tolerance of the coupled pairs	1-1
RefPlane	RefPlaneType	A child element that defines 0 to 2 reference planes	0-1

Example:

```
<Impedance value=”50” plusTol=”10” minusTol=”10”  tolPercent=“true” >
    <SingleEnded structure=”MICROSTRIP_MASK_COVERED”>
        <LineWidth value=”0.3” unit=”MM”/>
        <RefPlane layerOrGroupRef=”L2”/>
    </SingleEnded>
</Impedance>
```

8.1.1.7.2 EdgeCoupled

A permitted substitution of `TransmissionType` as a child of the `Impedance` element, to identify an edge coupled transmission type of impedance specification.

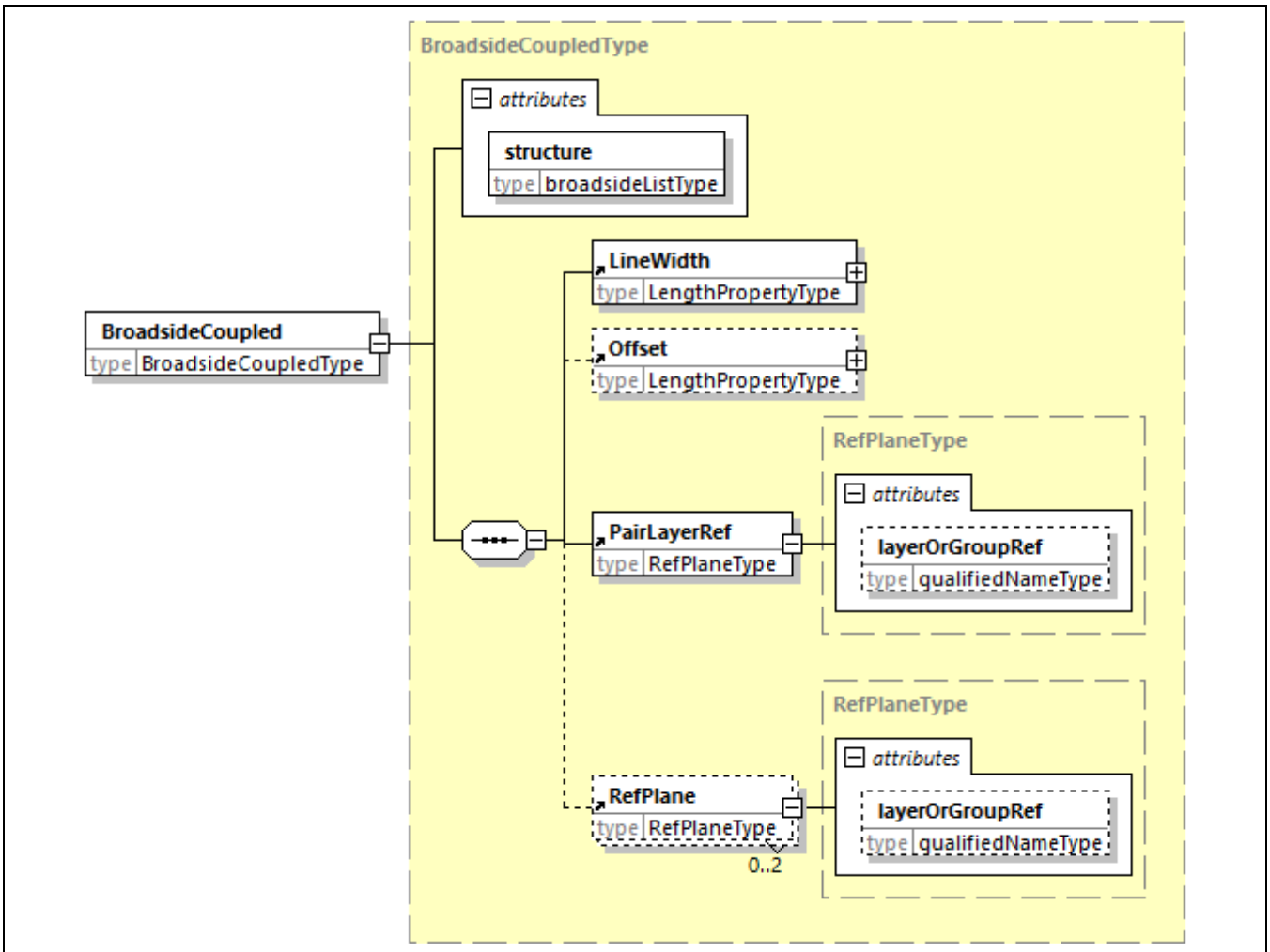


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
EdgeCoupled	EdgeCoupledType	The identification of a an Edge Coupled transmission type of impedance specification	0-1
structure	structureListType	A set of enumerated string descriptions that identify the impedance structure options. They consist of STRIPLINE_SYMMETRIC | STRIPLINE_ASYMMETRIC |STRIPLINE _PLANE_LESS | MICROSTRIP_EMBEDDED | MICROSTRIP_NO_MASK | MICROSTRIP_MASK_COVERED | MICROSTRIP_DUAL_MASK_COVERED	1-1
LineWidth	LengthPropertyType	A child element that defines the trace width, unit of measure, and tolerance of the coupled pairs	1-1
LineGap	ABSTRACT	A substitution group that permits the substitution of one of the following elements: Spacing (edge to edge distance), Pitch (center to center distance).	1-1
RefPlane	RefPlaneType	A child element that defines 0 to 2 reference planes	0-1

```
<Impedance value=”100” plusTol=”10” minusTol=”10”  tolPercent=“true” >
    <EdgeCoupled structure=”STRIPLINE_SYMMETRIC”>
        <LineWidth value=”300” unit=”MICRON” plusTol=”10” minusTol=”10”  tolPercent=“true” />
        <Pitch value=”200” unit=”MICRON” plusTol=”10” minusTol=”10”  tolPercent=“true” />
        <RefPlane layerOrGroupRef=”L8”/>
        <RefPlane layerOrGroupRef =”L10”/>
    </EdgeCoupled>
</Impedance>
```

8.1.1.7.3 BroadsideCoupled

A permitted substitution of `TransmissionType` as a child of the `Impedance` element, to identify a broadside coupled transmission type of impedance specification.

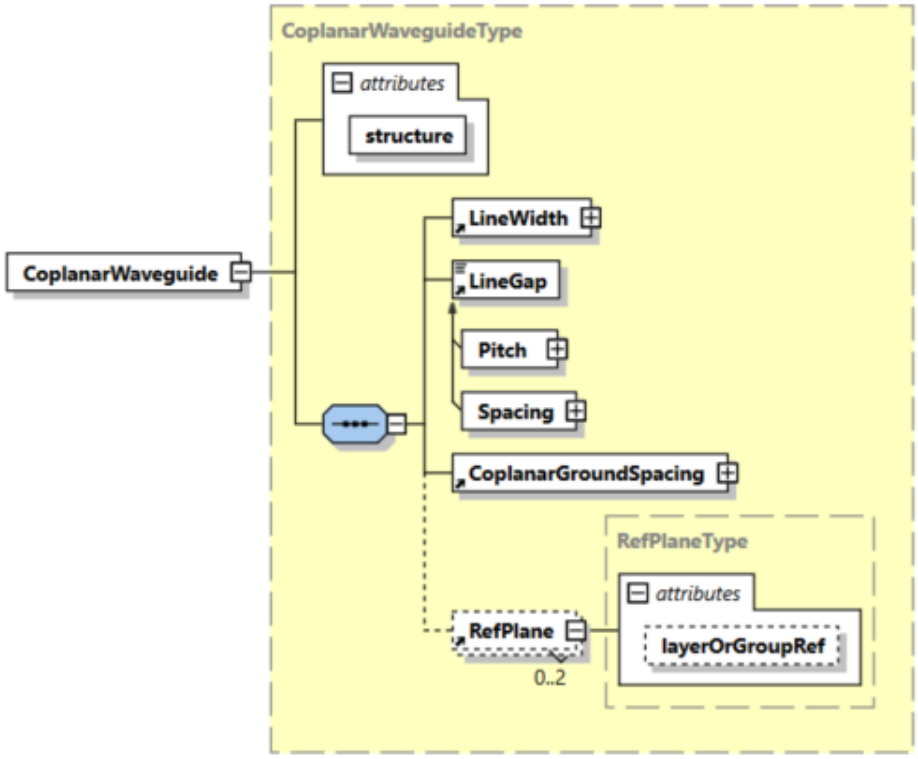


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
BroadsideCoupled	BroadsideCoupledType	The identification of a an Edge Coupled transmission type of impedance specification	0-1
structure	broadsideListType	A set of enumerated string descriptions that identify the broadside coupled specific impedance structure options. They consist of STRIPLINE_SYMMETRIC | STRIPLINE_ASYMMETRIC | STRIPLINE _PLANE_LESS	1-1
LineWidth	LengthPropertyType	The trace width, unit of measure, and tolerance of the coupled pair	1-1
Offset	LengthPropertyType	The horizontal distance between corresponding edges of the coupled pair, indicating an offset in vertical alignment. A zero value with tolerance can be used to specify a tolerance for the offset.	0-1
PairLayerRef	RefPlaneType	Reference to the layer occupied by the 2nd net in the broadside coupled pair, the layer referencing this `Spec` being the 1st net in the pair.	1-1
RefPlane	RefPlaneType	A child element that defines that there must be 0-2 reference planes	0-2

```
<Impedance value=”75” plusTol=”5” minusTol=”5”  tolPercent=“false” >
      <BroadsideCoupled structure=”STRIPLINE_ASYMMETRIC”>
  <LineWidth value=”25” unit=”MIL” plusTol=”10” minusTol=”10”  tolPercent=“true” />
  <Spacing value=”15” unit=”MIL” plusTol=”10” minusTol=”10”  tolPercent=“true” />
       <RefPlane layerOrGroupRef=”L5”/>
       <RefPlane layerOrGroupRef =”L7”/>
      </BroadsideCoupled>
</Impedance>
```

8.1.1.7.4 CoplanarWaveguide

A permitted substitution of `TransmissionType` as a child of the `Impedance` element, to identify a broadside coupled transmission type of impedance specification.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
CoplanarWaveguide	CoplanarWaveguide Type	The identification of a an Edge Coupled transmission type of impedance specification	0-1
structure	coplanarListType	A set of enumerated string descriptions that identify the broadside coupled specific impedance structure options. They consist of COPLANAR_WAVEGUIDE_STRIPLINE | COPLANAR_WAVEGUIDE_EMBEDDED | COPLANAR_WAVEGUIDE_NO_MASK | COPLANAR_WAVEGUIDE_MASK_COVERED | COPLANAR_WAVEGUIDE_DUAL_MASKED_COVERED | OTHER. If OTHER is noted a comment attribute is required.	1-1
LineWidth	LengthPropertyType	A child element that defines the trace width, unit of measure, and tolerance of the coupled pairs	1-1
LineGap	ABSTRACT	A substitution group that permits the substitution of one of the following elements: Spacing, Pitch	1-1
CoplanarGroundSpacing	LengthPropertyType	A child element that defines the distance from the outer edge of the coupled pair to the edge of the ground plane on the same layer. Unit of measure, and tolerance of the coupled pairs are also specified	1-1
RefPlane	RefPlaneType	A child element that defines that there can be an optional 0 to 2 reference planes	0-1

```
<Impedance value=”75” plusTol=”5” minusTol=”5”  tolPercent=“false” >
    <CoplanarWaveguide structure=”COPLANAR_WAVEGUIDE_EMBEDDED”>
        <LineWidth value=”25” unit=”MIL” plusTol=”10” minusTol=”10”  tolPercent=“true” />
        <Spacing value=”15” unit=”MIL” plusTol=”10” minusTol=”10”  tolPercent=“true” />
        <ColplanarGroundSpacing value=”5” unit=”MIL” plusTol=”10” minusTol=”10”  tolPercent=“true” />
    </ CoplanarWaveguide >
 </Impedance>
```

8.1.1.7.5 Transmission Type Common Child Elements

Attribute / Element Name	Attribute / Element Type	Description	Occurance
LineWidth, Spacing, Pitch, Offset, CoplanarGroundSpacing	lengthPropertyType	A type which describes a length in terms of value, unit of measure and tolerance	
value	double	The value of the length	1-1
unit	lengthUnitType	An enumerated list that defines the unit of length options. Possible options are: MM | MICRON | MIL | INCH	1-1
tolPlus	nonNegativeDouble	The plus tolerance that may be applied to the nominal number defined in the Value attribute setting the upper control limit.	0-1
tolMinus	nonNegativeDouble	The minus tolerance that may be applied to the nominal number defined in the Value attribute setting the lower control limit.	0-1
tolPercent	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lower case). True means that the attribute(s) `tolPlus` and/or `tolMinus` are a percentage. False indicates that they represent a nonNegativeDouble. The Default is “false”.	0-1

Attribute / Element Name	Attribute / Element Type	Description	Occurance
RefPlane, PairLayerRef	RefPlaneType	A type which describes a length im terms of values, unit of measure and tolerance	
layerOrGroupRef	qualifiedNameType	A unique `StackupLayer` or `StackupGroup` name that is used in the impedance definition	1-1

8.1.1.7.6 Impedance SpecRef Instantiation

There are 5 different parent element types to which a `SpecRef` child element can be added to reference an impedance specification, as shown below. They are listed in order of precedence, in the event that there is more than one impedance `SpecRef` covering the same net or segment of net.

8.1.1.7.6.1 LayerFeature:Set

An impedance `SpecRef` at this level defines the impedance of a given segment of a net on a given layer. The net is identified by the value of the `net` attribute of the `Set` element. In this case the transmission type specified in the `Spec` referenced by the `SpecRef` MUST match the transmission type of the net to which the `SpecRef` is attached. For example if the `Spec` is for transmission type `SingleEnded` then it should not be referenced by a `SpecRef` on a net of transmission type `EdgeCoupled`.
If all segments of a given net on a given layer are defined under the same `Set` element then a single `SpecRef` of that element can define the impedance for all segments of the net on that layer – see the example below. This is useful when defining the impedance of say an `EdgeCoupled` net on a given layer, when other `EdgeCoupled` nets on that layer have different impedance requirements.

```
<LayerFeature layerRef="TOP">
   <Set net="TX_P" netPair="TX_N">
     <SpecRef id="diff100"/>
     <Features>
       <Location x="0.0" y="0.0"/>
       <Polyline>
         <PolyBegin x="156.4760" y="-1.9535"/>
         <PolyStepSegment x="156.4760" y="-1.2189"/>
         <PolyStepSegment x="157.0000" y="-0.6949"/>
         <LineDesc lineEnd="ROUND" lineWidth="0.300000"/>
         </Polyline>
       </Features>
       <Features>
       <Location x="0.0" y="0.0"/>
       <Polyline>
         <PolyBegin x="167.5785" y="15.6010"/>
         <PolyStepSegment x="166.8439" y="15.6010"/>
         <PolyStepSegment x="166.3199" y="16.1250"/>
         <LineDesc lineEnd="ROUND" lineWidth="0.300000"/>
         </Polyline>
       </Features>
     </Set>
```

8.1.1.7.6.2 StackupLayer

An impedance `SpecRef` at this level defines the impedance of a given transmission type and a given line width across the whole of a given layer. For example if the `SpecRef` of a given layer references a `Spec` for transmission type `SingleEnded` and line width 0.3mm then it shall apply to all the net segments of transmission type `SingleEnded` and `lineWidth` of 0.3mm on that layer. There may be another SpecRef referencing a `SingleEnded` impedance assigned to the same layer, but applying to a different line width.

```
<StackupLayer layerOrGroupRef="TOP" thickness="0.0559" tolPlus="0.0056" tolMinus="0.0056">
    <SpecRef id="IMPEDANCE_SPEC_LAYER_TOP"/>
</StackupLayer>
```

8.1.1.7.6.3 LogicalNet

An impedance `SpecRef` at this level defines the impedance of all segments of a given net on all layers. The `TransmissionType` of the referenced `Spec` **shall** be omitted, so that only the impedance value and optional tolerance are specified. Then the `TransmissionType` and all child physical properties (linewidth, spacing, etc) are free to change across different layers, in order to match the specified impedance value.

```
<LogicalNet name="TX_P" netPair="TX_N">
   <PinRef componentRef="IC50" pin="22"/>
   <PinRef componentRef="R32" pin="1"/>
    <SpecRef id="100ohms"/>
</LogicalNet>
```

8.1.1.7.6.4 StackupGroup

An impedance `SpecRef` at this level defines the impedance of a given transmission type and a given line width across the whole of a given `StackupGroup`. For example if the `SpecRef` of a given layer references a `Spec` for transmission type `SingleEnded` and line width 0.3mm then it shall apply to all the net segments of transmission type `SingleEnded` and `lineWidth` of 0.3mm in that `StackupGroup`. There may be another SpecRef referencing a `SingleEnded` impedance assigned to the same group, but applying to a different line width.

```
<StackupGroup name="GROUP_PRIMARY" thickness="0.035" tolPlus="0" tolMinus="0">
   <StackupLayer layerOrGroupRef="L1" thickness="0.02" tolPlus="0" tolMinus="0"/>
   <StackupLayer layerOrGroupRef="L2" thickness="0.015" tolPlus="0" tolMinus="0"/>
   <SpecRef id="GROUP_PRIMARY_SPEC"/>
</StackupGroup>
```

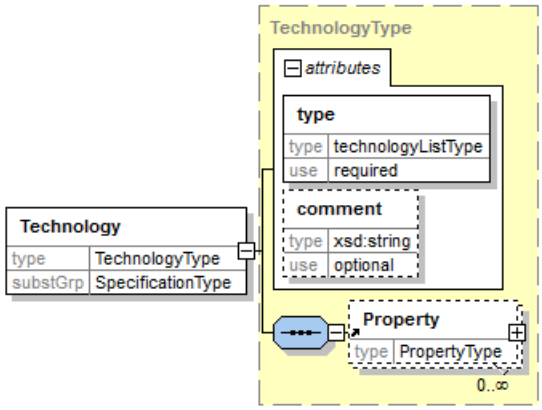
8.1.1.7.6.5 Stackup

An impedance `SpecRef` at this level defines the impedance of a given transmission type and a given line width across the whole of the `Stackup`. For example if the `SpecRef` of a given layer references a `Spec` for transmission type `SingleEnded` and line width 0.3mm then it shall apply to all the net segments of transmission type `SingleEnded` and `lineWidth` of 0.3mm in the `Stackup`. There may be another SpecRef referencing a `SingleEnded` impedance assigned to the same `Stackup`, but applying to a different line width.

```
<Stackup name="PRIMARY" overallThickness="2.23" tolPlus="0" tolMinus="0" whereMeasured="MASK">
            <SpecRef id="diff100"/>
            <SpecRef id="STANDARD_NOTES"/>
```

8.1.1.8 Technology

The Technology specification feature is intended to define the mounting platform technology so that the specifications applied relate to those materials and structures identified by the technology type in order to further influence the product defined in the IPC-2581 file.

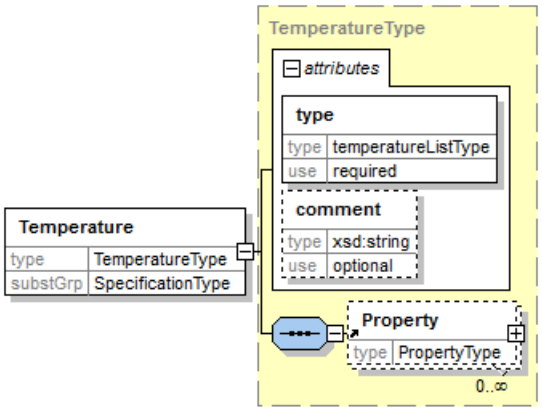


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Technology	TechnologyType	The identification of a SpecificationType related to the category of the product used for mounting components identified by the structure technology of the product identified in the IPC-2581 file.	0-n
type	technologyListType	A set of enumerated string descriptions that identify the mounting platform technology requirements. The description categories consist of RIGID | RIGID_FLEX | FLEX | HDI | EMBEDDED_COMPONENT | OTHER. If OTHER is noted a comment attribute is required.	1-1
comment	string	A description of any technology requirements that are a different category or supplement the technologyListType enumerations	0-1
Property	PropertyType	A nested element containing numerical attribute descriptions that relate to the technology requirements and permit the references to alternate conditions that may apply. Unit descriptions apply only to the `specificationType` and do not impact the requirements of the overall IPC-2581 file Unit descriptions	0-n

```
<Spec name=”board_technology”>
    <Technology type=”RIGID”>
    </Technology>
</Spec>
```

8.1.1.9 Temperature

The Temperature specification feature is intended to define the products' ability to withstand high temperatures in order to maintain its physical characteristics within the requirements identified in the IPC-2581 file.

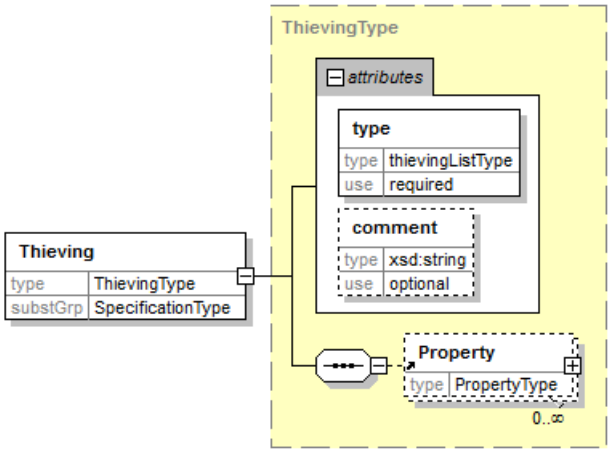


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Temperature	TemperatureType	The identification of a SpecificationType related to the temperature impact on the product identified in the IPC-2581 file and its' ability to maintain the physical properties needed for reliable performance.	0-n
type	temperatureListType	A set of enumerated string descriptions that identify the temperature extremes at which the material changes state. The description categories are THERMAL_DELAMINATION | EXPANSION_Z_AXIS | EXPANSION_X_Y AXIS | OTHER. If OTHER is noted, a comment attribute is required.	1-1
comment	string	A description of any temperature impact characteristic that is different or supplements the TemperatureListType enumerations	0-1
Property	PropertyType	A nested element containing numerical attribute descriptions that relate to the temperature impact characteristics and permit the references to alternate conditions that may apply. Unit descriptions apply only to the specificationType and do not impact the requirements of the overall IPC-2581 file Unit descriptions	0-n

```
<Spec name=”minimum_Td”>
    <Temperature type=”THERMAL_DELAMINATION”>
            <Property value=”180” unit=”CELCIUS”/>
    </Temperature>
</Spec>
```

8.1.1.10 Thieving

The Thieving specification feature is intended to define the areas where thieving is allowed (keep in) or not allowed (keep out) during fabrication of the PCB. This feature can also be used by a fabrication supplier to identify the areas where thieving will be added (keep in).



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Thieving	ThievingType	The identification of a SpecificationType related to the keep in and keep out areas in the copper layers where thieving is to be applied or not allowed.	0-n
type	thievingListType	A set of enumerated string descriptions that identify either keep in or keep out. The description categories are KEEP_IN | KEEP_OUT.	1-1
comment	string	A description of any thieving characteristic that is different or supplements the thievingListType enumerations	0-1
Property	PropertyType	A nested element containing numerical or text attribute descriptions that relate to the thieving characteristics and permit the references to alternate conditions that may apply. Unit descriptions apply only to the specificationType and do not impact the requirements of the overall IPC-2581 file Unit descriptions	0-n

```
<Spec name=”Ethernet_POE_Connector_Thieving”>
    <Thieving type=”KEEP_OUT”>
            <Property layerOrGroupRef =”Layer1”/>
            <Property layerOrGroupRef =”Layer2”/>
            <Property layerOrGroupRef =”Layer3”/>
    </Thieving>
</Spec>
```

The thieving keep in/out areas shall be identified by a geometric feature, such as an outline, on a dedicated layer with its `layerFunction` attribute set to THIEVING_KEEP_INOUT (see section 8.2.1 Layer). The feature can be linked to its associated `Spec` element in one of two ways:

1. A `SpecRef` element, referencing the Thieving `Spec` element, is added as a child of the `Layer` element, such that a single `Spec` element applies to all keep in/out areas on the layer. Then each `LayerFeature` on the layer inherits the `SpecRef` of its parent layer:

```
<Layer name="Thieving_Constraints" layerFunction="THIEVING_KEEP_INOUT" side="NONE" polarity="POSITIVE">
   <SpecRef id=”ThievingKeepOut”/>
 <Layer/>
 :
   <LayerFeature layerRef=" Thieving_Constraints">
            <Set polarity="POSITIVE">
              <Color r="0" g="0” b="255"/>
              <Features>
                 <Outline>
```

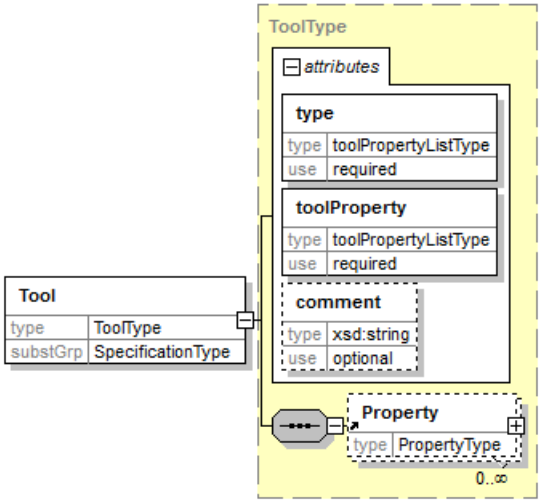
```
                    <Polygon>
                        <PolyBegin x = "10.4" y = "0.0"/>
                        <PolyStepSegment x = "12.4" y = "0.0"/>
                        <PolyStepSegment x = "12.4" y = "0.055"/>
                        <PolyStepSegment x = "10.4" y = "0.055"/>
                        <PolyStepSegment x = "10.4" y = "0.0"/>
                    </Polygon>
                    <LineDesc lineEnd = "ROUND" lineWidth = "0.2"/>
                </Outline>
            </Features>
        </Set>
    </LayerFeature>
```

2. A `SpecRef` element, referencing the Thieving `Spec` element is added as a child of the `LayerFeature:Set` element, such that each keep in/out area on the thieving layer can reference a different `Spec` element. In this case a `SpecRef` element is NOT added as a child of the parent `Layer` element.

```
 <Layer name="Thieving_Constraints" layerFunction="THIEVING_KEEP_INOUT" side="NONE" polarity="POSITIVE"/>
:
    <LayerFeature layerRef="Thieving_Constraints">
        <Set polarity="POSITIVE">
            <Color r="255" g="0" b="0"/>
            <SpecRef Id="ThievingKeepIn"/>
            <Features>
                <Outline>
                    <Polygon>
                        <PolyBegin x = "10.4" y = "0.0"/>
                        <PolyStepSegment x = "12.4" y = "0.0"/>
                        <PolyStepSegment x = "12.4" y = "0.055"/>
                        <PolyStepSegment x = "10.4" y = "0.055"/>
                        <PolyStepSegment x = "10.4" y = "0.0"/>
                    </Polygon>
                    <LineDesc lineEnd = "ROUND" lineWidth = "0.2"/>
                </Outline>
            </Features>
        </Set>
    </LayerFeature>
```

8.1.1.11 Tool

The Tool specification feature is intended to define the type of tool(s) needed in the fabrication of the board or the assembly and includes those necessary to prepare the products for separation from a panel or pallet as well as providing holes or cutouts needed within the product structure.

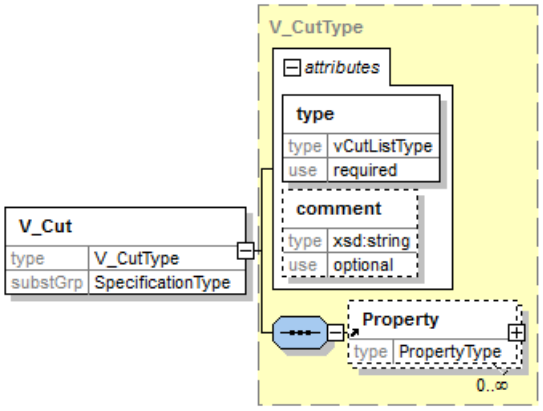


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Tool	ToolType	The identification of a SpecificationType related to the tools needed to manufacture the product identified in the IPC-2581 file and includes both hard tools and photo (Laser) ablation requirement.	0-n
type	toolListType	A set of enumerated string descriptions that identify the tool types required in the manufacturing of the boards or the assemblies. The tool types include CARBIDE | ROUTER | LASER | FLATNOSE | EXTENSION | V_CUTTER | SCREWDRIVER | WRENCH | OTHER. If OTHER is noted a comment attribute is required.	1-1
toolProperty	toolPropertyListType	A set of enumerated string descriptions that identify the tool properties required in the manufacturing of the boards or the assemblies. The tool properties include DRILL_SIZE | FINISHED_SIZE | BIT_ANGLE| TORQUE | HEX_NUT_SIZE | PHILLIPS_HEAD | FLAT_HEAD | TORX_HEAD | ALLEN_HEAD | OTHER. If OTHER is noted a comment attribute is required.	1-1
comment	string	A description of any tool characteristic that is different or supplements the toolListType or toolPropertyListType enumerations	0-1
Property	PropertyType	A nested element containing numerical attribute descriptions that relate to the tool descriptions and permits the references to alternate conditions that may apply. Unit descriptions apply only to the specificationType and do not impact the requirements of the overall IPC-2581 file Unit descriptions	0-n

```
<Spec name="drill_1">
  <Tool type="CARBIDE" toolProperty="DRILL_SIZE">
    <Property text="B" unit="GAUGE"/>
  </Tool>
  <Tool type="CARBIDE" toolProperty="FINISHED_SIZE">
    <Property value="0.5" unit="MM" plusTol ="0.05" minusTol="0.05"/>
  </Tool>
</Spec>

<Spec name=" v_cutter_tool">
  <Tool type="V_CUTTER" toolProperty="BIT_ANGLE">
    <Property value="90" unit="DEGREES"/>
  </Tool>
</Spec>

<Spec name=" screw_torque">
      <Tool type="SCREWDRIVER" toolProperty="FLAT_HEAD"/>
       <Tool type="SCREWDRIVER" toolProperty="TORQUE">
             <Property value="7.5" unit="IN-LB"/>
       </Tool>
</Spec>

<Spec name="nut_torque">
      <Tool type="WRENCH" toolProperty="HEX_NUT_SIZE">
            <Property text="M2"/>
       </Tool>
       <Tool type="WRENCH" toolProperty="TORQUE">
            <Property value="6.5" unit="IN-LB" refUnit="MIN"/>
             <Property value="7.0" unit="IN-LB" refUnit="MAX"/>
        </Tool>
</Spec>
```

8.1.1.12 V_Cut

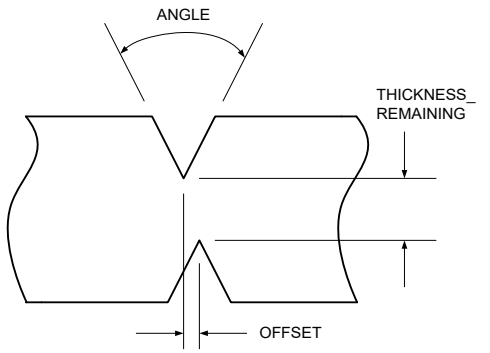
The V_Cut specification feature is intended to define the limits and detail descriptions of the scoring requirements when using a diamond wheel cutter intended to remove material from a board or panel in order to reduce the total thickness. This always implies cuts on both the top and bottom of the board. The specification type OFFSET can be used to define the vertical distance and/or tolerance between top and bottom cuts.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
V_Cut	V_CutType	The identification of a SpecificationType related to the scoring or groove cutting into a board, panel or pallet where a certain amount of material remains so that the product can be further processed before final separation.	0-n

type	vCutListType	A set of enumerated string descriptions that identify the characteristics of the groove or V cut as related to the thickness of the material. The description categories are ANGLE | THICKNESS_REMAINING | OFFSET | OTHER. If OTHER is noted a comment attribute is required.	1-1
comment	string	A description of any V cut characteristic that is different or supplements the vCutListType enumerations	0-1
Property	PropertyType	A nested element containing numerical attribute descriptions that relate to the temperature impact characteristics and permit the references to alternate conditions that may apply. Unit descriptions apply only to the specificationType and do not impact the requirements of the overall IPC-2581 file Unit descriptions	0-n

```
<Spec name="VCut_1">
    <V_Cut type="ANGLE">
            <Property value="90" unit="DEGREES" plusTol="5" minusTol="5"
tolPercent="true"/>
    </V_Cut>
    <V_Cut type="THICKNESS_REMAINING">
            <Property value="0.5" unit="MM" plusTol="0.1" minusTol="0.1"/>
    </V_Cut>
    <V_Cut type="OFFSET">
            <Property value="0.0" unit="MM" plusTol="0.1" minusTol="0.1"/>
    </V_Cut>
</Spec>
```



The V cut shall be identified by a geometric feature, that is a straight line, on a layer dedicated to V cuts, with its `layerFunction` attribute set to V_CUT (see section 8.2.1 Layer). The line denotes the path of the V cut tool. The co-ordinates of the start and end points of the line shall be relative to the datum of the board. The line width is irrelevant, since it is the line center path and attributes of the V-cut `Spec` element that define the V cut. The line on the V cut layer can be linked to its associated `Spec` element in one of two ways:

- A `SpecRef` element, referencing the V cut `Spec` element, is added as a child of the `Layer` element, such that a single `Spec` element applies to all V cuts on the layer. Then each `LayerFeature` on the layer inherits the `SpecRef` of its parent layer:

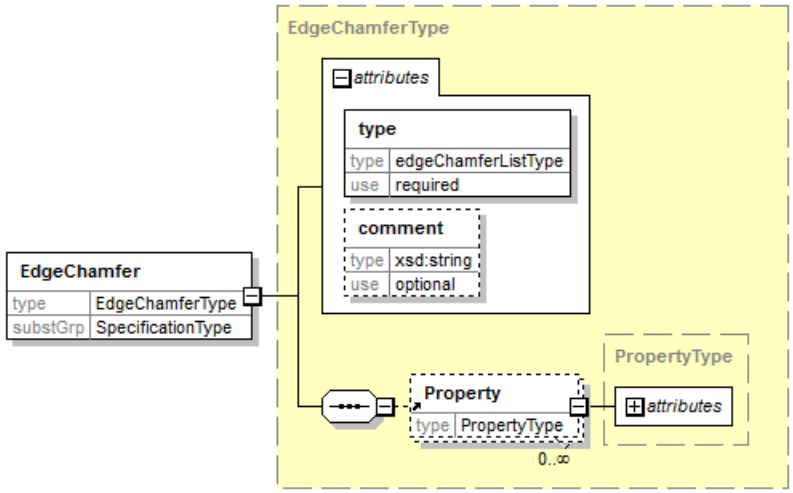
```
<Layer name="V-Score" layerFunction="V_CUT" side="NONE" polarity="POSITIVE">
   <SpecRef id="VCut_all"/>
 <Layer/>

      <LayerFeature layerRef="V-Score">
            <Set polarity="POSITIVE">
              <Color r="0" g="255" b="0"/>
              <Features>
                <Location x="0" y="0"/>
                <Line startX="-7" startY="-7" endX="-7" endY="223">
                  <LineDesc lineWidth="0.025" lineEnd="ROUND"/>
                </Line>
              </Features>
            </Set>
      </LayerFeature>
```

- A `SpecRef` element, referencing the V cut `Spec` element is added as a child of the `LayerFeature:Set` element, such that each V cut on the V cut layer can reference a different `Spec` element. In this case a `SpecRef` element is NOT added as a child of the parent `Layer` element:

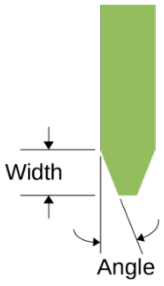
```
<Layer name="V-Score" layerFunction="V_CUT" side="NONE" polarity="POSITIVE"/>

      <LayerFeature layerRef="V-Score">
            <Set polarity="POSITIVE">
              <Color r="0" g="255" b="0"/>
              <SpecRef Id="VCut_1"/>
              <Features>
                <Location x="0" y="0"/>
                <Line startX="-7" startY="-7" endX="-7" endY="223">
                  <LineDesc lineWidth="0.025" lineEnd="ROUND"/>
                </Line>
              </Features>
            </Set>
      </LayerFeature>
```

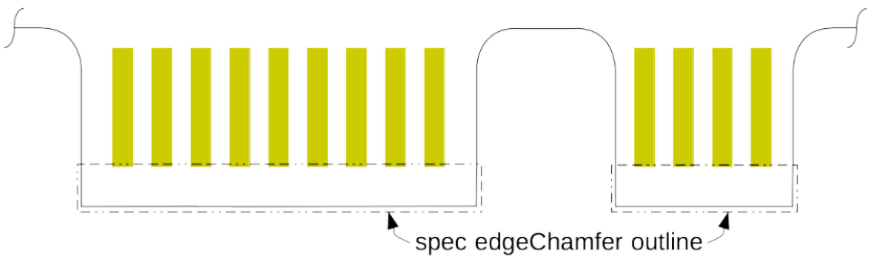
8.1.1.13 EdgeChamfer

The EdgeChamfer specification feature is intended to define the board edges where chamfered edges are needed during fabrication of the PCB.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
EdgeChamfer	EdgeChamfer Type	The identification of a SpecificationType related to the board edges where a chamfer is required across the depth (cross section) of the board.	0-n
type	edgeChamfer ListType	A set of enumerated string descriptions that identify characteristics of the edge chamfer. The description categories are ANGLE | WIDTH | SIDE. ANGLE shall be referenced from the plane of each layer identified by SIDE such that the edge of the board is the thinnest part of the chamfer. WIDTH shall define the perpendicular distance from the identified board edge to the innermost extent of the chamfer SIDE shall identify one or both outer layers to receive the chamfer	1-1
comment	string	A description of any edge chamfer characteristic that is different from, or supplements the edgeChamferListType enumerations	0-1
Property	PropertyType	A nested element containing numerical or text attribute descriptions that relate to the chamfer characteristics and permit the references to alternate conditions that may apply. Unit descriptions apply only to the specificationType and do not impact the requirements of the overall IPC-2581 file Unit descriptions	0-n





```
<Spec name="Edge_Connector_P1">
    <EdgeChamfer type="ANGLE">
        <Property value="20.0" unit="DEGREES" plusTol="5.0" minusTol="5.0" tolPercent="false"/>
    <EdgeChamfer type=" WIDTH">
        <Property value="0.055" unit="INCH" plusTol="0.010" minusTol="0.010" tolPercent="false"/>
    <EdgeChamfer type="SIDE">
        <Property layerOrGroupRef="L1"/>
        <Property layerOrGroupRef="BOTTOM"/>
    </EdgeChamfer>
</Spec>
```

The edge chamfer shall be identified by a geometric feature, such as an oultine or a line, on a layer dedicated to edge chamfers, with its `layerFunction` attribute set to EDGE_CHAMFER (see section 8.2.1 Layer). The feature denotes the location of the edge chamfer, but does not dictate the tool or method used to create it. The geometric feature shall be linked to the specification by a `SpecRef` element in one of two ways:

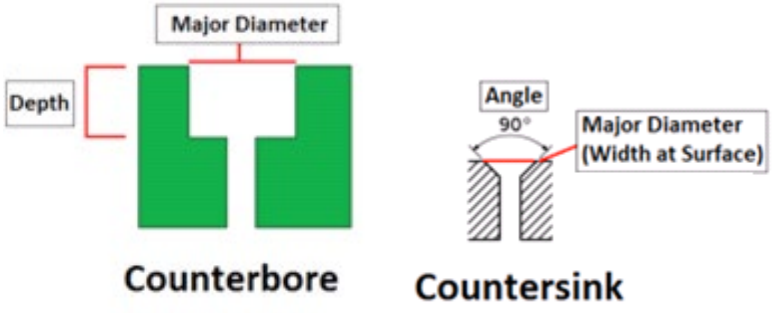
1. The `SpecRef` is assigned to the layer with the EDGE_CHAMFER `layerFunction`, such that all geometric features on that layer reference the same specification.

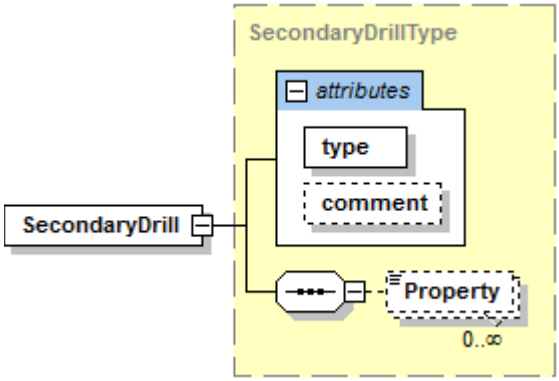
2. The `SpecRef` is assigned to `LayerFeature:Set` which defines an individual geometric feature, so that different features on the same layer can reference different specifiactions.

See Section 8.1.1.12 V_Cut for a more detailed example.

8.1.1.14 SecondaryDrill

The `SecondaryDrill` specification feature is intended to define the dimensions and characteristics of countersink or counter bore of a hole. Countersink and counterbore are secondary drilling operations, not routing operations. The secondary drill operation is purely a mechanical function and not related to electrical functions such as back drill which, has electrical characteristics.
For a counterbore hole, the MAJOR_DIAMETER value is the finished hole size of the secondary drill operation. A DEPTH value is also required to define the finished depth of the counter bore. The MAJOR_DIAMETER for a countersink is the maximum finished diameter at the top of the countersink. An ANGLE definition is required for the countersink hole. The combination of the MAJOR_DIAMETER and ANGLE will determine the depth of the countersink.





Attribute / Element Name	Attribute / Element Type	Description	Occurrence
SecondaryDrill	SecondaryDrillType	The identification of a SpecificationType related to the countersink or counterbore of the hole.	0-n
Type	SecondaryDrillListType	A set of enumerated string descriptions that identify the condition of the Secondary drill requirement. Values consist of MAJOR_DIAMETER | DEPTH | SIDE | ANGLE | OTHER. If OTHER is specified a `Comment` attribute is required.	1-1
Comment	String	A description of the requirements for Counterdrill that are different or supplement the SecondaryDrillListType enumerations	0-1

```
<Spec name=”Countersink_1”>
    <SecondaryDrill type=”MAJOR_DIAMETER”>
        <Property value=”0.5” unit=”MM” plusTol=”0” minusTol=”5” tolPercent=”TRUE”/>
    </ SecondaryDrill >
    < SecondaryDrill type=”ANGLE”>
        <Property value=”90” unit=”DEGREES”>
    </ SecondaryDrill >
    < SecondaryDrill type=”SIDE”>
        <Property text=”TOP”>
    </ SecondaryDrill >
</Spec>

<Spec name=” Counterbore _1”>
    < SecondaryDrill type=”MAJOR_DIAMETER”>
        <Property value=”0.5” unit=”MM” plusTol=”0” minusTol=”5” tolPercent=”TRUE”/>
    </ SecondaryDrill >
    < SecondaryDrill type=”DEPTH”>
        <Property value=”0.3” unit=”MM”>
    </ SecondaryDrill >
    < SecondaryDrill type=”SIDE”>
        <Property text=”BOTTOM”>
    </ SecondaryDrill >
</Spec>
```

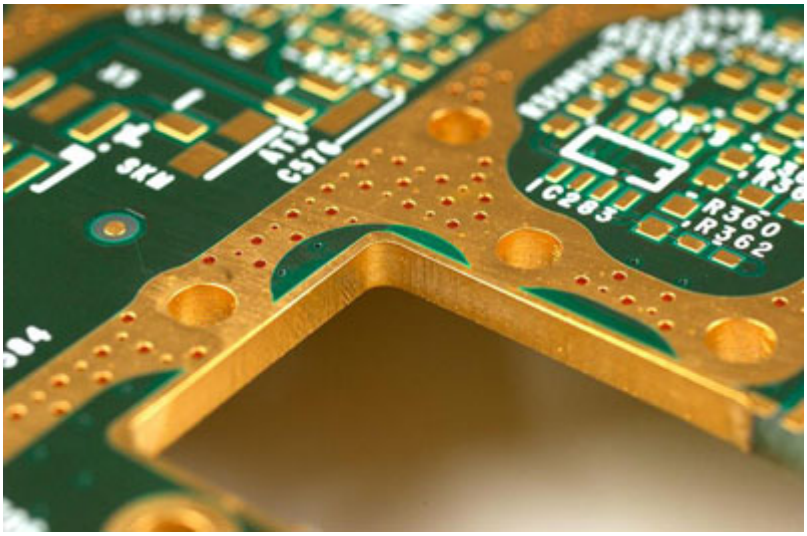
8.1.1.15 EdgePlating

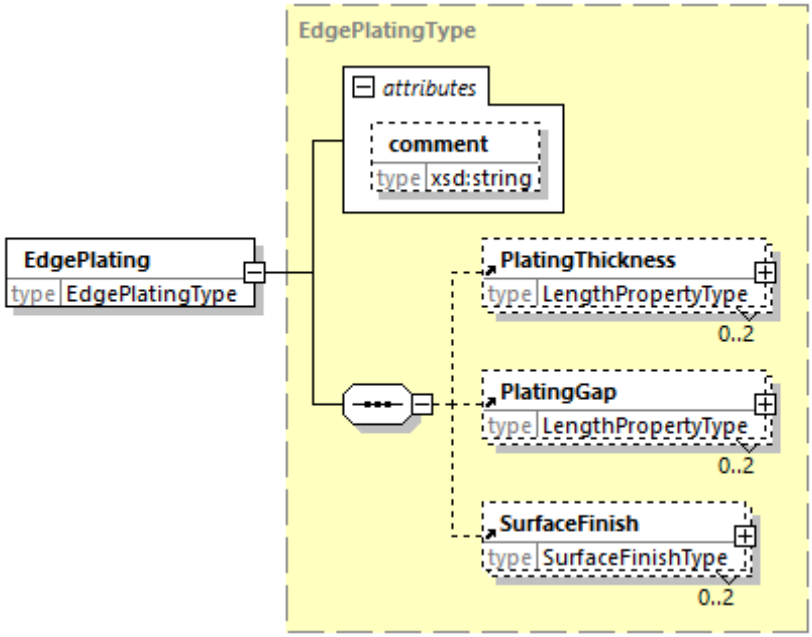
The EdgePlating specification feature is intended to define areas where copper plating is required on the edge (cross section) of the board, during fabrication of the PCB. An example is shown below. The edge plating shall be identified by a geometric feature, such as an oultine or a line, on a `Layer` dedicated to edge plating, with its `layerFunction` attribute set to EDGE_PLATING (see section 8.2.1 Layer). The geometric feature shall be linked to the specification by a `SpecRef` element in one of two ways:

1. The `SpecRef` is assigned to the layer with the EDGE_PLATING `layerFunction`, such that all geometric features on that layer reference the same specification.

2. The `SpecRef` is assigned to `LayerFeature:Set` which defines an individual geometric feature, so that different features on the same layer can reference different specifiactions.

Refer to Section 8.1.1.12 V_Cut for a more detailed example.





Attribute / Element Name	Attribute / Element Type	Description	Occurrence
EdgePlating	EdgePlatingType	The identification of a SpecificationType related to the board edges where copper plating is required across the depth (cross section) of the board.	0-n
comment	String	Any comment(s) relating to the edge plating	0-n
PlatingThickness	LengthPropertyType	The required thickness of the copper plating. Two instances are allowed so that a `constraintType` of both MIN & MAX can be specified. In the absence of any `constraintType` the value is nominal	0-2
PlatingGap	LengthPropertyType	The allowed gap in the edge plating, if gaps are required for holding the board, or other purposes. Two instances are allowed so that a `constraintType` of both MIN & MAX can be specified. In the absence of any `constraintType` the value is nominal.	0-2
SurfaceFinish	SurfaceFinishType	The surface finish required on the edge plated copper, defined as one of an enumerated list of finishes per IPC-6012. If a second finish is required then they shall be applied in the order that they appear. For example: <SurfaceFinish type="N"/> <SurfaceFinish type="G"/> means nickel is applied first, followed by gold on top of that.	0-2

```
<Spec name="Edge_Plating_P1">
    <EdgePlating>
        <PlatingThickness value="4.0" unit="MICRON" plusTol="0.5" minusTol="0.5" tolPercent="false" constraintType="MIN"/>
        <PlatingThickness value="6.0" unit="MICRON" plusTol="0.5" minusTol="0.5" tolPercent="false" constraintType="MAX"/>
        <PlatingGap value="2.0" unit="MIL" constraintType="MAX"/>
        <SurfaceFinish type="G"/>
        <SurfaceFinish type="N"/>
    </EdgePlating>
</Spec>
```

8.1.1.16 SurfaceFinish

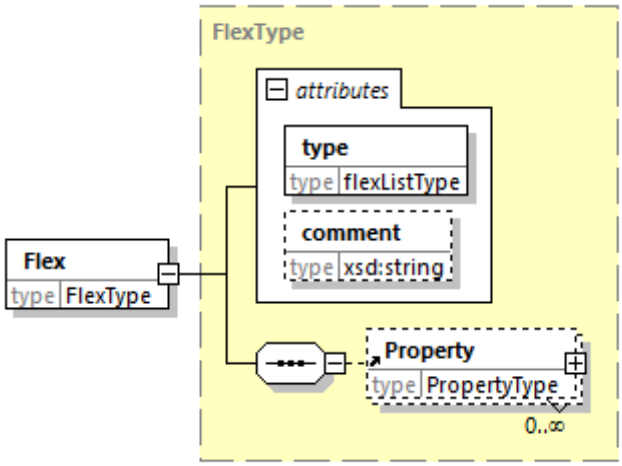
The SurfaceFinish specification type defines the material used for the surface finish(es) of a conductive material. It can be referenced by the `SpecRef` child of a `StackupLayer` that references a `Layer` with `layerFunction` COATINGCOND or COATINGNONCOND. Or it can be used as a child of `EdgePlating`, or referenced by a `SpecRef` of a conductive Model (e.g. a coin) that requires a surface finish.

Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Finish	FinishType	Indicates a surface finish material	0-1
type	surfaceFinishType	An enumerated string with values in accordance with IPC-6012. Refer to Table 3-3 “Final Finish and Coating Requirements” in IPC-6012 for a full description of the values. Possible values are: S | b1 | T | X | TLU | G | GS | GWB-1-G | GWB-1-N | GWB-2-G | GWB-2-N | N | NB | OSP | HT_OSP | ENIG-N | ENIG-G | ENEPIG-N | ENEPIG-G | ENEPIG-P | DIG | IAg | ISn | C | OTHER	1-1
comment	xsd:string	A comment field to describe a surfaceFinishType of OTHER, or to provide any other information about the finish	comment
Product	ProductType	A nested element to provide one or more specific product brand names of surface finish	0-n
name	string	The brand name of a specific surface finish	1-1
criteria	productCriteriaType	An enumerated list with values ALLOWED | SUGGESTED | PREFERRED | REQUIRED | CHOSEN	0-1

1. <Finish type =”HT_OSP”>
   <Product name=”Entek HT” criteria=”ALLOWED”/>
   <Product name= “Gilcoat SMD F2(LX)” criteria=”ALLOWED”/>
   </Finish>
2. <Finish type =”ENIG-G”/>
3. <Finish type =”OTHER” comment=”no finish rrequired”/>

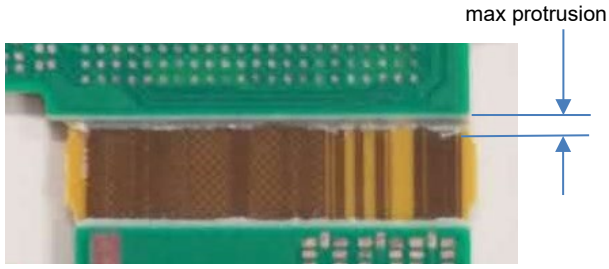
8.1.1.17 Flex

The `Flex` specification type defines the properties associated with a flex StackupZone. The `SpecRef` child element of a `StackupZone` flex element shall reference the Flex specification.

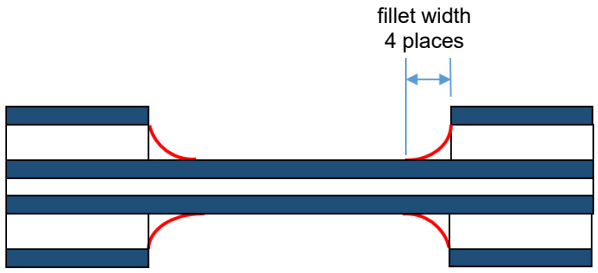


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Flex	FlexType	The identification of a SpecificationType related to the properties asscociated with a Flex stackup zone	0-n
type	flexListType	A set of enumerated string descriptions that identify the Flex associated property The description categories are ADHESIVE_SQUEEZE_OUT | DIELECTRIC_SQUEEZE_OUT | STRESS_RELIEF_FILLET_WIDTH | OTHER.	1-1
comment	string	A description of any thieving characteristic that is different or supplements the flexListType enumerations	0-1
Property	PropertyType	A nested element containing numerical or text attribute descriptions that relate to the thieving characteristics and permit the references to alternate conditions that may apply. Unit descriptions apply only to the specificationType and do not impact the requirements of the overall IPC-2581 file Unit descriptions	0-n

```
<Spec name="SqueezeOut">
    <Flex type="DIELECTRIC_SQUEEZE_OUT">
        <Property value="1.2" unit="MM" refUnit="MAX"/>
    <Flex>
</Spec>
```

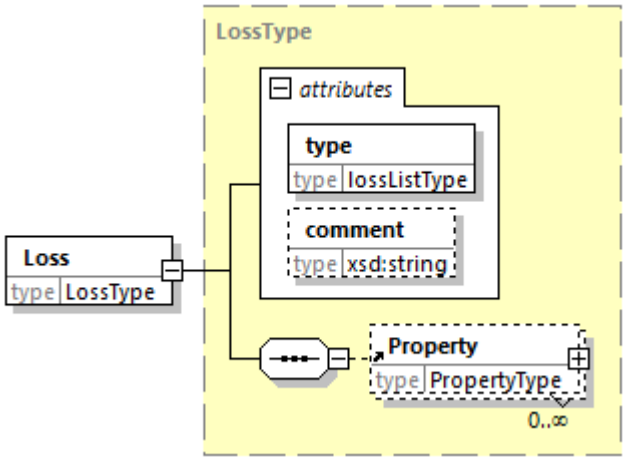



```
<Spec name="EpoxyAdhesiveStressRelief">
    <Flex type="STRESS_RELIEF_FILLET_WIDTH">
        <Property value="1" unit="MM" refUnit="MIN"/>
        <Property value="2" unit="MM" refUnit="MAX"/>
    <Flex>
</Spec>
```



8.1.1.18 Loss

The `Loss` specification type defines the loss requirements of the object that the associated `SpecRef` element is attached to. An appropriate object shall be a `LayerFeature Set` (signal trace, or shape, or group thereof), `LogicalNet, Stackup`, or `StackupGroup`.

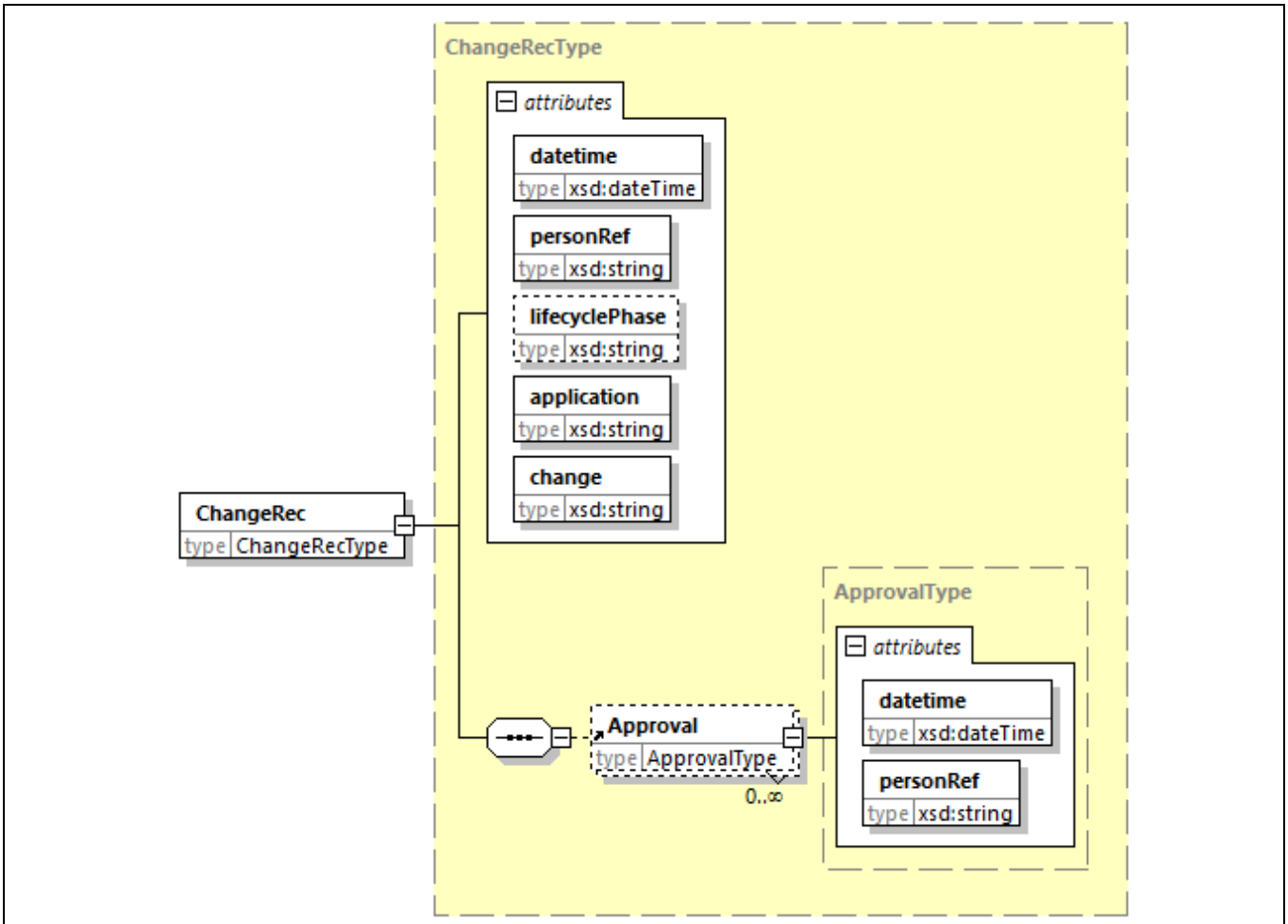


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Loss	LossType	The identification of a SpecificationType related to a type of electrical loss requirement	0-n
type	lossListType	An enumerated list to define the type of loss, with possible values ATTENUATION | IMPEDANCE | INSERTION | POWER | SIGNAL | VOLTAGE	1-1
comment	String	A comment describing a type of OTHER and/or any other comment related to the loss	0-1
Property	PropertyType	A nested element containing numerical attribute descriptions that relate to the loss requirements and permit the references to alternate conditions that may apply. Unit descriptions apply only to the specificationType and do not impact the requirements of the overall IPC-2581 file Unit descriptions	0-n

```
<Spec name=”attenuation_vs_frequency”>
        <Loss type =”ATTENUATION” >
                <Property value=”3” unit=”dB/INCH” refValue=”10e9” refUnit=”Hz”/>
                <Property value=”7” unit=”dB/INCH” refValue=”15e9” refUnit=”Hz”/>
        </Loss>
</Spec>

<Spec name=”max_insertion_loss”
        <Loss type =”INSERTION” >
                <Property value=”3” unit=”dB” refUnit=”MAX”/>
        </Loss>
</Spec>
```

8.1.2 ChangeRec

The `ChangeRec` element contains the `ChangeRec` elements specifying deviations requested by the manufacturer and approved by the customer (OEM, EMS, other).



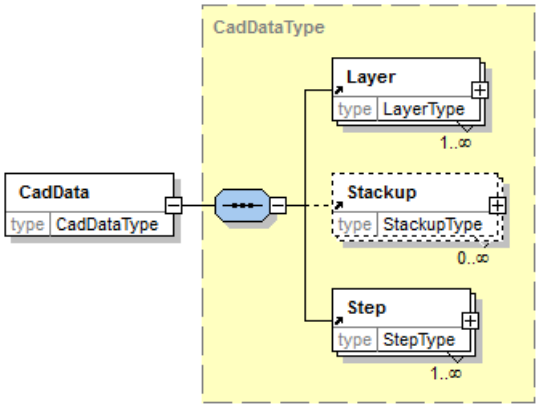
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
ChangeRec	ChangeRecType	A nested element containing identification and information about the description of the data in the Ecad section of the file.	0-n
datetime	dateTime	The standard date and time indication of the change request.	1-1
personRef	string	The name of the person to whom the request was made.	1-1
lifecyclePhase	string	A string indicating the lifecycle phase of the data, i.e., DFX review, stackup review, prototype, preproduction, production, etc.	0-1
application	string	The effectivity of the change and exactly where the change was to be made.	1-1
change	string	A detailed description of the change, including a reference to a URL if graphic descriptions are involved.	1-1
Approval	ApprovalType	A nested element that signifies who approved the suggested change submitted by the design, fabrication, assembly or test operation.	0-n
datetime	dateTime	The standard date and time indication of the change approval.	1-1
personRef	string	The name of the person who approved the change request.	1-1

8.2 CadData

The `CadData` element is the three-dimensional structure of the design that is retrieved from the Cad system as a group of layers. The information is contained in the `Layer` elements. The layers are listed in the correct order inside `CadData` and are grouped by `name`, and `layerFunction`.

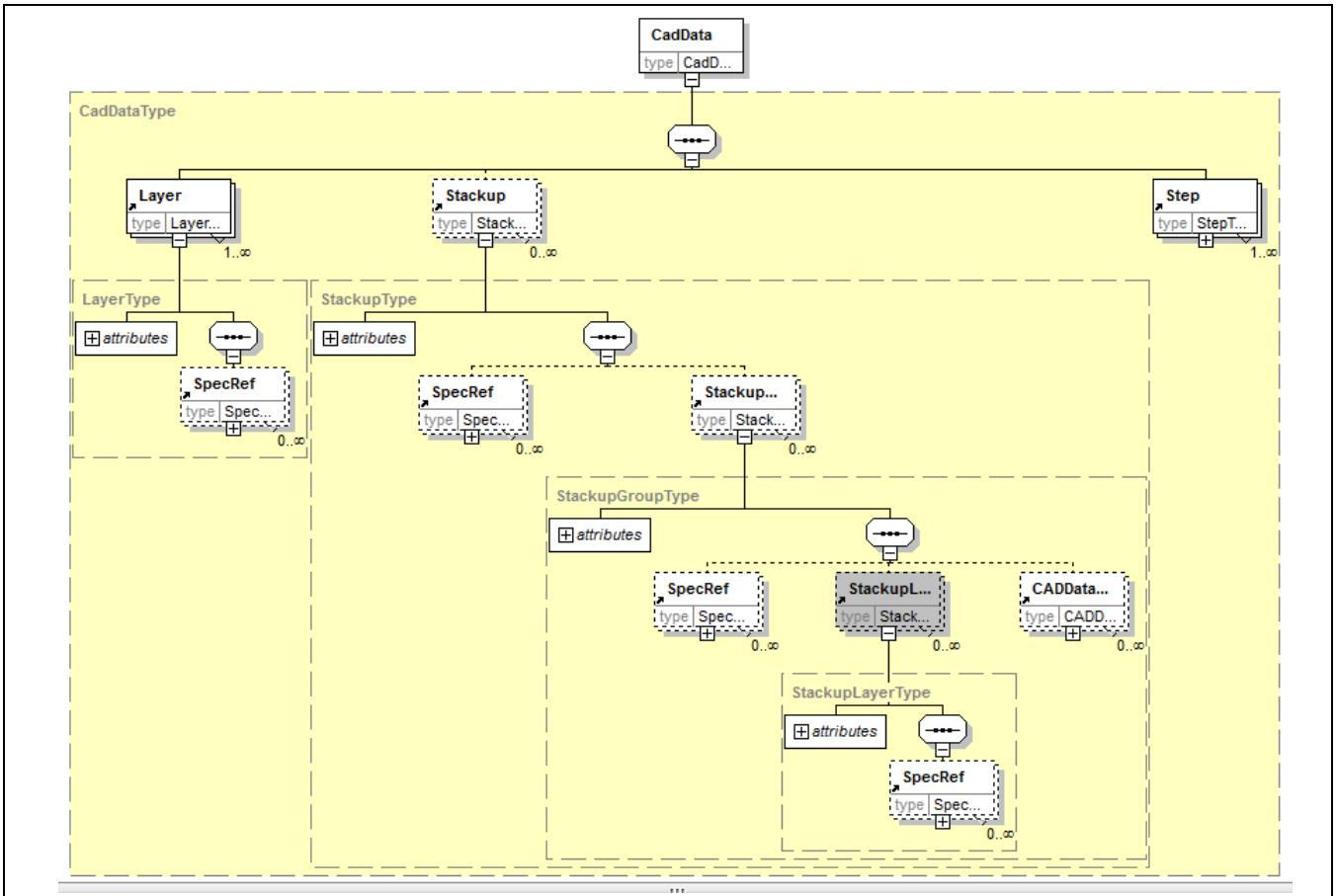
Layers are also identified by, `side`, `polarity`, `span`, and `SpecRef` that can be applied to help define each layer. CAD data layers are required in IPC-2581 to successfully hold ECAD layout information. These layers are not necessarily physical layers, but the myriad of layers that can be represented in the `Ecad` data but not actually fabricated into the bare board.

Layers, as the name implies, are sheets of two-dimensional data which, when laid on top of each other, create the Printed Circuit Assembly (unpopulated PCB and components or other related information). Some layers are physical layers that are laminated together to form the board. Other layers represent masks, films or phototools used to expose the board in a process that applies materials selectively on the outer layers of the boards. Some layers contain only drawings and annotations, which are not put physically on the board but can be used to further define it. These layers are organized in the IPC-2581 file by their specific type.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
CadData	CadDataType	A nested element containing the actual Cad data describing the printed board and printed board assembly characteristics.	1-1
Layer	LayerType	A nested element containing the different layer information represented in the CAD data base which describes the electronic assembly to be manufactured.	1-n
Stackup	StackupType	A nested element containing the construction information for the printed board. If the file contains more than one Stackup the unique name assignment is required for each board configuration	0-n
Step	StepType	The `Step` element consists of multiple sub-elements each intended to help describe the different steps needed in the board fabrication, or assembling the electronic product.	1-n

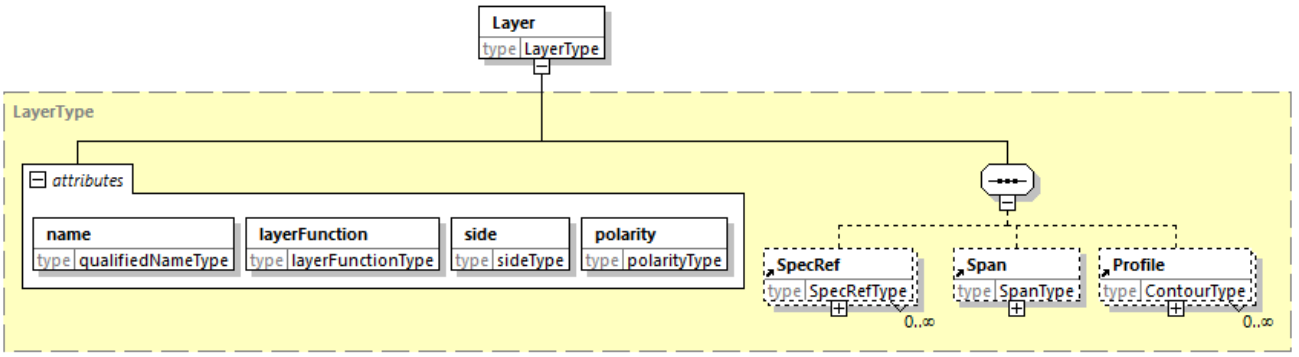
Everything about the design starts in CadData. The relationships are expanded through the hierarchy of the layer and stackup elements.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
CadData	CadDataType	A nested element containing the actual Cad data describing the printed board and printed board assembly characteristics.	1-1
Layer	LayerType	A nested element containing the different layer information represented in the CAD data base which describes the electronic assembly to be manufactured.	1-n
Stackup	StackupType	A nested element containing the construction information for the printed board. If the file contains more than one Stackup the unique name assignment is required for each board configuration	0-n
Step	StepType	The `Step` element consists of multiple sub-elements each intended to help describe the different steps needed in the board fabrication, or assembling the electronic product.	1-n

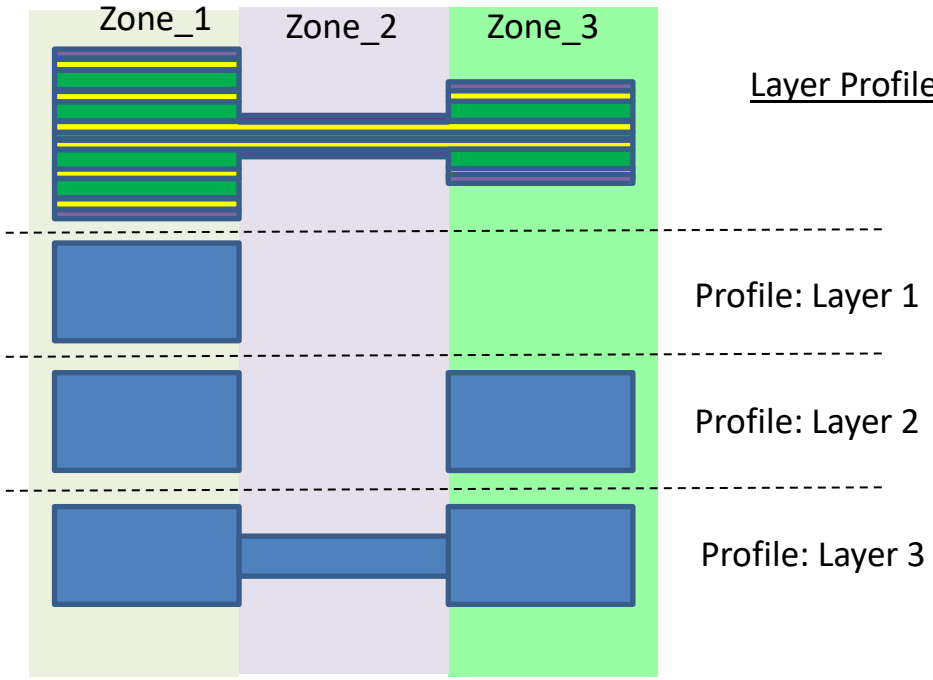
8.2.1 Layer

The `Layer` element describes the characteristics of specific layers. The layers may be for the board or the assembly and may be individual characterization or those of the board fabrication panel and the arrangement of boards in the assembly pallet. There are also documentation, tooling and miscellaneous layers. The `layerFunction` helps to identify the purpose of the layer.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Layer	LayerType	A nested element containing the different layer information represented in the CAD data base which describes the electronic assembly to be manufactured.	1-n
name	qualifiedNameType	The identification of the Cad data element identifying a particular layer. If the layerFunction STACKUP_COMPOSITE is indicated the unique name assigned shall match the name assigned to the related `Stackup` element	1-1
layerFunction	layerFunctionType	The type of layer and its main use as established by the following: ASSEMBLY | BOARDFAB | BOARD_OUTLINE | CAPACITIVE | COATINGCOND | COATINGNONCOND | COMPONENT | COMPONENT_BOTTOM | COMPONENT_TOP | COMPONENT_EMBEDDED | COMPONENT_FORMED | CONDFILM | CONDFOIL | CONDUCTIVE_ADHESIVE | CONDUCTOR | COURTYARD | DIELBASE | DIELCORE | DIELPREG | DIELADHV | DIELBONDPLY | DIELCOVERLAY | DOCUMENT | DRILL | FIXTURE | GLUE | GRAPHIC | HOLEFILL | SOLDERBUMP | PASTEMASK | LANDPATTERN | LEGEND | MIXED | OTHER | PIN | PLANE | PROBE | RESISTIVE | SIGNAL | SILKSCREEN | SOLDERMASK | SOLDERPASTE | STACKUP_COMPOSITE | REWORK | ROUT | V_CUT | EDGE_CHAMFER | EDGE_PLATING | THIEVING_KEEP_INOUT | STIFFENER	1-1
side	sideType	A fixed field parameter that defines the side of the layer. The fixed attribute is one of the following TOP | BOTTOM | BOTH | INTERNAL | ALL | NONE	1-1
polarity	polarityType	Applies for layers of type signal, power/ground or mixed. In such layers, positive means that the layer features represent copper. NEGATIVE means that the layer features represent laminate. For example, on a negative power/ground layer, features represent clearances. All other layers should be defined as positive. POSITIVE is the default.	1-1
SpecRef	SpecRefType	The identification of a particular specification(s) from the SpecificationType that is able to provide additional details or instructions that apply to the board or assembly descriptions of the Layer. The linkage is provided through the specific “specificationId” (spec Name)	0-n
Span	SpanType	A nested element where the field may be added to layers of type drill and route that are either buried or blind. In such a case, it represents the start and end board layers between which drilling/routing is done. If the drill is going through the board a span subsection will NOT be included.	0-1

Profile	ContourType	One or more outlines as seen at the `Layer`, caused by variations in the stackup cross section across the stackup zones in a rigid-flex design. The profile is affectively a 'slice' of the stackup at this `Layer`, made horizontally, that is parallel to the board. The absence of `Profile` indicates the main profile defined in the the `Step` should be referenced.	0-n



There is a close relationship between the `Layer` and `Step` elements of the IPC-2581 format. The correlation exists between the `Step` elements and attributes and the `layerFunction` attribute of the particular layer.

In general, the `Step` elements and their respective attributes have a significant purpose. Since there may be many steps in a IPC-2581 file, users usually identify the step `name` attribute as a method to group steps that relate to a particular purpose. The following are the recommended organizational structures:

BOARD for all the 'important' steps representing the graphics of the board itself.

BOARDPANEL for all the 'important' steps representing the graphics of the board panel itself.

ASSEMBLY for all the 'important' steps representing the graphics of the assembly itself.

ASSEMBLYPALLET for all the 'important' steps representing the graphics of the assembly pallet itself.

DOCUMENTATION for all the 'important' steps representing the documentation of the board or the assembly.

TOOLING for all the 'important' steps representing the tooling used on the board or the assembly.

COUPON for test coupons that are embedded in the design of the board or assembly.

MISCELLANEOUS, for all the remaining steps that do not have a home in any of the other context identification.

The purpose of a group of step elements should relate to the layer descriptions which are identified by their attributes that include the mandatory requirements of name, layerFunction, side, and polarity. Due to the layerRef attributes of several Step elements, the recommendations shown in Table 5 apply to good file management. Table 5 shows all possible combinations of the layerFunction attributes and their potential characteristics. The designation to make the link to the Bill of Material has been added to the table where appropriate and consists of Material Designations (MatDes), Document Designations (DocDes), Component Designations (RefDes) and Tooling Designations (ToolDes).

Table 5 Step Elements to Layer Attribute Recommendations

Key: A=TOP, B=BOTTOM, C=INTERNAL, D=OTHER, 1=POSITIVE, 2=NEGATIVE

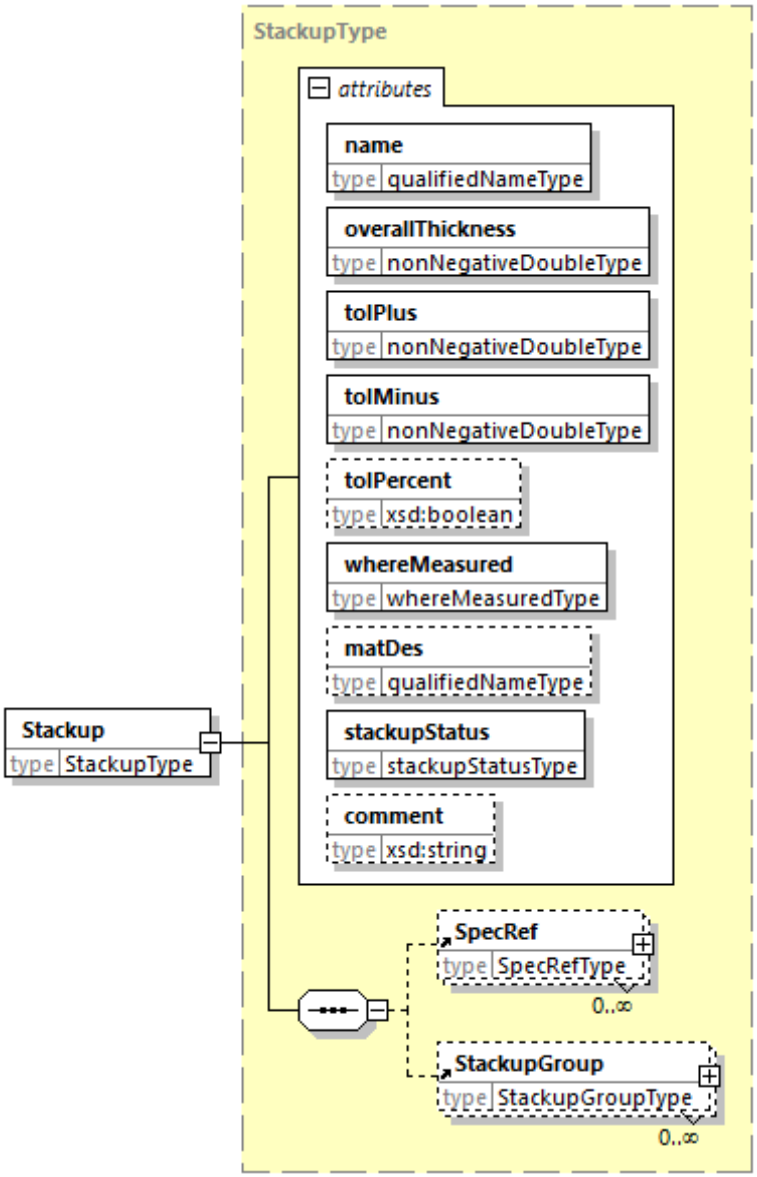
Step elements that describe:	Layer Attributes				
	Layer Type	layerFunction		side letters for reference only	Polarity numbers for reference only
All Possible Combinations	MATERIAL	MatDes	CAPACITIVE	C	1
		MatDes	COATINGCOND	A | B | C	1
		MatDes	COATINGNONCOND	A | B | C	1
		MatDes	COMPONENT_FORMED	A | B | C	1
		MatDes	CONDFILM	A | B | C	1
		MatDes	CONDFOIL	A | B | C	1
		MatDes	CONDUCTOR	A | B | C	1
		MatDes	CONDUCTIVE_ADHESIVE.	A | B | C	1
		MatDes	DIELBASE	C	1
		MatDes	DIELCORE	C	1
		MatDes	DIELPREG	C	1
		MatDes	DIELADHV	C	1
		MatDes	DIELBONDPLY	C	1
		MatDes	DIELCOVERLAY	C	1
		MatDes	GLUE	A | B	1
		MatDes	HOLEFILL	A | B	1
		MatDes	LEGEND	A | B	1
		MatDes	MIXED	A | B | C	1
		MatDes	PASTEMASK	A | B	1 | 2
		MatDes	PLANE	A | B | C	1
		MatDes	RESISTIVE	C	1
		MatDes	SIGNAL	A | B | C	1
		MatDes	SILKSCREEN	A | B	1
		MatDes	STACKUP_COMPOSITE	D	1
		MatDes	SOLDERBUMP	A | B	1
		MatDes	SOLDERMASK	A | B	1
		MatDes	SOLDERPASTE	A | B	1
		MatDes	EDGE_PLATING	A | B	1
	PROCESS	MatDes	STIFFENER	A | B	
		ToolDes	DRILL	A | B	1
		ToolDes	FIXTURE	A | B	1
		ToolDes	ROUTE	A | B	1
		ToolDes	SCORE	A | B	1
		ToolDes	PROBE	A | B	1
		ToolDes	V_CUT	A | B	1
		ToolDes	EDE_CHAMFER	A | B	1
	DOCUMENTATION	DocDes	ASSEMBLY	A | B	1
		DocDes	BOARD_FAB	?	1
		DocDes	BOARD_OUTLINE	A	1
		DocDes	COMPONENT_TOP	A	1
		DocDes	COMPONENT_BOTTOM	B	1
		DocDes	COMPONENT_EMBEDDED	C	1
		DocDes	COURTYARD	A | B	1
		DocDes	DOCUMENT	A | B | C	1
		DocDes	GRAPHIC	A | B | C	1
		DocDes	LANDPATTERN	A | B | C	1
		DocDes	REWORK	A | B | C	1
		DocDes	THIEVING_KEEP_INOUT	A | B	1

When combining steps that describe BOARD and COUPON information on the same PANEL, the layer construction **shall** be identical between those elements being instantiated on the same panel. They also refer to the layering so that it is consistent such that the top layer is identical for all steps referenced in the panel construction.

The viewer should always display the graphical features but store the layer polarity as an attribute.

8.2.2 Stackup

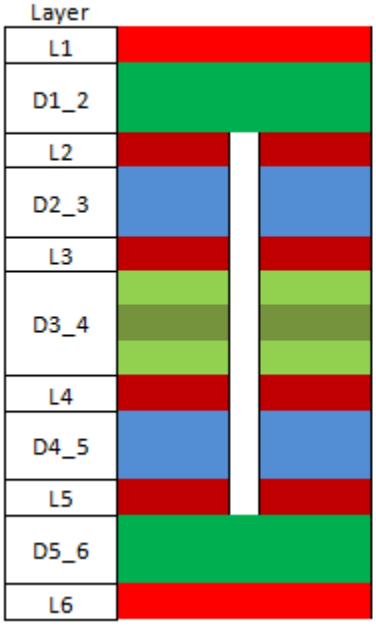
The `Stackup` element represents the construction for the printed board. The `Stackup` element consists of several sub-elements that help to define various sections of the construction permitting the description of core material or prepreg definition. These are accomplished in the `StackupGroup` element. The attributes of the `Stackup` element represent the finished board and apply to the entire board characteristics. The reference in the Bill of Material is to the finished board and as such uses theMatDes identification as the BomDesType.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Stackup	StackupType	A nested element containing the construction information for the printed board.	0-1

name	qualifiedNameType	A unique name assigned by the user to the printed board. The name must be unique and match the name used for the layerFunction STACKUP_COMPOSITE assigned by the `CadData` element	1-1
overallThickness	nonNegativeDouble Type	Describes the overall nominal thickness of the finished printed board including all plating and coatings.	1-1
tolPlus	nonNegativeDouble Type	The plus tolerance that may be applied to the nominal thickness to set the printed board upper control limit.	1-1
tolMinus	nonNegativeDouble Type	The minus tolerance that may be applied to the nominal thickness to set the printed board lower control limit.	1-1
tolPercent	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lower case). True means that the attribute(s) `tolPlus` and `tolMinus` are a percentage. False indicates that they represent a nonNegativeDouble. The Default is “false”.	0-1
whereMeasured	whereMeasuredType	An enumerated string that may be one of the following: LAMINATE | METAL | MASK | OTHER that defines the location on the printed board, panel, or assembly where the overall thickness is to be measured.	1-1
matDes	qualifiedNameType	A unique reference designator, using the letters in Appendix C followed by a number, in order to represent the composite stackup that could be called for in a Bill of Material for the printed board construction	0-1
comment	Xsd:string	An optional attribute used to provide any special instructions about the layering, specification requirements or stackup of an HDI, multilayer, single-sided, or double-sided printed board.	0-1
stackupStatus	stackupStatusType	An enumerated string that may be one of the following: SPECIFIED | PROPOSED | APPROVED. Generally SPECIFIED would indicate a request from the OEM, PROPOSED would indicate a response from a fabricator, and APPROVED would indicate approval by the OEM of what the fabricator proposed. However specific circumstances may dictate otherwise.	1-1
SpecRef	SpecRefType	The identification of a particular specification(s) from the SpecificationType that is able to provide additional details or instructions that apply to the board or assembly descriptions of the Stackup. The linkage is provided through the specific “specificationId” (spec Name)	0-n
StackupGroup	StackupGroupType	A nested element containing in formation of the printed board construction. StackupGroup elements should be defined within the Stackup before they are referenced in the IPC-2581 file.	0-n

CAD View



BOM View

sequence	MatDes name
1	CF1
2	DP1
3	DC1
4	DP2
5	DP3
6	DP2
7	DC1
8	DP1
9	CF1

Initial Draft From Designer

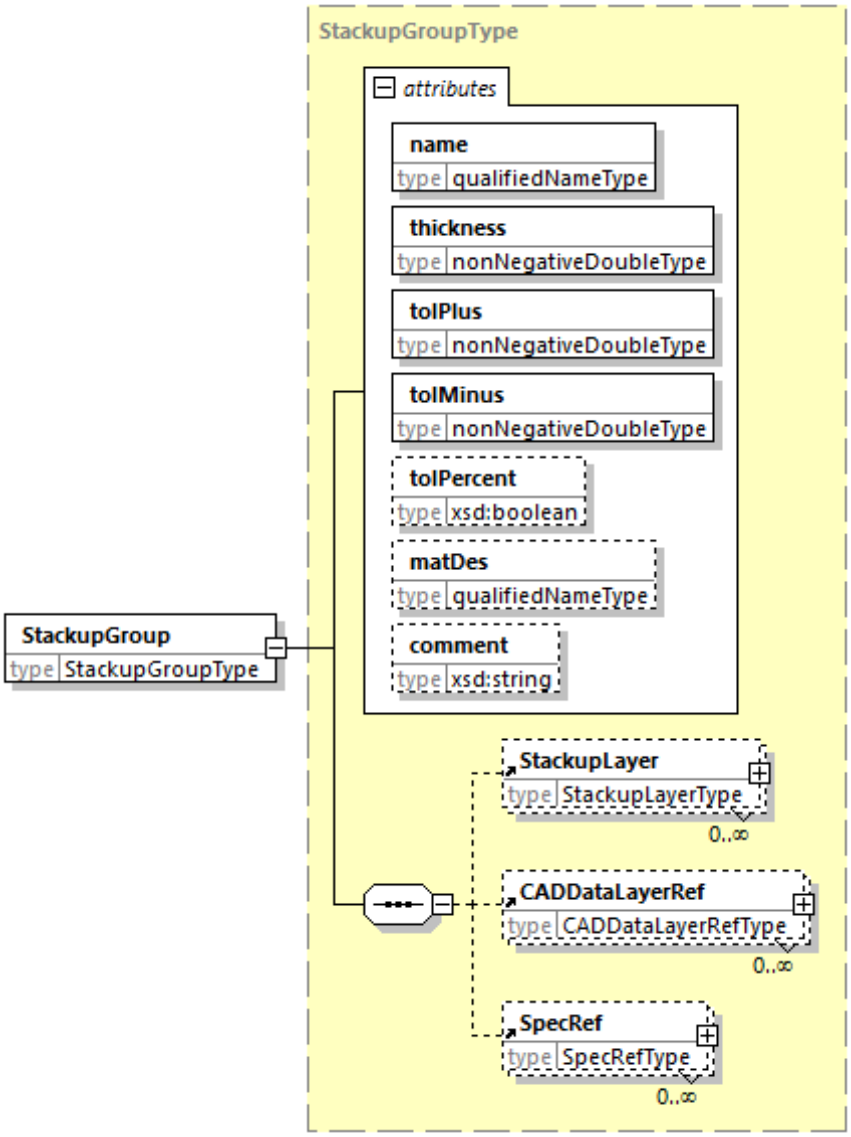
```
<Stackup overallThickness="1.1778" tolPlus="0.0" tolMinus="0.0" whereMeasured="OTHER">
  <StackupGroup name="AllStackupLayers" thickness="1.1778" tolPlus="0.0" tolMinus="0.0">
      <StackupLayer layerOrGroupRef=”L1” sequence="1" thickness="0.0559">
          </StackupLayer>
      <StackupLayer layerOrGroupRef=”D1_2”sequence="2" thickness="0.1321">
          </StackupLayer>
      <StackupLayer layerOrGroupRef=”L2” sequence="3" thickness="0.0559">
          </StackupLayer>
      <StackupLayer layerOrGroupRef=”D2_3”sequence="4" thickness="0.1625">
          </StackupLayer>
      <StackupLayer layerOrGroupRef=”L3” sequence="5" thickness="0.0559">
          </StackupLayer>
      <StackupLayer layerOrGroupRef=”D3_4”sequence="6" thickness="0.2532">
          </StackupLayer>
      <StackupLayer layerOrGroupRef=”L4” sequence="7" thickness="0.0559">
          </StackupLayer>
      <StackupLayer layerOrGroupRef=”D4_5”sequence="8" thickness="0.1625">
          </StackupLayer>
      <StackupLayer layerOrGroupRef=”L5” sequence="9" thickness="0.0559">
          </StackupLayer>
    <StackupLayer layerOrGroupRef=”D5_6” sequence=”10” thickness="0.1321">
          </StackupLayer>
      <StackupLayer layerOrGroupRef=”L6” sequence=”11” thickness="0.0559">
          </StackupLayer>
  </StackupGroup>
</Stackup>
```

Stackup back from FAB Vendor

```
<Stackup overallThickness="1.1778" tolPlus="0.0" tolMinus="0.0" whereMeasured="OTHER">
  <StackupGroup name="AllStackupLayers" thickness="1.1778" tolPlus="0.0" tolMinus="0.0">
      <StackupLayer layerOrGroupRef=”L1” sequence="1" thickness="0.0559" matDes=”CF1”/>
      <StackupLayer layerOrGroupRef=”D1_2”sequence="2" thickness="0.1321" matDes=”DP1”/>
      <StackupLayer layerOrGroupRef=”FabMe1st” thickness="0.7018"/>
      <StackupLayer layerOrGroupRef=”D5_6” sequence=”8” thickness="0.1321" matDes=”DP1”/>
      <StackupLayer layerOrGroupRef=”L6” sequence=”9” thickness="0.0559" matDes=”CF1”/>
  </StackupGroup>
  <StackupGroup name="FabMe1st" thickness="0.7018" tolPlus="0.0" tolMinus="0.0">
      <StackupLayer layerOrGroupRef=”laminate1” sequence=”3” thickness="0.2243" matDes=”DC1”/>
      <StackupLayer layerOrGroupRef=”inner_prepreg” thickness="0.2532"/>
      <StackupLayer layerOrGroupRef=”laminate2” sequence=”7” thickness="0.2243" matDes=”DC1”/>
  </StackupGroup>
  <StackupGroup name=”laminate1” thickness="0.2243" tolPlus="0.0" tolMinus="0.0">
      <StackupLayer layerOrGroupRef=”L2” thickness="0.0559"/>
      <StackupLayer layerOrGroupRef=”D2_3” thickness="0.1625"/>
      <StackupLayer layerOrGroupRef =”L3” thickness="0.0559"/>
  </StackupGroup>
  <StackupGroup name="laminate2" thickness="0.2243" tolPlus="0.0" tolMinus="0.0">
      <StackupLayer layerOrGroupRef=”L4” thickness="0.0559"/>
      <StackupLayer layerOrGroupRef=”D4_5” thickness="0.1625"/>
      <StackupLayer layerOrGroupRef=”L5” thickness="0.0559"/>
  </StackupGroup>
  <StackupGroup name=”inner_prepreg” thickness="0.2532" tolPlus="0.0" tolMinus="0.0" >
      <CADDataLayerRef layerRef=”D3_4”/>
      <StackupLayer sequence=”4” thickness="0.1015" matDes=”DP2”/>
      <StackupLayer sequence=”5” thickness="0.0502” matDes=”DP3”/>
      <StackupLayer sequence=”6” thickness="0.1015" matDes=”DP2”/>
  </StackupGroup>
</Stackup>
```

8.2.2.1 StackupGroup

The `StackupGroup` represents all the layers of the printed board and defines the order of their occurrence in the board construction. Individual layers may be identified as layer pairs. In this manner the user has the ability to define the characterization of the multilayer construction as well as preparing layer prelamination sequences. The order, however, must be in accordance with the description of the final board. If StackupGroup represents the finished board the attributes apply to the entire finished board; if StackupGroup identifies layer pairs the attributes apply to the specific layer pair only and that product is called for in the stackup by its qualified name in the order of occurrence in the total stackup.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
StackupGroup	StackupGroupType	A nested element containing in formation of the printed board construction.	0-n
name	qualifiedNameType	A unique name assigned to an individual or group of layers that make up the printed board. The name must be unique so that when a group becomes nested in the over all board it is referenced in the proper order of occurrence in the stackup.	1-1
thickness	nonNegativeDouble Type	The nominal thickness of the stackup group. If the stackup group represents the total board rather than a subset the thickness must match the information provided in the stackup element attributes.	1-1
tolPlus	nonNegativeDouble Type	The plus tolerance that may be applied to the nominal thickness to set the stackupGroup upper control limit.	1-1
tolMinus	nonNegativeDouble Type	The minus tolerance that may be applied to the nominal thickness to set the stackupGroup lower control limit.	1-1
tolPercent	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lower case). True means that the attribute(s) `tolPlus` and `tolMinus` are a percentage. False indicates that they represent a nonNegativeDouble. The Default is “false”.	0-1

matDes	qualifiedNameType	A unique reference designator in order to represent the `StackupGroup` as orderable item that could be called for in a Bill of Material. For example a laminated core.	0-1
comment	string	An optional attribute used to provide any special instructions about the layering, layer pairs or stackupGroup	0-1
StackupLayer	StackupLayerType	A nested element containing in all the layer formation as to how the printed board is constructed. If layer pairs are produced separately possibly containing buried vias they are defined as a separate group and then positioned in the appropriate order of their occurrence in the stackup. A relationship to the particular BOMItem should be established through use of the appropriate RefDes description provided in bomItem such as DB2 for a dielectric base material type. See Appendix C.	0-n
CADDataLayerRef	LayerType	A single or set of layer names that have been assigned by the CAD system and represent a link or comparison to the names assigned to the layers shown in the stackupGroup	0-n
SpecRef	SpecRefType	The identification of a particular specification(s) from the SpecificationType that is able to provide additional details or instructions that apply to the board or assembly descriptions of the StackupGroup. The linkage is provided through the specific “specificationId” (spec Name)	0-n

Example of Property References for a Stackup Spec

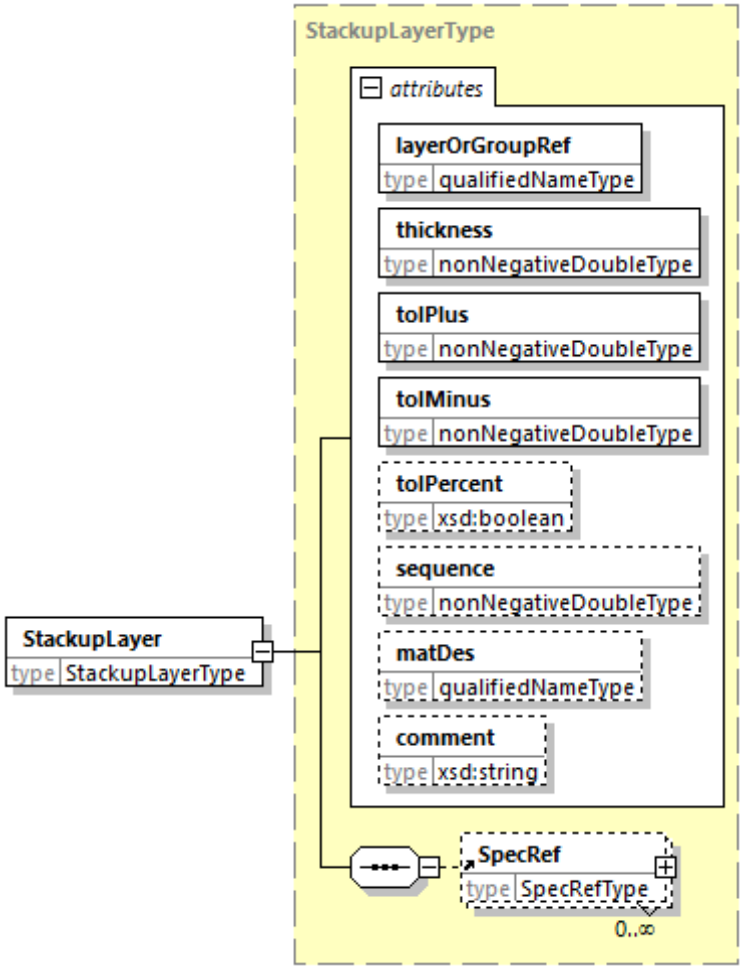
The example below shows:

a. A given Spec definition can contain multiple Specification Types
b. When a Specification Type for a StackupLayer Spec contains multiple values at different refValues, which could be required for simulation purposes (see DIELECTRIC_CONSTANT example in section 8.1.1.5) then a refValue can be a assigned within a Spec referenced at the Stackup level, to indicate a reference value for board fabrication

```
<Spec name = “property_references”>
  <Dielectric type=”DIELECTRIC_CONSTANT”>
    <Property refValue=”10.0e9” refUnit=”Hz”/>
  </Dielectric>
  <Dielectric type=”LOSS_TANGENT”>
    <Property refValue=”10.0e9” refUnit=”Hz”/>
  </Dielectric>
  <Conductor type=”CONDUCTIVITY”>
    <Property refValue=”25” refUnit=”CELCIUS”/>
  </Conductor>
  <Impedance value=”50” plusTol=”10” minusTol=”10”  tolPercent=“true” >
     <SingleEnded structure=”MICROSTRIP_MASK_COVERED”>
        <LineWidth value=”0.3” unit=”MM”/>
     </SingleEnded>
  </Impedance>
</Spec>
```

8.2.2.1.1 StackupLayer

The `Stackuplayer` represents all the layers of the printed board. Individual layers may be identified as material layers and are used to help define the layer pairs designated in StackupGroup.



LayerOrGroupRef	qualifiedNameType	A reference to an individual layer or a group that has been previously identified. A single sheet of copper foil may be a named layer and would thus apply to the group, as would a layer pair of copper clad laminate purchased from a laminator.	1-1
thickness	nonNegativeDouble Type	The thickness of the particular material being defined. The thickness matches nominal thickness of a predefined StackupGroup.	1-1
tolPlus	nonNegativeDouble Type	The plus tolerance that may be applied to the nominal thickness to set the layer upper control limit. If this is a stackupGroup the tolerances shall match those indicated in the group description	1-1
tolMinus	nonNegativeDouble Type	The minus tolerance that may be applied to the nominal thickness to set the layer lower control limit. If this is a stackupGroup the tolerances shall match those indicated in the group description	1-1
tolPercent	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lower case). True means that the attribute(s) `tolPlus` and `tolMinus` are a percentage. False indicates that they represent a nonNegativeDouble. The Default is “false”.	0-1
sequence	nonNegativeDouble Type	The specific order in which the layer materials or layer pairs are positioned in the stackup stating with the top of the board as sequence 1 followed by the entire board construction of imaged or un imaged layers/layer pairs.	0-1

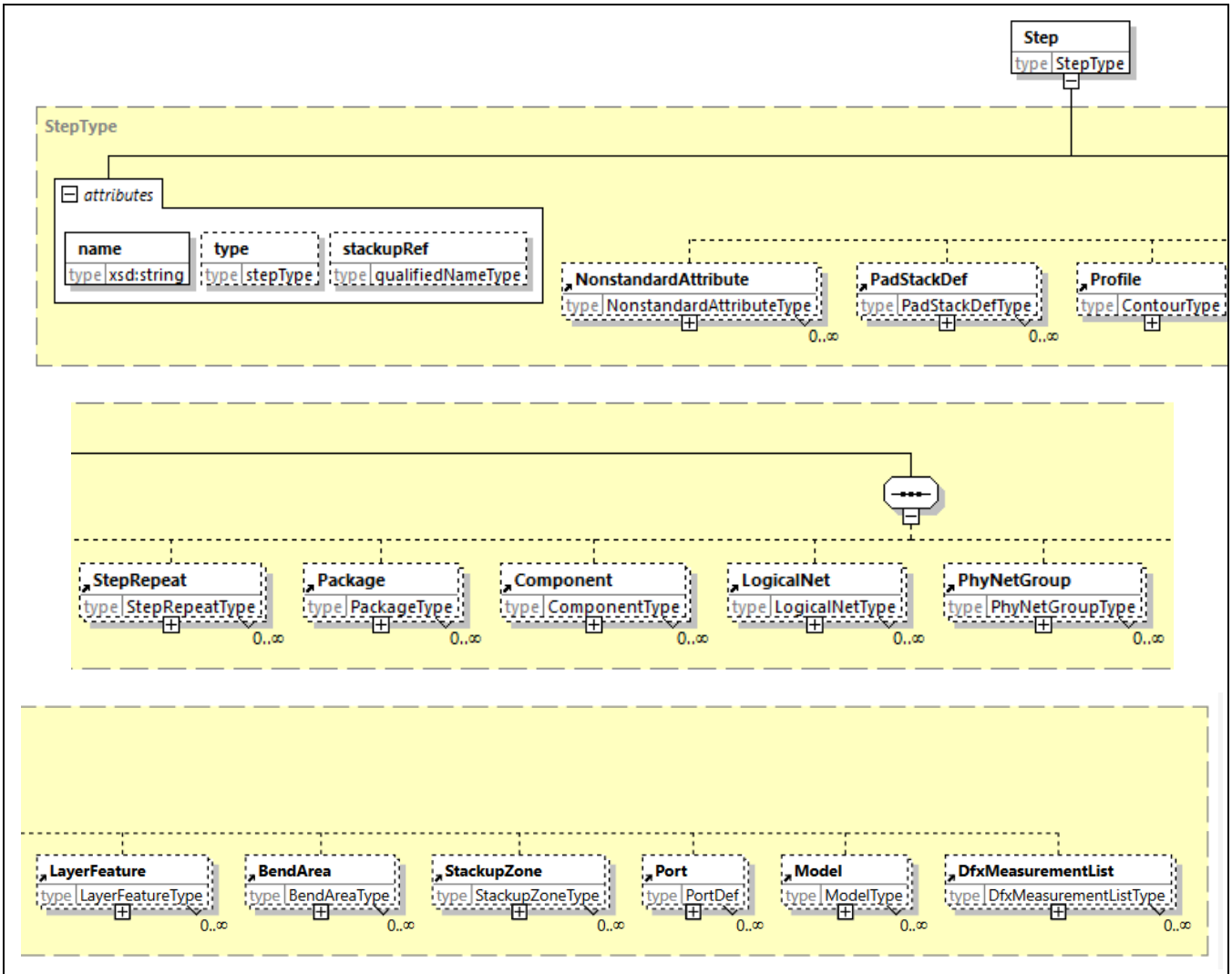
matDes	qualifiedNameType	A unique reference designator using the letters in Appendix C followed by a number in order to represent the `StackupLayer` that could be called for in a Bill of Material for the printed board construction	0-1
comment	string	An optional attribute used to provide any special instructions about the layering or stackup of a multilayer single-sided, or double-sided printed board.	0-1
SpecRef	SpecRefType	The identification of a particular specification(s) from the SpecificationType that is able to provide additional details or instructions that apply to the board or assembly descriptions of the Stackup Layer. The linkage is provided through the specific “specificationId” (spec Name)	0-n

8.2.3 Step

The `Step` element represents a collection of layers, each with a profile that defines its outer shape. The basic step is the Printed Circuit Assembly (PCA), the unpopulated board or other related information (e.g., documentation). In manufacturing, this basic step is often step and repeated (nested) inside a larger step (called array, or sub-panel). This array step can be further nested into another step; called a production panel. The `Ecad` element always contains at least one `Step`, but may contain several, some basic ones and others nesting previous steps.

The CAD `Step` tag can be repeated multiple times inside a job to represent several job Steps and their optional panelization. Each Step contains all the relevant information including `Datum`, `Profile`, `StepRepeat`, `LayerAttribute`, `Package`, `Component`, `VplComponent`, `LogicalNet` and `LayerFeature`.

All steps inside an `Ecad` element share the exact same layer structure, since they are ‘cut’ from the same basic panel. Each `layer`, in the list of layers, exists in every step, although in each `step` it may contain different graphical information or be empty

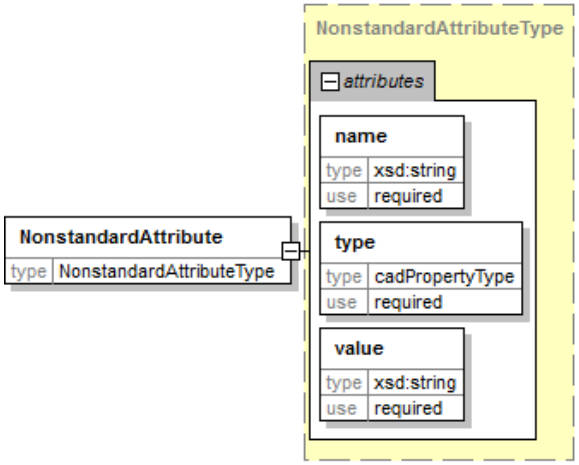


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Step	StepType	The `Step` element consists of multiple sub-elements each intended to help describe the different steps needed in the board fabrication, or assembling the electronic product.	1-n
name	string	The name assigned by the user for the individual Steps used to make up the layers identified in a particular `StepList`.	1-1
type	stepType	An optional enumerated string to identify what the Step represents. Choices are BOARD | ARRAY | IC	0-1
stackupRef	qualifiedNameType	An optional reference to an associated stackup, if there is another step with a different stackup (that is not a stackup zone)	0-1
NonstandardAttribute	Nonstandard AttributeType	A nested element that can be used to provide extra details regarding a particular “Step” description in terms of BOOLEAN, DOUBLE, INTEGER, or STRING data. The properties of the NonstandardAttribute become unique due to its’ Name.	0-n
PadStackDef	PadStackDefType	An element that describes the construction of various padstack types used in the design of the printed board, and provides a correlation to the usage of the padstack to those holes or pads that help to form the padstack.	0-n
Datum	LocationType	The `Datum` element defines the location of the point of origin for the individual `Step` file. The name of the `StepList` helps to associate the datum between boards and panels or arrays.	1-1

Profile	ContourType	The profile of all the elements in the `Step` established as a `Contour`.	0-1
StepRepeat	StepRepeatType	A nested element specifying the quantity and location of a "single piece" board `Step`, if this is the `Step` of a panel or assembly pallet.	0-n
Package	PackageType	Generic component package descriptions for use by the `Step` file schemas.	0-n
Component	ComponentType	A nested element list of component descriptions and their application to the electronic product. Each component references a package style from the `Package` section.	0-n
LogicalNet	LogicalNetType	A nested element list of logical net descriptions and their application to the electronic product.	0-n
PhyNetGroup	PhyNetGroupType	A nested element list of physical net descriptions and their application to the electronic product.	0-n
LayerFeature	LayerFeatureType	A nested element list of all the features associated with a specific layer and their application to the electronic product.	0-n
BendArea	BendAreaType	A nested element describing a bend in a flex section of a design	0-1
StackupZone	StackupZoneType	A nested element defining stackup zones in a rigid flex design	0-1
Port	PortType	A nested element defining a port in the `Step`, and its connection to other `Step(s)` or component(s)	0-1
Model	ModelType	A nested element defining a simple 3D model that can be used to describe coins, components, and other objects attached to the PCB	0-1
DfxMeasurementList	DfxMeasurement ListType	A nested element list of the recommended modifications of the design features, indicating the measurements made of the physical conditions that might be considered as manufacturing improvements.	0-n

8.2.3.1 NonstandardAttribute

The NonstandardAttribute element consists of various conditions that may be used in association with the `Step` element. The NonstandardAttribute element has several attributes which include a name, type and value thus making each NonstandardAttribute unique. The type attribute is a cadPropertyType and identifies the "value" attribute as being BOOLEAN, DOUBLE, INTEGER or STRING data.

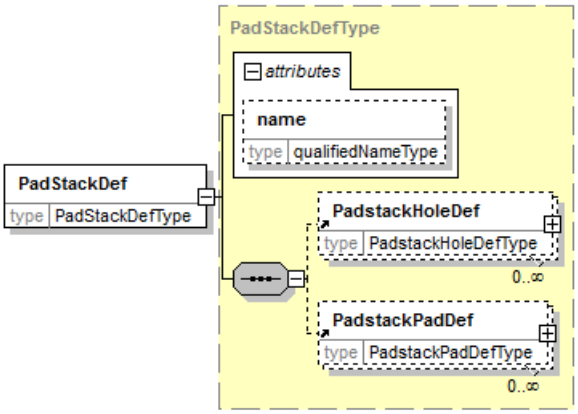


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Nonstandard Attribute	Nonstandard AttributeType	A nested element that can be used to provide extra details regarding a particular "Step" description in terms of BOOLEAN, DOUBLE, INTEGER, or STRING data. The properties of the NonstandardAttribute become unique due to its Name.	0-n
name	qualifiedNameType	A unique name assigned by the user to the NonstandardAttribute. The name must be unique and should match the purpose of the actions to be taken.	1-1

type	cadPropertyType	A set of enumerated string descriptions that identify the condition of the value requirement. They consist of BOOLEAN | DOUBLE | INTEGER or STRING data.	1-1
value	string	The detail requirements as they pertain to the particular Step description and interpreted - according to the type enumeration that is part of the particular NonstandardAttribute.	1-1

8.2.3.2 PadStackDef

The `PadStackDef` element consists of multiple padstacks types or descriptions taken from the CAD system and is intended to preserve the data from the layout system. The information noted pertain to the `CadProperty` of which the padstack is a part. The relationship is identified by the `CadProperty` unique name and is the original design file from the CAD system. The data becomes although redundant when the individual layered features are defined provides a reference for the padstack usage.

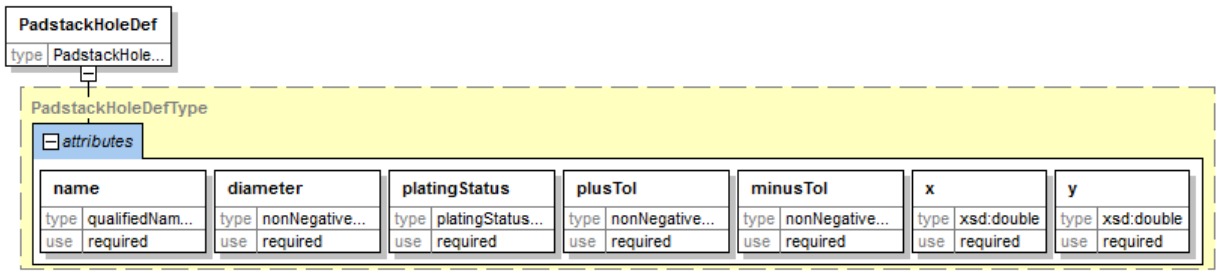


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
PadStackDef	PadStackDefType	An element that provides the characteristics of individual padstacks in the design, as established by the CAD system, and provides the link to the layer distribution of the padstacks individual characteristics.	0-n
name	qualifiedNameType	A unique name assigned to the construction of a particular padstack type.	0-n
PadstackHoleDef	PadstackHoleDefType	The hole description of a padstack including all its attributes as defined by the CAD system.	0-n
PadstackPadDef	PadstackPadDefType	The Pad description of a padstack including all its attributes as defined by the CAD system.	0-n

```
<Step name = "C027D17P">
   <PadStackDef name = " C027D17P ">
       <PadstackHoleDef name=”PSHD_1” Diameter= “.027” platingStatus=”PLATED” plusTol=”0.002” minusTol=”0.002” X=”0.0” Y=”0.0”/>
       <PadstackPadDef  layerRef=”TOP”  PadUse=”REGULAR” >
           <Location X = “0.0” Y=”0.0”/>
          <StandardPrimitive id = “CIRCLE_10”/>
        </PadstackPadDef>
        <PadstackPadDef  layerRef=”IL-1”  PadUse=”REGULAR” >
           <Location X = “0.0” Y=”0.0”/>
          <StandardPrimitive id = “CIRCLE_10”/>
        </PadstackPadDef>
       <PadstackPadDef  layerRef=”IL-1”  PadUse=”ANTIPAD” >
           <Location X = “0.0” Y=”0.0”/>
          <StandardPrimitive id = “PAD15”/>
        </PadstackPadDef>
       <PadstackPadDef  layerRef=”IL-1”  PadUse=”THERMAL” >
           <Location X = “0.0” Y=”0.0”/>
          <StandardPrimitive id = “FIGURE_THERMAL_25_+”/>
        </PadstackPadDef>
       <PadstackPadDef  layerRef=”IL-2”  PadUse=”REGULAR” >
           <Location X = “0.0” Y=”0.0”/>
          <StandardPrimitive id = “CIRCLE_10”/>
        </PadstackPadDef>
        <PadstackPadDef  layerRef=”IL-2”  PadUse=”ANTIPAD” >
           <Location X = “0.0” Y=”0.0”/>
          <StandardPrimitive id = “PAD15”/>
        </PadstackPadDef>
       <PadstackPadDef  layerRef=”IL-2”  PadUse=”THERMAL” >
           <Location X = “0.0” Y=”0.0”/>
          <StandardPrimitive id = “FIGURE_THERMAL_25_+”/>
        </PadstackPadDef>
       <PadstackPadDef  layerRef=”BOTTOM”  PadUse=”REGULAR” >
           <Location X = “0.0” Y=”0.0”/>
          <StandardPrimitive id = “CIRCLE_10”/>
        </PadstackPadDef>
   </PadStackDef>
</Step>
```

8.2.3.2.1 PadstackHoleDef

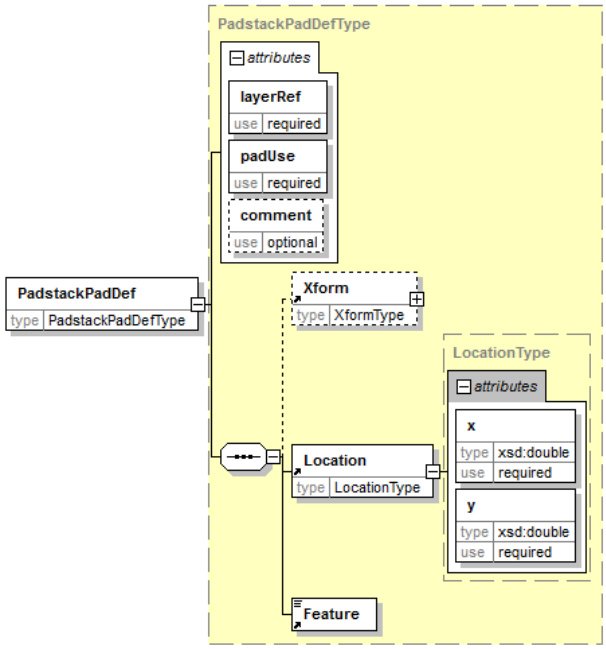
The `PadstackHoleDef` element associated with a padstack identifies the diameter, tolerance and plating status of a particular padstack location.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
PadstackHoleDef	PadstackHoleDefType	The hole description of the padstack including all its attributes.	0-1
name	qualifiedNameType	A unique identification of a particular padstack hole description.	1-1
diameter	nonNegativeDouble Type	The nominal diameter of the hole in the as-finished state.	1-1
platingStatus	platingStatusType	The type of hole defined as an enumerated string indicating PLATED | NONPLATED | VIA | VIA_CAPPED	1-1
plusTol	nonNegativeDouble Type	The plus tolerance that defines the variation permitted from the nominal hole-diameter.	1-1
minusTol	nonNegativeDouble Type	The minus tolerance that defines the variation permitted from the nominal hole-diameter.	1-1
x	double	The x-location of the hole.	1-1
y	double	The y-location of the hole.	1-1

8.2.3.2.2 PadstackPadDef

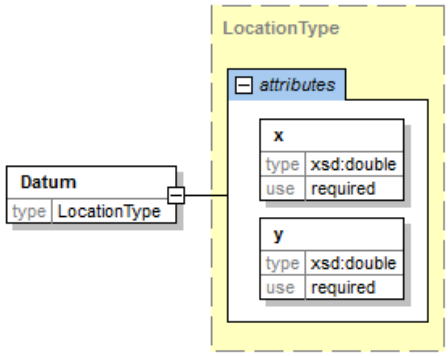
The `PadstackPadPDef` element is a specific graphic feature that becomes part of the padstack with a description of shape (Feature) and a location in order to establish the linkage to pads on individual layers.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
PadstackPadDef	PadstackPadDefType	An element that provides the definition of various pads that are used to make up a padstack within the CAD system and provides the ability to locate where these are used in the design of the board.	0-n
layerRef	qualifiedNameType	The reference to a specific layer element by it's "name" attribute. The layer referenced is where the pad image is located.	1-1
padUse	padUseType	An enumerated value of the pad type used on the particular layer, such as a power or signal or mask layer. The enumerations are REGULAR | ANTIPAD | THERMAL | OTHER. If OTHER is used a comment is required to explain the purpose of the pad.	1-1
comment	string	An explanation as to the condition of the Pad if OTHER is defined as the padUse. A comment may also be used if the pad is not connected to circuitry and has been removed by the manufacturer to improve clearance conditons.	0-1
Xform	XformType	An element that provides the ability to reset the point of origin of a predefined target that can be scaled, mirror imaged or rotated. (See 3.3)	0-1
Location	LocationType	The location of the image defined by the standard shape, user shape or a pre-defined standard shape of the pad. The image may have been reorientated by the Xform.	1-1
Feature	ABSTRACT	An embedded element that defines a substitution group of any `StandardShape` or `UserShape` that may be instantiated as a part of the `Pad` Description. A predefined `StandardShape or UserShape` may also be instantiated by its unique "id" when the feature is contained in the `DictionaryStandard or User`. When a reference is made to the dictionary predefined primitive the `Units` must match.	1-1

8.2.3.3 Datum

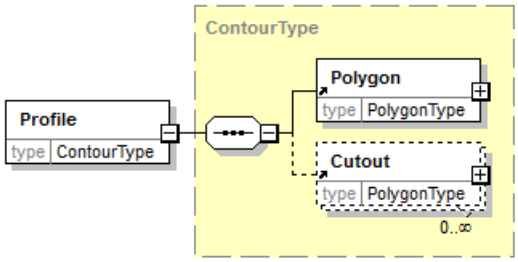
The `Datum` element of the `Step` schema (`StepType/Datum`) defines the location of the point of origin for the individual `Step` file. The unique name of the `Step` helps to associate the datum between boards and panels or pallets.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Datum	LocationType	A nested element containing the datum origin for the overall Step.	1-1
x	double	The x location of the datum.	1-1
y	double	The y location of the datum.	1-1

8.2.3.4 Profile

The `Profile` element of the `Step` schema (`StepTypeProfile`) defines the exact periphery of the board or assembly and therefore all the characteristics of the `Step` element.



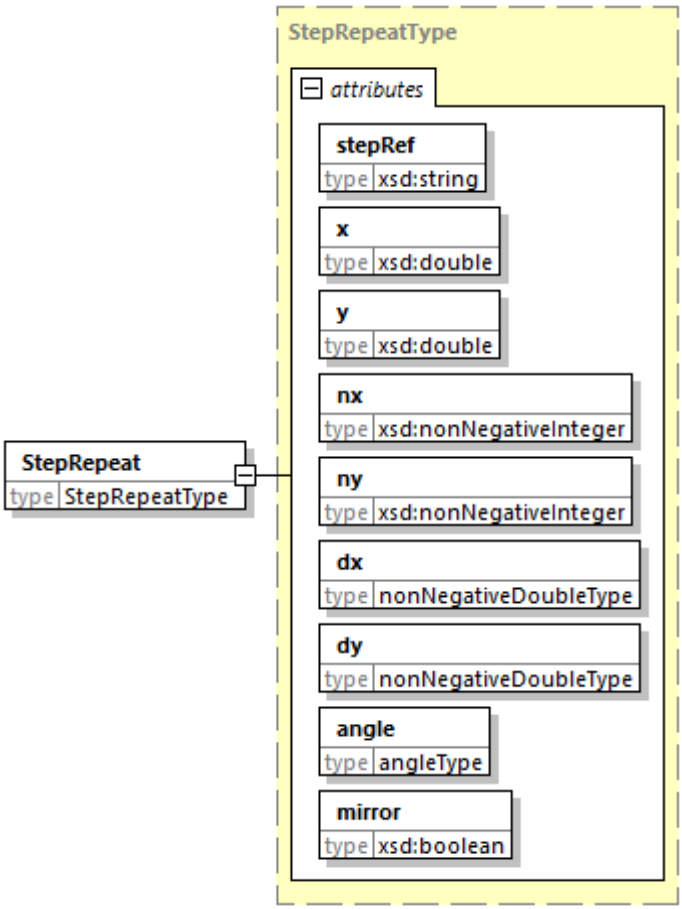
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Profile	ContourType	The profile is a `contourType` and thus describes the periphery that encompasses all the elements in the `Step`.	1-1
Cutout	CutoutType	A polygon closed shape whose edges do not cross, which adopts the coordinates of the original polygon, however represents the absence of material within the original polygon shape.	0-n

```
<Step name = "KarensSingleBoard">
    <Datum x = "10.00" y = "10.00"/>
      <Profile>
        <Polygon>
          <PolyBegin x = "0.00" y = "10.00"/>
          <PolyStepSegment x = "0.00" y = "90.00"/>
          <PolyStepCurve x = "10.00" y = "100.00" centerX = "10.00" centerY = "90.00" clockwise = “true”/>
          <PolyStepSegment x = "200.00" y = "100.00"/>
          <PolyStepSegment x = "200.00" y = "50.00"/>
          <PolyStepSegment x = "150.00" y = "50.00"/>
          <PolyStepSegment x = "150.00" y = "0.00"/>
          <PolyStepSegment x = "10.00" y = "0.00"/>
          <PolyStepCurve x = "0.00" y = "10.00" centerX = "10.00" centerY = "10.00" clockwise = “true”/>
        </Polygon>
      </Profile>
  </Step>
```

8.2.3.5 StepRepeat

The `StepRepeat` elements provides information for steps representing panels or assembly pallets. Coupons may also use this feature to step the coupon description on the borders of the panel. The layer descriptions of any `Board` and `Coupon` combined in a `Panel` description must be of the same construction. The attribute stepRef is restricted in the XML schema to the unique name of the `Step` element referenced.

If the features of a `StepRepeat` function become unique due to different characteristics such as one `Step` is at 90 degrees while the next `Step` is at 180 degrees two separate `stepRepeat` elements are required.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
StepRepeat	StepRepeatType	A nested element list containing the Step and Repeat functions that impact the information of the electronic product.	0-n
stepRef	string	A reference to the step that should be replicated on the panel.	1-1
x	double	The X point of origin where the first step should be placed in relationship to the datum. This may be coincident or may be offset from the datum of the particular step.	1-1
y	double	The Y point of origin where the first step should be placed in relationship to the datum. This may be coincident or may be offset from the datum of the particular step.	1-1
nx	nonNegativeInteger	The total number of steps in the X direction, including the first step	1-1
ny	nonNegativeInteger	The total number of steps in the Y direction, including the first step	1-1
dx	nonNegativeDouble Type	The delta distance in the positive X direction from one step's origin to the next, starting from the first step's origin x, y (not necessarily the datum).	1-1
dy	nonNegativeDouble Type	The delta distance in the positive Y direction from one step's origin to the next, starting from the first step's origin x, y (not necessarily the datum).	1-1
angle	angleType	A unique angle to allow rotation of the `StepRepeat` image description where "0°" is as defined with the angle descriptions being counterclockwise (i.e., 45° 90°) from the horizontal zero angle.	1-1
mirror	boolean	An enumerated string is either "true" or "false" (part of the 3WC standard – must be lower case). True equals that the image is mirrored; False indicates that it is not.	1-1

```
<Step name = "KarensAssemblyPanel">
    <Datum x = "0.00" y = "0.00"/>
        <Profile>
            <Polygon>
                <PolyBegin x = "0.00" y = "0.00"/>
                <PolyStepSegment x = "0.00" y = "427.00"/>
                <PolyStepSegment x = "260.00" y = "427.00"/>
                <PolyStepSegment x = "260.00" y = "0.00"/>
                <PolyStepSegment x = "0.00" y = "0.00"/>
            </Polygon>
        </Profile>
        <StepRepeat stepRef = "KarensSingleBoard" x = "110.00" y = "20.00" nx = "1" ny = "1" dx = "120.00" dy = "207.00" angle =
                        "90.00" mirror = “false”/>
        <LayerFeature layerRef = "1-Top Signal">
            <Set polarity = "POSITIVE">
            <GlobalFiducial>
                <Location x = "250.00" y = "10.00"/>
                <Circle diameter = "1.00"/>
            </GlobalFiducial>
            <GlobalFiducial>
                <Location x = "250.00" y = "417.00"/>
                <Circle diameter = "1.00"/>
            </GlobalFiducial>
            <GlobalFiducial>
                <Location x = "10.00" y = "10.00"/>
                <Circle diameter = "1.00"/>
            </GlobalFiducial>
            <BadBoardMark>
                <Location x = "190.00" y = "5.00"/>
                <Circle diameter = "1.50"/>
            </BadBoardMark>
            <BadBoardMark>
                <Location x = "70.00" y = "5.00"/>
                <Circle diameter = "1.00"/>
            </BadBoardMark>
            <BadBoardMark>
                <Location x = "190.00" y = "213.00"/>
                <Circle diameter = "1.00"/>
            </BadBoardMark>
            <BadBoardMark>
                <Location x = "70.00" y = "213.00"/>
                <Circle diameter = "1.00"/>
            </BadBoardMark>
            <GoodPanelMark>
                <Location x = "250.00" y = "213.00"/>
                <Donut shape = "ROUND" outerDiameter = "1.50" innerDiameter = "0.80"/>
            </GoodPanelMark>
        </Set>
    </LayerFeature>
</Step>
```

The following are examples of the step and repeat functions

The following are 6 Panelization use cases that the IPC-2580 series must consider in its output. The number in the dark green area refers to a Design (in the last example, there are 4 unique Designs placed within a Panel).

8.2.3.5.1 Single

A single Design is placed in a Panel.

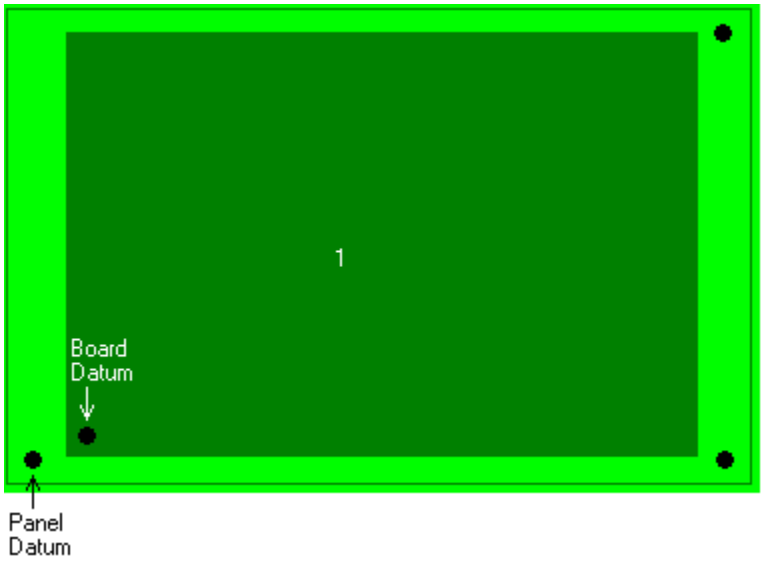


Figure 9 - Single Design within a Panel

Requires one StepRepeat element that positions the design on the panel at the appropriate X-Y location.

<StepRepeat stepRef="SingleBoard" x="10.0" y="7.0" nx="1" ny="1" dx="0.0" dy="0.0" angle="0.00" mirror="false" />

8.2.3.5.2 Single Array

One Design is placed in the same orientation throughout the panel. The size of the matrix is nx in the x direction, and ny in the y direction.

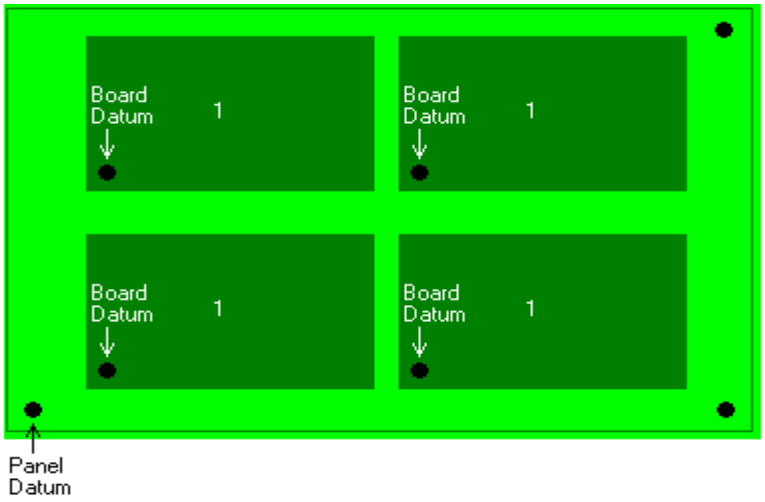


Figure 10 - Design Array within a Panel

Requires one StepRepeat element that references the design on the panel at the initial X-Y location.

<StepRepeat stepRef="SingleBoard" x="23" y="2.13" nx="2" ny="2" dx="152.5" dy="78.2" angle="0.00" mirror="false" />

8.2.3.5.3 Double Array

A single design, but arrayed in two distinct row X column matrices. This panelization method is to use the maximum area of the PCB fabricator's raw panel stock.

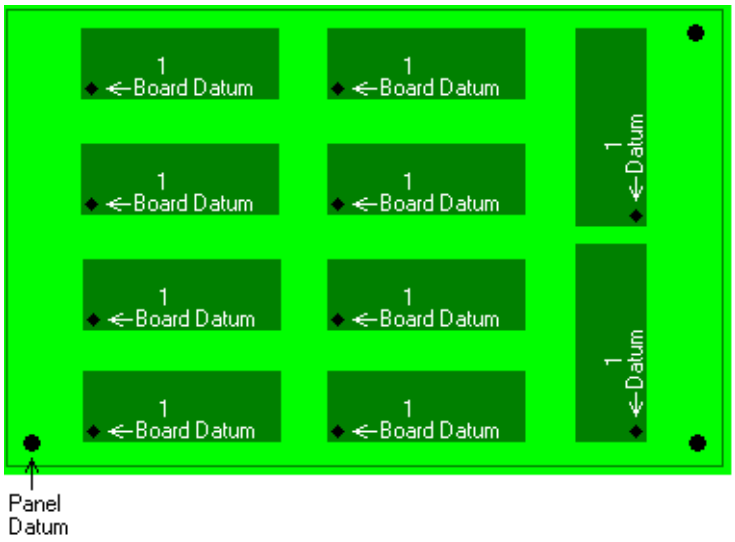


Figure 11 - Design Array based on two matrices within a Panel

Requires two `StepRepeat` elements. The first `StepRepeat` element initially positions the design on the panel at the appropriate X-Y location. Then the design is stepped such that there is a total of two in the X direction and four in the Y direction. The second `StepRepeat` element orients the design on a 90° angle. This new orientation is then positioned on the panel at the appropriate X-Y location, and stepped such that there is a totgal of one in the X direction and two in the Y direction.

<StepRepeat stepRef="board_1" x="60.0" y="12.0" nx="2" ny="4" dx="500" dy="100.00" angle="0.00" mirror="false" />

<StepRepeat stepRef="board_1" x="950.00" y="12.0" nx="1" ny="2" dx="0" dy="190.0" angle="90.00" mirror="false" />

8.2.3.5.4 Tiled

This example is to place a pair of single designs 180° out of phase with each other.

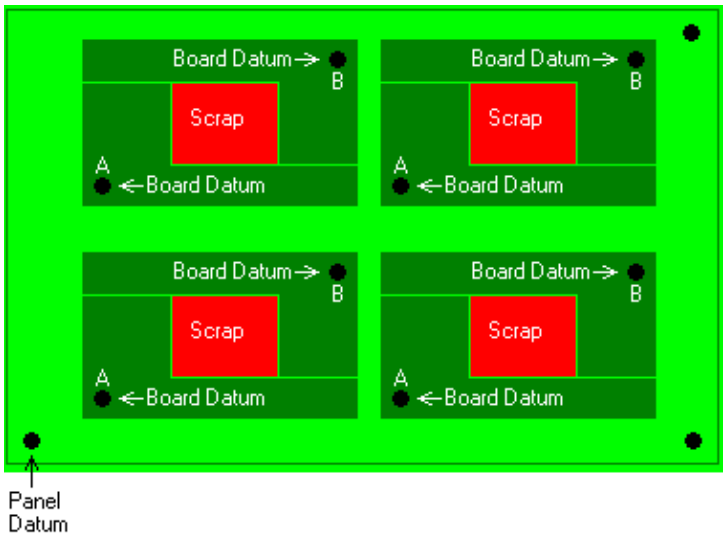


Figure 12 - Design tiled as pairs within a Panel

Requires two `StepRepeat` elements, where the first positions the design on the panel at the appropriate X-Y location, and then steps it such that there is a total of two in the X direction and two in the Y direction. The second `StepRepeat` element rotates the design 180°, and then positions the new orientation on the panel at the appropriate X-Y location. This reoriented design is then stepped such that again there is a total of two in the X direction and two in the Y direction. An appropriate initial X-Y location may be in the lower left portion of the panel with a positive X-Y step or in the upper right corner of the panel with a negative dx,dy step.

<StepRepeat stepRef=”board_1" x="30.0" y="20.0" nx="2" ny="2" dx="300.0" dy="100.0" angle="0.00" mirror=“false” />

<StepRepeat stepRef="board_1" x="1200.00" y="800.0" nx="2" ny="2" dx="-300.0" dy="-100.0" angle="180.00" mirror=“false” />

8.2.3.5.5 Flipped

“Flip” board pairing. This methodology is recent. The most important concern is that the board must be a symmetrical stackup, which means that the layer stackup must be verified to allow this type of panelization.

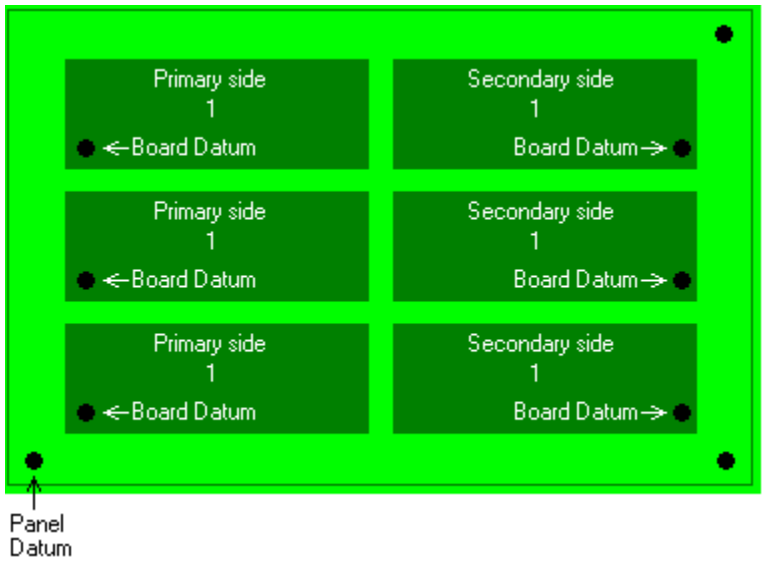


Figure 13 - Design flipped as a pair within a Panel

Requires two `StepRepeat` elements, the first positioning the design on the panel at the appropriate X-Y location, and then repeating such that there are a total of one in the X direction and 3 in the Y direction. The second `StepRepeat` element identifies the design as a mirror image, which is then positioned on the panel at the appropriate X-Y location, and repeated like the first.

<StepRepeat stepRef=”board_1" x="20.0" y="20.0" nx="1" ny="3" dx="0" dy="75.00" angle="0.00" mirror=“false” />

<StepRepeat stepRef=”board_1" x="250.0" y="20.0" nx="1" ny="3" dx="0" dy="75.00" angle="0.00" mirror=“true” />

8.2.3.5.6 Multiple Designs

The most important consideration with placing multiple designs (each number represents a unique design) within one panel is that all designs need to have been created within the same layer stackup. The advantage is that an entire product can be assembled/tested all at once. The disadvantage is that if one board of the panel has a problem (either with part availability or performance), this can lead to several additional scheduling/building/etc. problems as well.

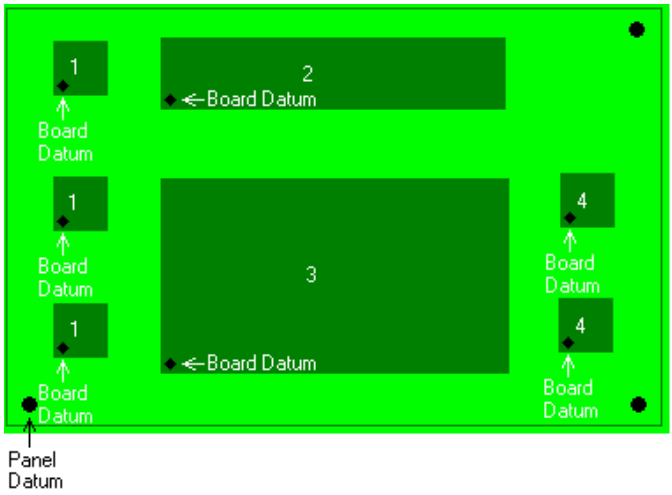


Figure 14 - Multiple Designs placed within a Panel

The panel requires four `StepRepeat` elements. Design 1 is positioned on the panel at the appropriate X-Y location, and is then stepped such that there is a total of one in the X direction and three in the Y direction. Design 2 references a different Step and is positioned on the panel at the appropriate X-Y location. Design 3 references a different Step and is positioned on the panel at the appropriate X-Y location. For both Designs 2 and 3 the `dx` and `dy` attribute values are zero The fourth `StepRepeat` element identifies Design 4 as another different step, which is positioned on the panel at the appropriate X-Y location, and stepped such that there is a total of one in the X direction and two in the Y direction.

<StepRepeat stepRef="board_1" x="57.41629" y="12.30125" nx="1" ny="3" dx="0" dy="102.19426" angle="0.00" mirror="false"/> <StepRepeat stepRef="board_2" x="260.56222" y="467.09018" nx="1" ny="1" dx="0" dy="0" angle="0.00" mirror="false"/>
<StepRepeat stepRef="board_3" x="260.56222" y="110.17554" nx="1" ny="1" dx="0" dy="0" angle="0.00" mirror="false"/>
<StepRepeat stepRef="board_4" x="927.85875" y="17.57629" nx="1" ny="2" dx="0" dy="138.0" angle="0.00" mirror="false"/>

See Appendix B for an example of an XML instance file.

8.2.3.5.7 Tooling Step and Repeat

The Step and repeat function can also be used to duplicate features that are to become part of the board, panel, or pallet. Figure 15 shows a series of slots that need to be included in a set of layers of the stackup. They may be replicated on each sheet that must be matched within the stackup.

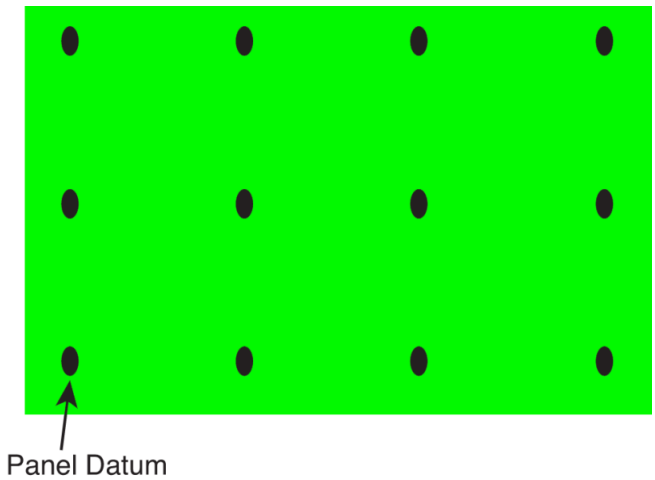


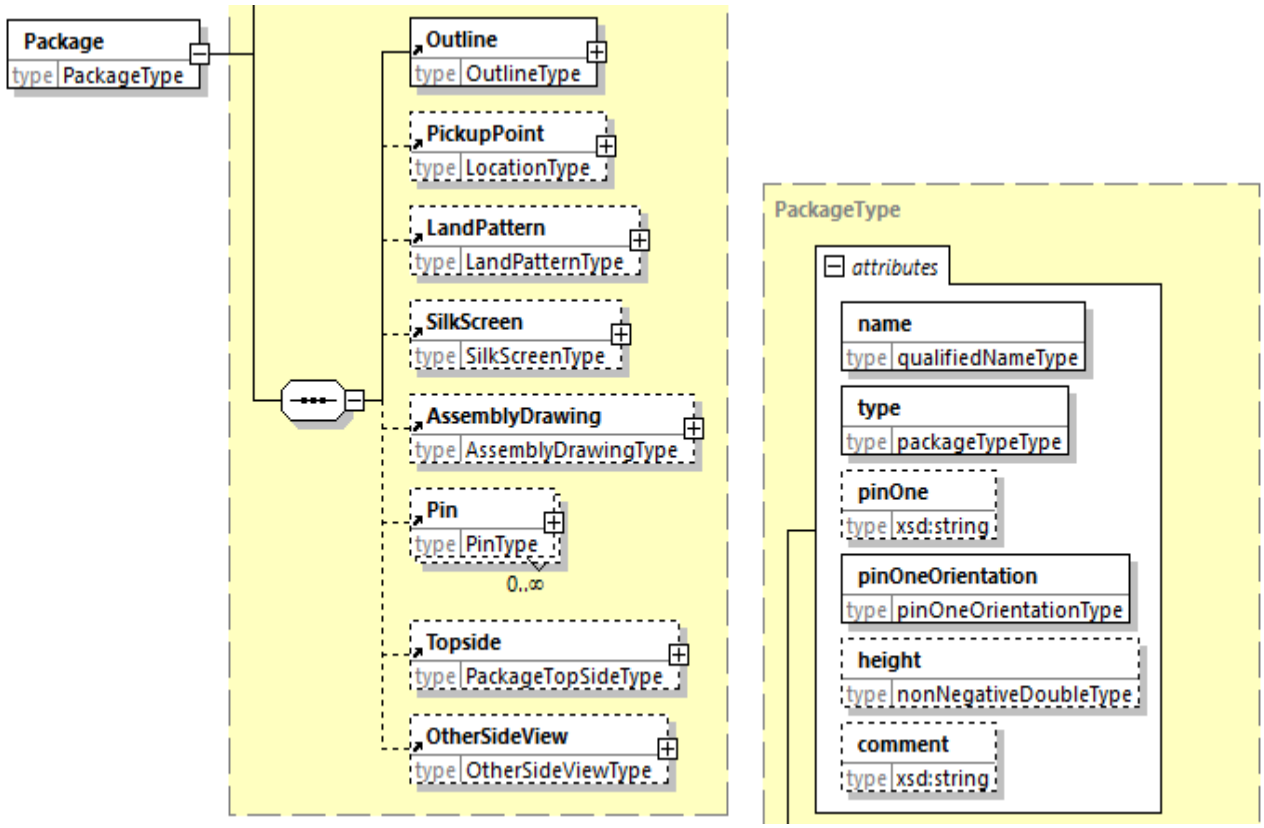
Figure 15 - Multiple Slots provided for tooling within a Panel

Requires one StepRepeat element that positions the slot on the panel at the X-Y location that becomes the point of origin or Panel Datum. The Slot is stepped three times in the X direction and twice in the Y direction. The function

may be accomplished as a step and repeat even if the features has a third dimension such as a depth of a cutout. The feature needs to be fully described as a single entity before it is repeated.

8.2.3.6 Package

The `Package` element descriptions define the package shape (`Outline`), library descriptions including land patterns, silk screen information, assembly drawing details, and pin identification. The `Package` element defines all the physical description of all the packages used by the `Component` element inside the `Step`. The names assigned to the package should be consistent with the naming convention established in IPC-7351 for parts and land pattern descriptions. (Refer to Section 2 APPLICABLE DOCUMENTS)

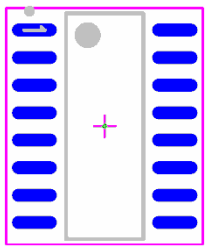
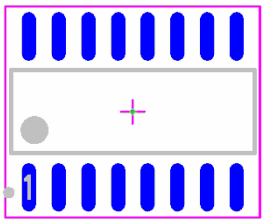
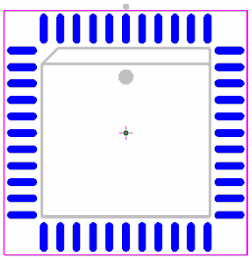
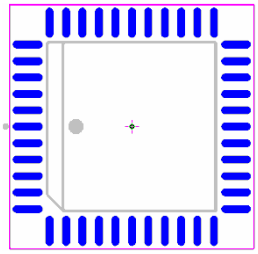
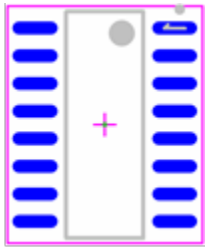
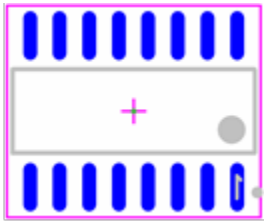
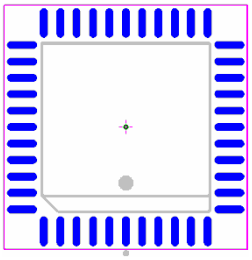
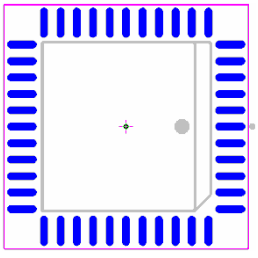
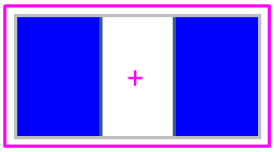
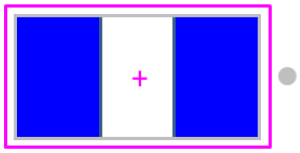


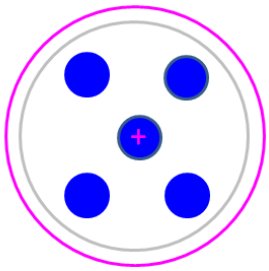
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Package	PackageType	Generic component package descriptions for use by the `Step` file schemas.	0-n
name	qualifiedNameType	A unique name assigned to the physical and graphical description of a part in accordance with the IPC-7351 Naming Convention for Packages and Land Patterns.	1-1

type	PackageTypeType	A specific body construction indicated as an enumerated string using one of the following naming conventions: AXIAL_LEADED | BARE_DIE | CERAMIC_BGA | CERAMIC_DIP | CERAMIC_FLATPACK | CERAMIC_QUAD_FLATPACK | CERAMIC_SIP | CHIP | CHIP_SCALE | CHOKE_SWITCH_SM | COIL | CONNECTOR_SM | CONNECTOR_TH | EMBEDDED | FLIPCHIP | HERMETIC_HYBRID | LEADLESS_CERAMIC_CHIP_CARRIER | MCM | MELF | FINEPITCH_BGA | MOLDED | NETWORK | PGA | PLASTIC_BGA | PLASTIC_CHIP_CARRIER | PLASTIC_DIP | PLASTIC_SIP | POWER_TRANSISTOR | RADIAL_LEADED | RECTANGULAR_QUAD_FLATPACK | RELAY_SM | RELAY_TH | SOD123 | SOIC | SOJ | SOPIC | SOT143 | SOT23 | SOT52 | SOT89 | SQUARE_QUAD_FLATPACK | SSOIC | SWITCH_TH | TANTALUM | TO_TYPE | TRANSFORMER | TRIMPOT_SM | TRIMPOT_TH | OTHER	1-1
pinOne	string	A reference to the landpattern PinRef pin number which is considered the first pin in the sequence of all pin numbers in the package. Examples are '1', 'A1', 'ANODE', '2' (when there is no pin '1')."	0-1
pinOneOrientation	pinOneOrientationtype	An enumerated string that defines the location of pinOne relative to the centroid of the package, within the library definition of the package that is captured in the design. The enumerations are LOWER_LEFT | LEFT | LEFT_CENTER | UPPER_LEFT | UPPER_CENTER | UPPER_RIGHT | RIGHT | RIGHT_CENTER | LOWER_RIGHT | LOWER_CENTER | OTHER. If OTHER is used a comment is required to describe the condition. The intent is to describe the default orientation of the package (e.g. 'landscape' or 'portrait' for an IC), which could establish a relationship to a library definition standard such as IEC or IPC, or provide a zero degree rotation reference for assembly." See the diagrams below for examples.	1-1
height	double	A description of the component height in terms of the mounting surface to the highest protrusion of the `Package`. The units are in the `Units` set by the `Cadheader`.	0-1
negativeBodyExtension	double	An optional height to which the body of the package extends below the XY plane of the landpattern and contact surface of the pins, per the package definition. That is, the extension below the surface plane of the mounting layer. A positive value defines an extension in the –z direction when the instantiated package is not mirrored, and in the +z direction when it is mirrored. The package will need to be seated in a cutout or cavity. See .Figure 16	0-1
comment	string	A description of the orientation of the Package as provided in the original CAD data	0-1
Outline	OutlineType	A nested element that defines the physical outline of the part as seen from the top, related to the graphical image that appears on the assembly. Includes body and pin profiles if applicable. The outline is an enclosed polygon type. Refer to section 3.4.7	1-1
PickupPoint	LocationType	The optimum location for an automatic assembly machine to pickup the package. This may or may not be the centroid of the package outline"	0-1
x	double	The x coordinate of the location of the PickupPoint.	1-1
y	double	The y coordinate of the location of the PickupPoint.	1-1
LandPattern	LandPatternType	A nested element that defines the surface land pattern consisting of Lands in a particular pattern that matches the footprint of the component outline. The point of origin of the `LandPattern` and `Outline` are identical.	0-1
SilkScreen	SilkScreenType	A nested element that defines the symbolization and legend required to be placed on the board for the particular package. Includes location of the reference designator or other `Text`. The point of origin of the image is the same as the origin of the `LandPattern` and `Outline`.	0-1

AssemblyDrawing	AssemblyDrawing Type	A nested element that defines the graphics required for the assembly drawing. The images relate to the component body outline and any text needed. The point of origin for the assembly drawing is the same as the images of the `Outline`, LandPattern, and `SilkScreen` schema.	0-1
Pin	PinType	A nested element that defines the pin relationship of all the pins that are a part of the package style related to the land pattern description.	0-n
Topside	PackageTopType	An optional nested element to describe pins and/or other features on the top of the `Package`, for dual side components, or components with pins only on the topside. See . Figure 16 - Package Views	0-1
OtherSideView	OtherSideViewType	An optional nested element to describe any `Outline`, `AssemblyDrawing` or `Silkscreen` elements required on the other side of the board to the mounting layer. See Figure 16 - Package Views	0-1

pinOneOrientation Examples

UPPER_LEFT	LOWER_LEFT	UPPER_CENTER	LEFT_CENTER
			
UPPER_RIGHT	**LOWER_RIGHT**	**LOWER_CENTER**	**RIGHT_CENTER**
			
LEFT	**RIGHT**		
			



Note: You may encounter a footprint similar to that shown on the left, which has only one signal pin in the center and the surrounding pins are ground pins. You could say that `pinOne` is the center pin, and assign a `pinOneOrientation` value of “OTHER” for this, with a comment value of “CENTER”. However there is then no point of reference for a rotation, and the case could have protusions or markings that make it non symmetrical. It is therefore preferential to assign `pinOne` to one of the outer pins, and set the `pinOneOrientation` from there accordingly.

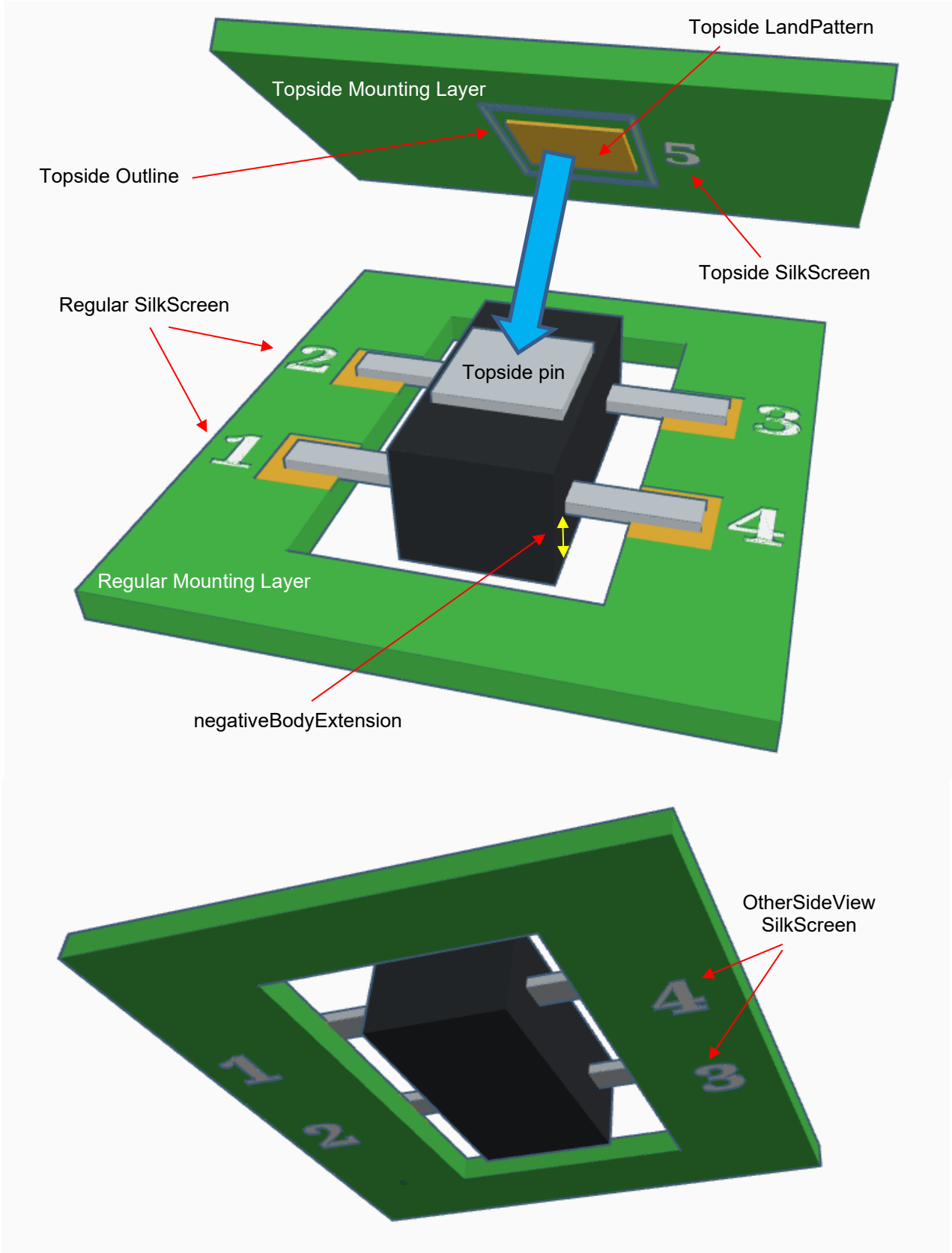
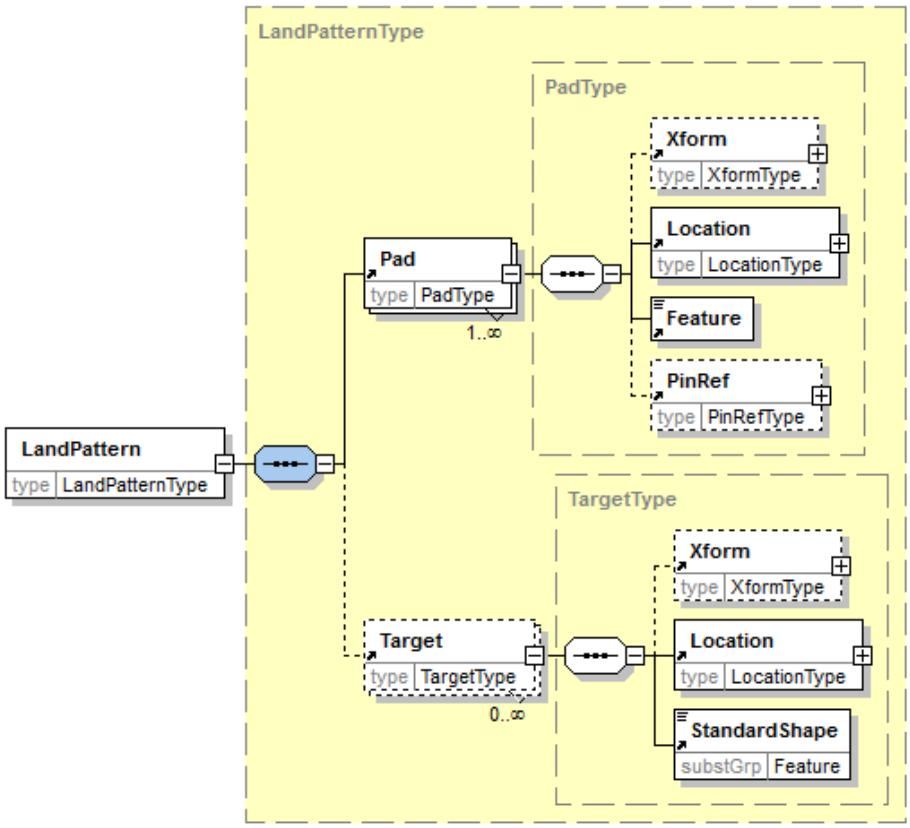


Figure 16 - Package Views

8.2.3.6.1 LandPattern

The `LandPattern` element consists of those characteristics that define the pattern to which surface mount components are attached. The embedded elements include both the `Pad` description and the potential for providing a target, usually indicating `pinOne`. Land pattern descriptions should be used wherever a relationship to component pins needs to be established. This information is redundant when layers for component attachment are defined.

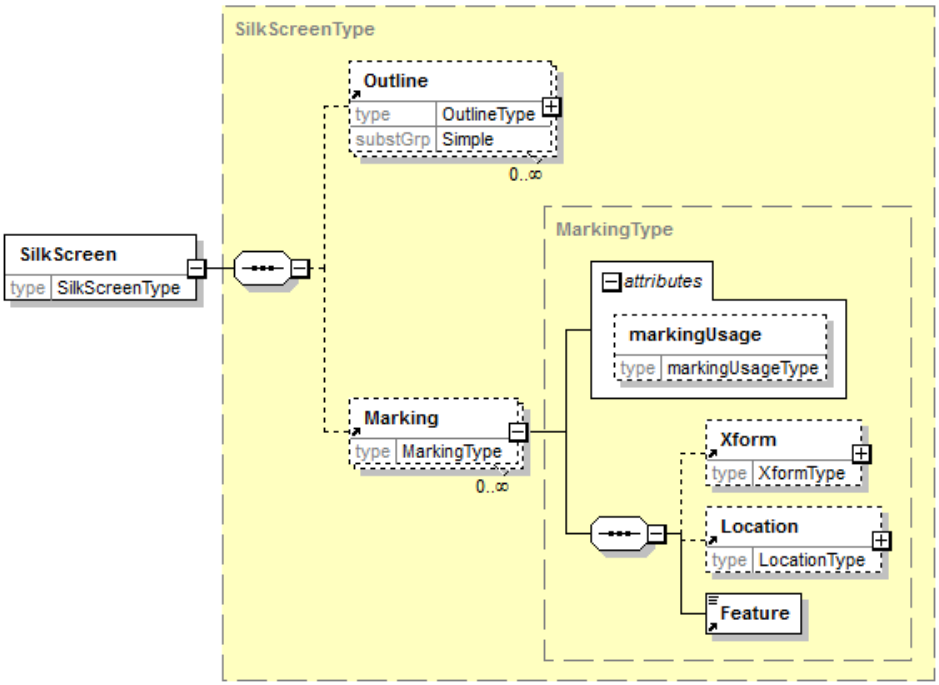


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
LandPattern	LandPatternType	A nested element that defines the surface land pattern consisting of Pads in a particular pattern that matches the footprint of the component.	0-1
Pad	PadType	A nested element defining the pad to be located as part of the land pattern.	1-n
Xform	XformType	An element that provides the ability to reset the point of origin of a predefined pad that can be scaled, mirror imaged or rotated. See 3.3	0-1
Location	LocationType	The location of the image defined by the standard shape or Usershape or a pre-defined standard shape, or Usershape of the pad. The image may have been reorientated by the Xform.	1-1
Feature	ABSTRACT	An embedded element that defines a substitution group of any `StandardShape` or `UserShape` that may be instantiated as a part of the `Pad` Description. A predefined `StandardShape or UserShape` may also be instantiated by its unique “id” when the feature is contained in the `DictionaryStandard or User`. When a reference is made to the dictionary predefined primitive the `Units` must match.	1-1
PinRef	PinRefType	An individual `Pin` related to the place where a component attaches to the net. This description facilitates location of lands on the surface of the board or places where through-hole components are mounted.	1-n
Target	TargetType	A nested element defining the target to be located as part of the land pattern.	0-n

StandardShape	ABSTRACT	A substitution group that permits the substitution of any of the `StandardPrimitive` shapes in accordance with their individual descriptions. A predefined `StandardPrimitive` may also be instantiated by its unique “id” when the feature is contained in the `DictionaryStandard`. When a reference is made to the dictionary predefined primitive the `Units` must match.	1-1

8.2.3.6.2 SilkScreen

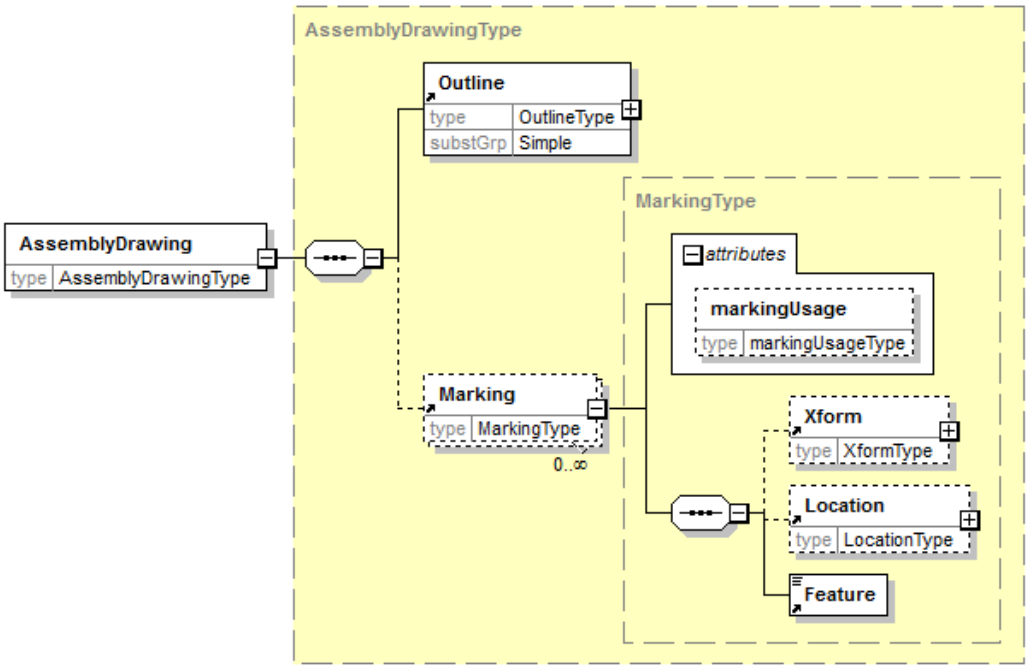
The `SilkScreen` element defines the symbolization and legend required to be placed on the board for the particular package. The `SilkScreen` descriptions include location of the reference designator or other `Text`. The point of origin of the image is the same as the origin of the `LandPattern` and `Outline`.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
SilkScreen	SilkScreenType	A nested element that defines the symbolization and legend required to be placed on the board for the particular package. Includes location of the reference designator or other `Text`. The point of origin of the image is the same as the origin of the `LandPattern` and `Outline`.	0-1
Outline	OutlineType	A nested element that defines the outlines of the part related to the graphical image that appears on the board. The outline is an enclosed polygon type. Refer to section 3.4.7	0-n
Marking	markingType	A nested element that defines the characteristics of the feature being instantiated as a part of the SilkScreen	0-n
markingUsage	markingUsageType	An indication as to the usage of any marking that becomes a part of the SilkScreen. The descriptions are enumerated strings and must be one of the following: REFDES | PARTNAME | TARGET | POLARITY_MARKING | ATTRIBUTE_GRAPHICS | PIN_ONE | NONE>	1-1
Xform	XformType	An element that provides the ability to reset the point of origin of the marking, then scale, mirror image or rotate the marking feature after it has been placed at an X and Y location.	0-1
Location	LocationType	The location of the image defined by the feature or a pre-defined feature. The image may have been reorientated by the Xform.	1-1
Feature	ABSTRACT	A substitution group that permits the substitution of any of the `StandardShape`, `StandardPrimitive`, or `UserPrimitive` shape in accordance with their individual descriptions. A predefined `StandardPrimitive`, or `UserPrimitive` may also be instantiated by its unique “id” when the feature is contained in the DictionaryStandard or DictionaryUser. When a reference is made to either of the dictionaries the predefined primitive Units must match with the Units of the file.	0-n

8.2.3.6.3 AssemblyDrawing

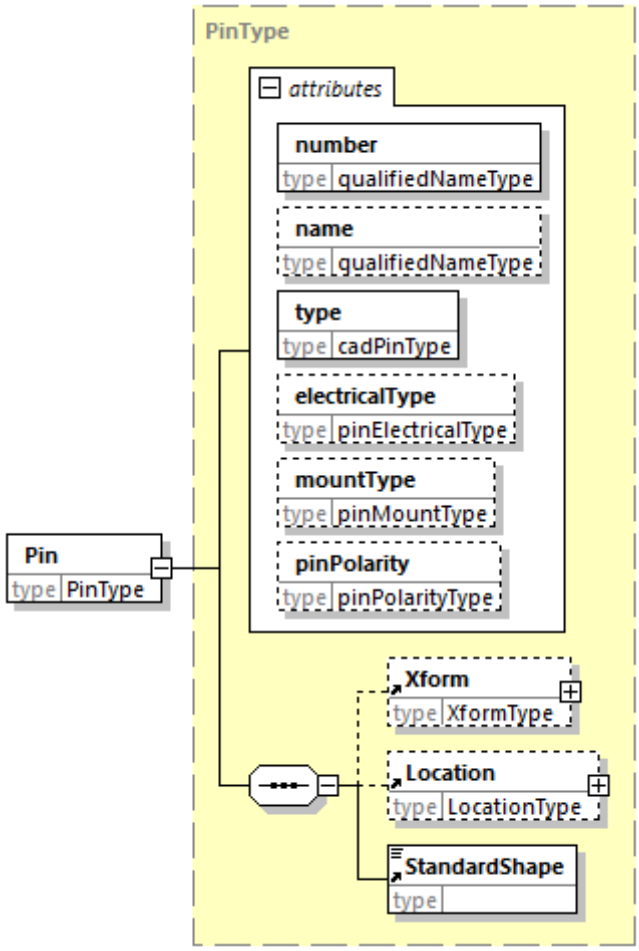
The `AssemblyDrawing` element reuses the same embedded elements and attributes as defined for the `Silkscreen` characteristics. The construction schemas are repeated to aid the reader in interpretation of the library structure.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
AssemblyDrawing	AssemblyDrawing Type	A nested element that defines the graphics required for the assembly drawing. The images relate to the component body outline and any text needed. The point of origin for the assembly drawing is the same as the images of the outline, land pattern, and silk screen schema.	0-1
Outline	OutlineType	A nested element that defines the outlines of the part related to the graphical image that appears on the board. The `Outline` is an enclosed polygon type.	0-1
Marking	markingType	A nested element that defines the characteristics of the feature being instantiated as a part of the AssemblyDrawing	0-n
markingUsage	markingUsageType	An indication as to the usage of any marking that becomes a part of the AssemblyDrawing. The descriptions are enumerated strings and must be one of the following: REFDES | PARTNAME | TARGET | POLARITY_MARKING | ATTRIBUTE_GRAPHICS | PIN_ONE | NONE.	1-1
Xform	XformType	An element that provides the ability to reset the point of origin of the marking, then scale, mirror image or rotate the marking feature after it has been placed at an X and Y location.	0-1
Location	LocationType	The location of the image defined by the feature or a pre-defined feature. The image may have been reorientated by the Xform.	1-1
Feature	ABSTRACT	A substitution group that permits the substitution of any of the `StandardShape, StandardPrimitive, or UserPrimitive` shape in accordance with their individual descriptions. A predefined `StandardPrimitive, or UserPrimitive` may also be instantiated by its unique “id” when the feature is contained in the DictionaryStandard or DictionaryUser. When a reference is made to either of the dictionaries the predefined primitive Units must match with the Units of the file.	1-1

8.2.3.6.4 Pin

The `Pin` element represents a set of `Pin` characteristics that are attached to each component package. Each `Pin` has a `number`, `name`, type, `electricalType` and `mountType`. Each `Pin` also contains its relative location and outline.

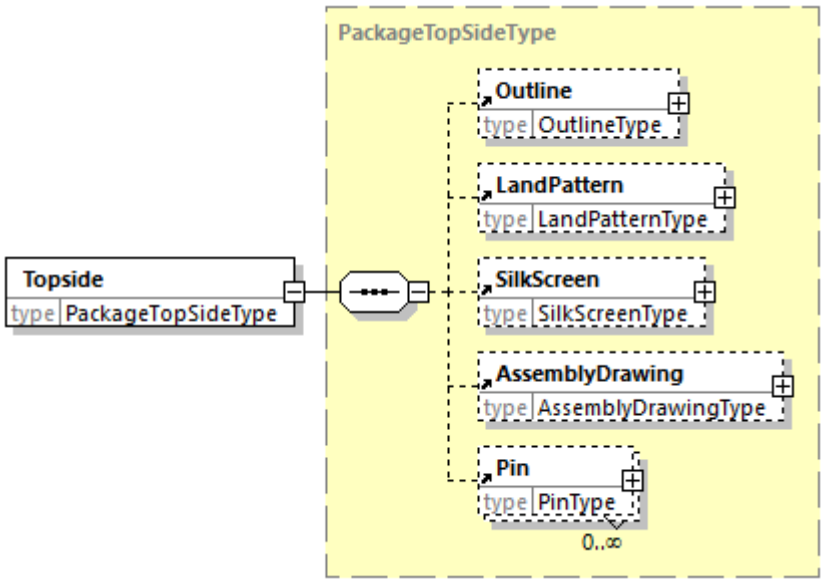


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Pin	PinType	A nested element that defines the pin relationship of all the pins that are a part of the package style related to the land pattern description.	0-n
number	qualifiedNameType	An alphanumeric indicator identified as the pin number which is unique within the package description and is established by the netlist	1-1
name	qualifiedNameType	A name assigned by the user to describe the Pin at a particular location. The same name may be applied to multiple pins at the users discretion	0-1
type	cadPinType	An enumerated string that defines the type of Pin as being one of the following: THRU | BLIND | SURFACE.	1-1
electricalType	pinElectricalType	The electrical type enumerated string that defines the Pin as one of three possible conditions. These are: ELECTRICAL | MECHANICAL | UNDEFINED.	0-1
mountType	pinMountType	An enumerated string that defines the mounting characteristics of the `Pin` and may be any one of the following: SURFACE_MOUNT_PIN | SURFACE_MOUNT_PAD | THROUGH_HOLE_PIN | THROUGH_HOLE_HOLE | PRESSFIT | NONBOARD | HOLE | WIRE_BOND | UNDEFINED	0-1

pinPolarity	polarityType	An enumerated string that defines the polarity of the pin. Possible values are: PLUS | MINUS | ANODE | CATHODE For example PLUS | MINUS for a polarized capacitor or ANODE | CATHODE for a diode	0-1
Xform	XformType	An element that provides the ability to reset the point of origin of the graphic outline pin shape, then scale, mirror image or rotate the shape it has been placed at an X and Y location. See 3.3	0-1
Location	LocationType	The location of the image defined by the pin shape or a pre-defined standard shape of the Pin relative to the component origin The image may have been reorientated by the Xform. The location is optional element allow the `Pin` section to be populated when the geometry and location of the pin is not known, but it is required to include the (per) Pin attributes *type, electricalType*, and *mountType* for Assembly purposes.	0-1
StandardShape	ABSTRACT	A substitution group that permits the substitution of any of the `StandardPrimitive` shapes in accordance with their individual descriptions. A predefined `StandardPrimitive` may also be instantiated by its unique "id" when the feature is contained in the DictionaryStandard. When a reference is made to the dictionary predefined primitive the `Units` must match. The intent of the standard shape is to represent the contact area or "footprint" of the Pin, as conducted to its corresponding land pattern.	1-1

8.2.3.6.5 Topside

An optional nested element to define any `Pin` and associated `Landpattern` on the topside of the package. Optional `Outline`, `Silkscreen`, and `AssemblyDrawing` child elements can be included if required on the surface of the flex or other non-embedded layer to which the Topside pins are making contact. The topside child elements are defined at a height set by the `height` attribute of the `Package`. The contact surfaces of the pins are facing up. **Note** that the `Outline`, `Landpattern`, `Silkscreen`, and `AssemblyDrawing` objects need to be defined in the mirror state in the package definition, because they will be placed on the underside of the layer above (the same as they woud if they were being defined for the bottom layer of the board).

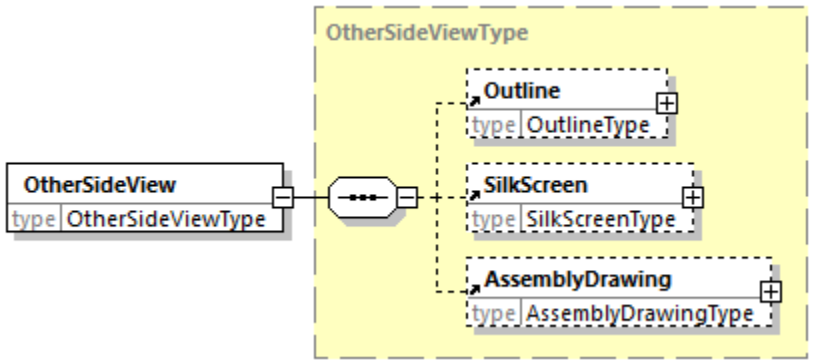


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Topside	PackageTopType	An optional netsed element to define pins and other feaures on the top of the package, at a height set by the `height` attribute of the parent `Package`.	0-n
Outline	OutlineType	A nested element that defines the physical outline of the part as seen from the top, related to the graphical image that appears on the assembly. Includes body and pin profiles if applicable. The outline is an enclosed polygon type. Refer to section 3.4.7	0-1

LandPattern	LandPatternType	A nested element that defines the surface land pattern consisting of Lands in a particular pattern that matches the footprint of the component outline. The point of origin of the `LandPattern` and `Outline` are identical.	0-1
SilkScreen	SilkScreenType	A nested element that defines the symbolization and legend required to be placed on the board for the particular package. Includes location of the reference designator or other `Text`. The point of origin of the image is the same as the origin of the `LandPattern` and `Outline`.	0-1
AssemblyDrawing	AssemblyDrawing Type	A nested element that defines the graphics required for the assembly drawing. The images relate to the component body outline and any text needed. The point of origin for the assembly drawing is the same as the images of the `Outline`, LandPattern, and `SilkScreen` schema.	0-1
Pin	PinType	A nested element that defines the pin relationship of all the pins that are a part of the package style related to the land pattern description.	0-n

8.2.3.6.6 OtherSideView

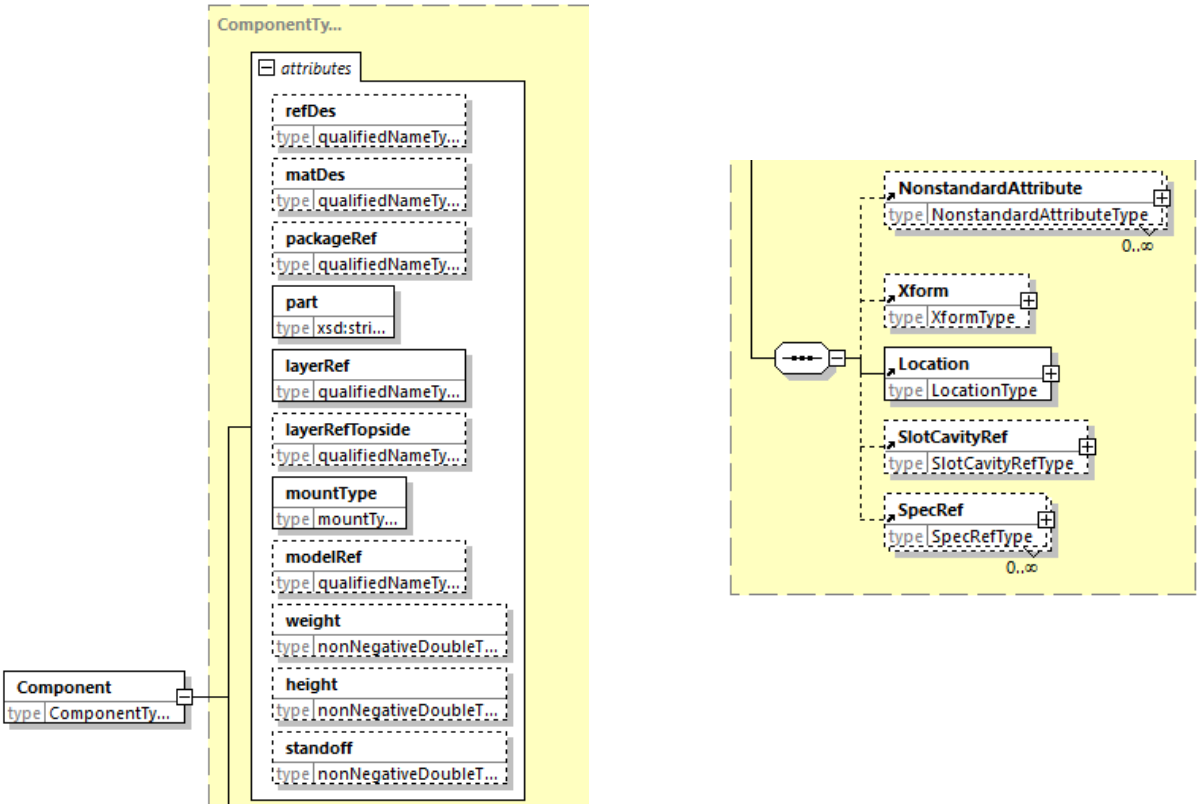
An optional nested element to define any `Outline`,`Silkscreen`, or `AssemblyDrawing` objects required on the other side of the board to where the package is mounted. Refer to Figure 16 - Package Views.
Note that all objects need to be defined in the mirror state in the package definition.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
OtherSideView	OtherSideViewType	An optional nested element to describe any `Outline`, `AssemblyDrawing` or `Silkscreen` elements required on the other side of the board to the mounting layer. Required in cases such as a component mounted on the top side of the board that protrudes through a cutout to the bottom side of the board	0-1
Outline	OutlineType	A nested element that defines the physical outline of the part as seen from the top, related to the graphical image that appears on the assembly. Includes body and pin profiles if applicable. The outline is an enclosed polygon type. Refer to section 3.4.7	0-1
SilkScreen	SilkScreenType	A nested element that defines the symbolization and legend required to be placed on the board for the particular package. Includes location of the reference designator or other `Text`. The point of origin of the image is the same as the origin of the `LandPattern` and `Outline`.	0-1
AssemblyDrawing	AssemblyDrawing Type	A nested element that defines the graphics required for the assembly drawing. The images relate to the component body outline and any text needed. The point of origin for the assembly drawing is the same as the images of the `Outline`, LandPattern, and `SilkScreen` schema.	0-1

8.2.3.7 Component

The `Component` section contains all the `Component` elements that were read from the originating CAD system and were captured in the `Component` element descriptions.

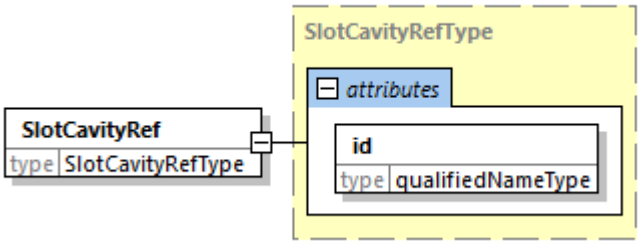


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Component	ComponentType	A nested element list of component descriptions and their application to the electronic product.	0-n
refDes	qualifiedNameType	A unique name assigned to the particular electrical or mechanical component that is defined in the `Bom` section	0-1
matDes	qualifiedNameType	A unique name assigned to the particular material that requires a precise location (e.g. coin or adhesive), that is defined in the `Bom` section	0-1
packageRef	qualifiedNameType	A reference to the package Type used to house the component.	0-1
part	string	A unique identifier of the part, such as `OEMDesignNumber`	1-1
layerRef	qualifiedNameType	The reference to a specific layer element, by its “name” attribute. The layer referenced is to where the component is primarily mounted. The reference is usually a surface layer, however it may be an internal layer for embedded component attachment. The layer Function Type shall be of the Board usage only.	1-1
layerRefTopside	qualifiedNameType	The reference to a specific layer above an embedded component that any Topside pins connect to. Refer to section 8.2.3.6.5	0-1
mountType	mountType	The mount type as defined by an enumerated string which may be one of the following: SMT | THMT | EMBEDDED | PRESSFIT | WIRE_BONDED | GLUED | CLAMPED | SOCKETED | FORMED | OTHER. This attribute can be used to modify the `Package` description i.e., a through-hole mount modified to be surface mounted. If at least one pin needs to be flow soldered then the component should be considered SMT, so that the pick and place machine can account for it, even though the majority of the other pins are not SMT.	1-1

modelRef	qualifiedNameType	Reference to a unique 3D model name that represents the component. Model examples include a simplified model of an IC, or a precise model of a copper coin	0-1
weight	nonNegativeDouble Type	The weight of the particular component in grams.	0-1
height	nonNegativeDouble Type	The height that the top protrusion of the component body is above the surface of the printed board in units assigned in the `CadHeader`.	0-1
standoff	nonNegativeDouble Type	The standoff clearance between the body and the printed board in units assigned in the `CadHeader`.	0-1
NonstandardAttribute	NonstandardAttribute Type	A nested element that can be used to provide extra details regarding a particular "component" description in terms of BOOLEAN, DOUBLE, INTEGER, or STRING data. The properties of the NonstandardAttribute become unique due to its' Name.	0-n
Xform	XformType	An element that provides the ability to reset the point of origin of a predefined area that can be scaled, mirror imaged or rotated. The image may also be located by the Xform. See 3.3.	0-1
Location	LocationType	The location of the component defined by the `packageRef` or a pre-defined standard shape of the `Package`. The image may have been reorientated by the Xform.	1-1
SlotCavityRef	SlotCavityRefType	A reference to a unique name of a `SlotCavity` definition, that has properties and attributes that contribute to the placement of the component. The existence of of `SlotCavityRef` means that the component is either totally embedded in the PCB or is located at the floor of the an intrusion into the PCB surface	0-1
SpecRef	SpecRefType	A reference to a unique `Spec` that pertains to the component instance, such as assembly instructions or mounting requirements.	0-n

8.2.3.7.1 SlotCavityRef

A child element of `Component` to reference a unique ID of the `SlotCavity` in which the component is mounted. The location of the referenced `SlotCavity` must be related to the location of the `Component` such that the extents of the component completely, or partially, fit within the feature describing the outline of the slot or cavity.



id	qualifiedNameType	The unique name of a SlotCavity definition	1-1

8.2.3.7.2 Embedded Discrete Components

The following describes examples of different placement scenarios for discrete embedded components

8.2.3.7.2.1 Face Down, Not Mirrored

Component IC1 is positioned onto layer L2 and attached by an adhesive. Prepreg layer(s) D2 with a pre-drilled cut-out, is placed to form a cavity around IC1. Prepreg D1 is placed on top, which fills in the remainder of the cavity when pressed. The above stackup could form a laminate in the overall stackup, and be described like this:

```
<StackupGroup name="laminate1" thickness="0.2638" tolPlus="0.0" tolMinus="0.0">
        <StackupLayer layerOrGroupRef="L1" thickness="0.0559">
                </StackupLayer>
        <StackupLayer layerOrGroupRef="D1_2" thickness="0.152">
                </StackupLayer>
        <StackupLayer layerOrGroupRef ="L2" thickness="0.0559">
                </StackupLayer>
</StackupGroup>
<StackupGroup name="inner_prepreg" thickness="0.152" tolPlus="0.0" tolMinus="0.0" >
        <CADDataLayerRef layerRef="D1_2"/>
        <StackupLayer layerOrGroupRef="D1" thickness="0.1015">
                </StackupLayer>
        <StackupLayer layerOrGroupRef="D2" thickness="0.0505">
                </StackupLayer>
</StackupGroup>
```

The component definition could be described:

```
<Component refDes="IC1" packageRef="BGA123" layerRef="L2" part="TA46898-0107F1"
        mountType="EMBEDDED" standoff="0.0" height="1.1000">
    <Xform rotation="0.0" mirror="false"/>
    <Location x="114.9500" y="37.3750"/>
    <SlotCavityRef id="D2_CUT"/>
</Component>
```

The routing layer for the cavity, say Route_D2, would be defined with a span across only dielectric layer D2:

```
<Layer name="Route_D2" layerFunction="ROUT" side="NONE" polarity="POSITIVE">
        <Span fromLayer="D2" toLayer="D2"/>
</Layer>
```

And then the SolotCavity itself would be defined as a LayerFeature->Set of the route layer

```
<SlotCavity name="D2_CUT" platingStatus="NONPLATED" plusTol="0.03" minusTol="0.03">
        <Location x="114.9500" y="37.3750"/>
        <Circle diameter="5.0"/>
</SlotCavity>
```

The physical net of a pin could be described:

```
<PhyNet name="IC1_A1">
    <PhyNetPoint x="76.5000" y="109.2500" layerRef="L2" netNode="MIDDLE" exposure="COVERED"
                                                                                via="false">
     <StandardPrimitiveRef id="CIRCLE_1"/>
```

```
</PhyNetPoint>
```

8.2.3.7.2.2 Face Down, Mirrored

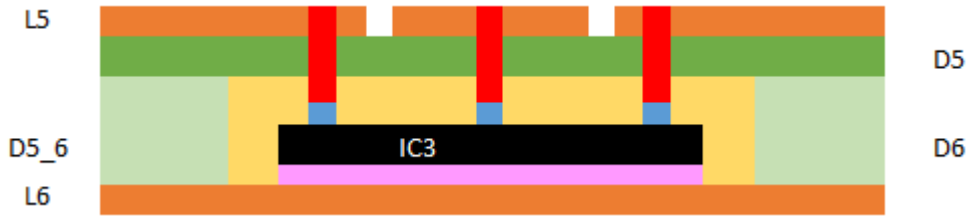
Formed the same way as for IC1 then the whole laminate is inverted. The stackup has the same structure as IC1. The cavity in D3 is formed the same way as for D2 above.
The component definition could be described:

```
<Component refDes="IC2" packageRef="BGA123" layerRef="L3" part="TA46898-0107F1"
            mountType="EMBEDDED" standoff="0.0" height="1.1000">
      <Xform rotation="0.0" mirror="true"/>
      <Location x="114.9500" y="37.3750"/>
      <SlotCavityRef id="D3_CUT"/>
</Component>
```

And a physical net of a pin could be described:

```
<PhyNet name="IC2_C3">
        <PhyNetPoint x="76.5000" y="109.2500" layerRef="L3" netNode="MIDDLE" exposure="COVERED"
via="false">
          <StandardPrimitiveRef id="CIRCLE_1"/>
          </PhyNetPoint>
```

8.2.3.7.2.3 Face Up, Not Mirrored



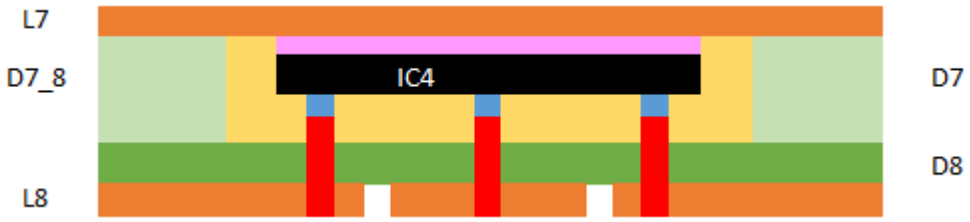
Component IC3 is turned face up and adhered to layer L6, positioned accurately in the X-Y plane. Prepreg layer(s) D6 with a pre-drilled hole, is placed to form a cavity around IC3. The cavity is closed with filler material and another prepreg layer D5 is placed on top. A laser drill, with a frequency that cuts through dielectric but not copper, is used to create holes to the pins of IC3, which are plated to form micro vias to the pins. The exact depth (z axis dimension) of the pins is not required, and may be diffcult to know anyway once the laminate is pressed. Only the precise x & y location of the embedded pins are required.
The stackup has the same structure as IC1. The cavity in D6 is formed the same way as for D2 above.
The component definition could be described:

```
<Component refDes="IC3" packageRef="BGA123" layerRef="L6" part="TA46898-0107F1" mountType="EMBEDDED"
standoff="0.0" height="1.1000">
          <Xform rotation="0.0" mirror="false" faceUp="true"/>
          <Location x="114.9500" y="37.3750"/>
          <SlotCavityRef id="D6_CUT"/>
</Component>
```

And a physical net of a pin could be described as follows. As I say the exact z dimension is not required, so just saying what prepreg layer it lies within is sufficient.

```
<PhyNet name="IC3_4">
        <PhyNetPoint x="76.5000" y="109.2500" layerRef="D5_6" netNode="MIDDLE"
                     exposure="COVERED" via="false">
                <StandardPrimitiveRef id="CIRCLE_1"/>
        </PhyNetPoint>
```

8.2.3.7.2.4 Face Up, Mirrored



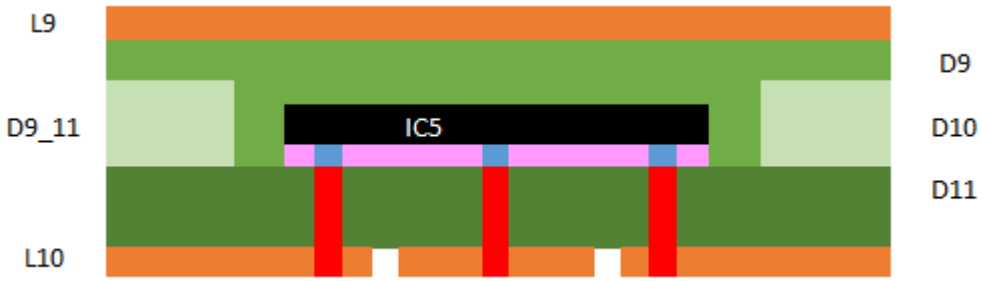
Formed the same way as for IC3 then the whole laminate is inverted. The stackup has the same structure as IC1. The cavity in D7 is formed the same way as for D2 above.
The component definition could be described:

```
<Component refDes="IC4" packageRef="BGA123" layerRef="L7" part="TA46898-0107F1"
          mountType="EMBEDDED" standoff="0.0" height="1.1000">
        <Xform rotation="0.0" mirror="true" faceUp="true"/>
        <Location x="114.9500" y="37.3750"/>
        <SlotCavityRef id="D6_CUT"/>
</Component>
```

And a physical net of a pin could be described:

```
<PhyNet name="IC4_B7">
        <PhyNetPoint x="76.5000" y="109.2500" layerRef="D7_8" netNode="MIDDLE"
             exposure="COVERED" via="false">
         <StandardPrimitiveRef id="CIRCLE_1"/>
        </PhyNetPoint>
```

8.2.3.7.2.5 Placed On Dielectric Base



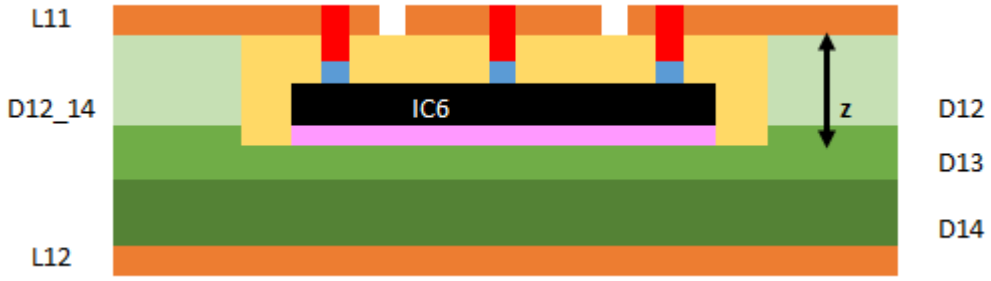
In the case where the component is not placed on a copper layer, so it ‘floats’ in the dielectric, but it is placed on a prepreg ‘base’ layer then other prepreg layers built up around it. The cavity in D10 is formed the same way as for D2 above.
The component definition could be described:

```
<Component refDes="IC5" packageRef="BGA123" layerRef="D11" part="TA46898-0107F1"
          mountType="EMBEDDED" standoff="0.0" height="1.1000">
        <Xform rotation="0.0" mirror="false" faceUp="false"/>
        <Location x="114.9500" y="37.3750"/>
        <SlotCavityRef id="D10_CUT"/>
 </Component>
```

And a physical net of a pin could be described:

```
<PhyNet name="IC5_64">
        <PhyNetPoint x="76.5000" y="109.2500" layerRef="D9_11" netNode="MIDDLE"
            exposure="COVERED" via="false">
         <StandardPrimitiveRef id="CIRCLE_1"/>
        </PhyNetPoint>
```

8.2.3.7.2.6 Placed in Cavity Drilled To Given Depth



A cavity is drilled to a given depth, z, in a pre-formed dielectric laminate D12_14. Once D12 thru D14 are pressed together, and given the tolerance of the depth, it is not known for certain in which dielectric layer (D12 thru D14) the cavity floor lies. So the component is said to be placed on the laminate layer D12_14. This, together with the cavity definition in a coincident location, as referenced by SlotCavityRef, defines the component placement in all 3 dimensions.

```
<Component refDes="IC6" packageRef="BGA123" layerRef="D12_14" part="TA46898-0107F1"
            mountType="EMBEDDED" standoff="0.0" height="1.1000">
      <Xform rotation="0.0" mirror="false" faceUp="true"/>
      <Location x="114.9500" y="37.3750"/>
   <SlotCavityRef id="D12_14_CUT"/>
</Component>
```

In this case the routing layer for the cavity, say Route_D2, would NOT be defined with a span:

```
        <Layer name="Route_D2" layerFunction="ROUT" side="NONE" polarity="POSITIVE">
        </Layer>
```

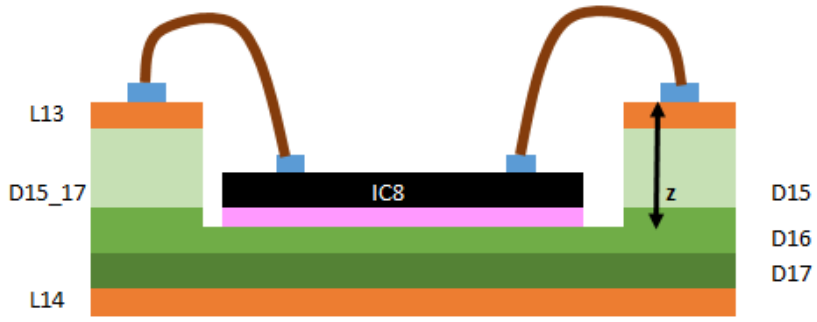
Instead the SlotCavity, defined as a LayerFeature->Set of the route layer, would contain a MaterialCut child element, defining the properties of the cut

```
        <SlotCavity name="D12_14_CUT" platingStatus="NONPLATED" plusTol="0.03"
                                        minusTol="0.03">
                <Location x="114.9500" y="37.3750"/>
                <RectCenter width="2.3" height="1.5"/>
                <MaterialCut depth="0.15" startCutLayer="D12" direction="DOWN"
                        plusTol="0.02" minusTol="0.02"/>
        </SlotCavity>
```

The physical net of a pin could be described:

```
<PhyNet name="IC6_A1">
        <PhyNetPoint x="76.5000" y="109.2500" layerRef="D12_14" netNode="MIDDLE"
                        exposure="COVERED" via="false">
          <StandardPrimitiveRef id="CIRCLE_1"/>
        </PhyNetPoint>
```

8.2.3.7.3 Wire Bonded Components



The component IC8 is placed face up in a cavity drilled to a given depth z from the top surface. The mountType is WIRE_BONDED.

```
<Component refDes="IC8" packageRef="BGA123" layerRef="D15_17" part="DEF"
            mountType="WIRE_BONDED" height="1.1000">
      <Xform rotation="0.0" mirror="false" faceUp="true"/>
      <Location x="114.9500" y="37.3750"/>
      <SlotCavityRef id="D15_17_CUT"/>
</Component>
```

The SlotCavityRef in the Component Definition and the cut from the TOP layer to depth z at the coincident location to ensure the component is placed correctly in the cavity.

```
<SlotCavity name="D12_14_CUT" platingStatus="NONPLATED" plusTol="0.03" minusTol="0.03">
        <Location x="114.9500" y="37.3750"/>
        <RectCenter width="2.3" height="1.5"/>
        <MaterialCut depth="0.36" startCutLayer="TOP" direction="DOWN" plusTol="0.01" minusTol="0.01"/>
</SlotCavity>
```

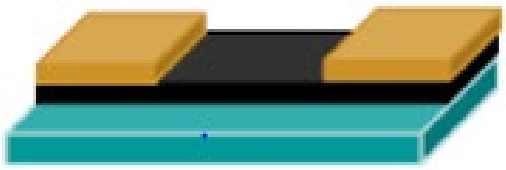
The physical net of a pin could be described:

```
<PhyNet name="IC8_24">
      <PhyNetPoint x="76.5000" y="109.2500" layerRef="D15_17" netNode="MIDDLE"
                   exposure="COVERED" via="false">
        <StandardPrimitiveRef id="CIRCLE_1"/>
      </PhyNetPoint>
```

8.2.3.7.4 Formed Components

The following are examples of how formed components, created utilizing printing (thick-film) or etching (thin film) processes, can be described in IPC-2581.

8.2.3.7.4.1 Etched Components



The above image represents a resistor created by sequential etching of a thin-film laminate of resistive material (black) and copper foil (orange), on a dielectric base (blue). The resistive material would be defined as a layer with layerFunction COMPONENT_FORMED:

```
<Layer name="F1" layerFunction="COMPONENT_FORMED" side="NONE" polarity="POSITIVE"/>
```

The thin-film laminate could be represented as a `StackupGroup`, which has a `SpecRef` referencing the laminate material:

```
<StackupGroup name="thin-film-lam" thickness="0.2079" tolPlus="5" tolMinus="5" tolPercent="true">
    <StackupLayer layerOrGroupRef="L2" thickness="0.0559"/
    <StackupLayer layerOrGroupRef="F1" thickness="0.152"/>
    <SpecRef id="laminate_material"/>
</StackupGroup>
```

Or the resistive material could simply be a standalone layer defined below the copper layer in the CAD tool.

The pin pads of the component would be defined within a library package in the CAD tool, so that when placed the pads would become etched features on the copper layer, L2 in our example. The component instantiation would have a mountType of "FORMED":

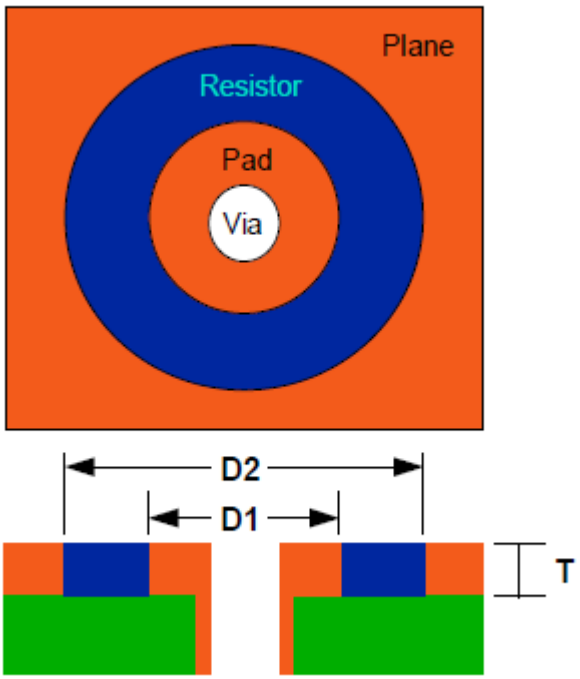
```
<Component refDes="CF1" packageRef="formed1" layerRef="L2" part="XZY" mountType="FORMED">
    <Location x="32.76" y="134.00"/>
</Component>
```

The resistive element would be a rectangular layer feature in this example (but in reality could be a serpentined or other complex shape) etched onto the resistive layer, with a componentRef referencing the etched component, CF1. Its location would be that same as that of the component origin.

```
<LayerFeature layerRef="F1">
    <Set componentRef="CF1">
        <Features>
            <RectCenter width = "1.2" height = "0.6"/>
            <Location x="32.76" y="134.00"/>
        </Features>
    </Set>
</LayerFeature>
```

If possible the resistive layer feature could be added to the component library definition, so feature and pads could be moved as one.

8.2.3.7.4.2 Printed Components



The above image represents an annular resistor formed by (screen) printing a resistive material (blue) over a copper plane (orange). There is a dielectric layer (green) below the plane. A donut shaped cutout is etched into the plane, and the exposed dielectric is then printed over with the resistive material. A via is drilled through the central pad.

The resistive material would be defined as a layer with layerFunction COMPONENT_FORMED:

```
<Layer name="F2" layerFunction="COMPONENT_FORMED" side="NONE" polarity="POSITIVE"/>
```

This layer would be defined above the plane layer in the stackup:

```
<StackupLayer layerOrGroupRef="F2" thickness="0.001" tolPlus="10" tolMinus="10" tolPercent="true">
    <SpecRef id="Print_Material "/>
</StackupLayer>
<StackupLayer layerOrGroupRef="PLANE" thickness="0.0508" tolPlus="10" tolMinus="10" tolPercent="true">
    <SpecRef id="SPEC_LAYER_TOP"/>
</StackupLayer>
<StackupLayer layerOrGroupRef="DIEL1" thickness="0.152" tolPlus="10" tolMinus="10" tolPercent="true">
    <SpecRef id="SPEC_LAYER_DIEL1"/>
</StackupLayer>
```

A circular cut-out would be made in the plane:

```
<LayerFeature layerRef="PLANE">
    <Set>
        <Features>
            <Contour>
                # Polygon defining the outline of the plane goes here #
                <Cutout>
                    <Polybegin x="5" y="3">
                    <PolyStepCurve x="1" y="3" centerX="3" centerY="3" clockwise="false"/>
                    <PolyStepCurve x="5" y="3" centerX="3" centerY="3" clockwise="false"/>
                </CutOut>
            </Contour>
        </Features>
    </Set>
</LayerFeature>
```

The component would be placed in the same location as the cutout. Its package would contain a central pad that would be placed in the center of the cutout, and a concentric annular ring pad that would be placed outside the cutout and so merge with surrounding plane. The component instantiation would have a mountType of "FORMED":

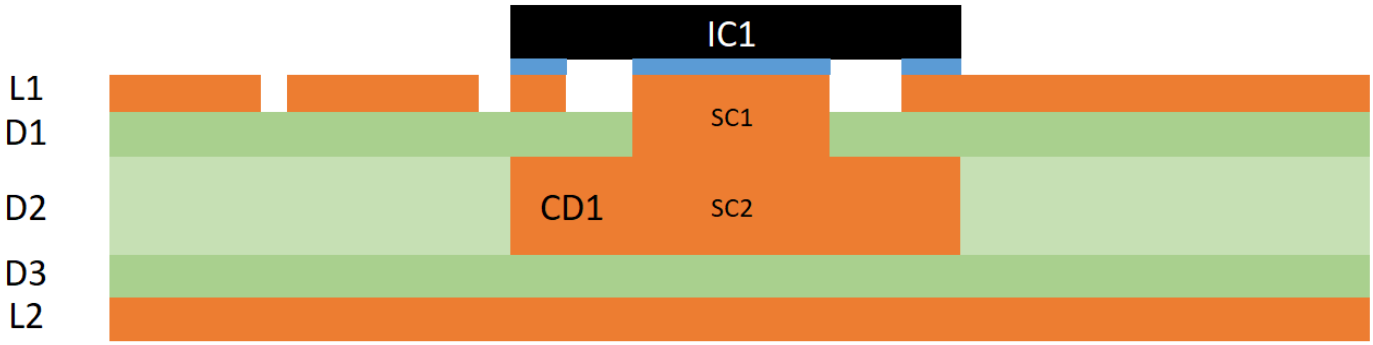
```
<Component refDes="CF2" packageRef="formed2" layerRef="L2" part="ABC" mountType="FORMED">
    <Location x="3" y="3"/>
</Component>
```

The resistive material layer (above the plane layer) would contain a donut shape in the same location as the component, which would reference the component:

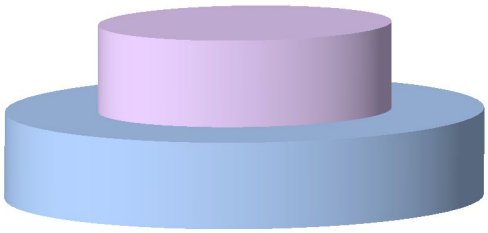
```
<LayerFeature layerRef="F2">
    <Set componentRef="CF2">
        <Features>
            <Donut shape = "ROUND" outerDiameter = "4" innerDiameter = "2"/>
            <Location x="3" y="3"/>
        </Features>
    </Set>
</LayerFeature>
```

If possible the resistive layer feature could be added to the component library definition, so feature and pads could be moved as one.

8.2.3.7.5 Coins



Consider the above example cross section of a thermal relief coin, refDes CD1, in a PCB substrate. The coin is a 'hat' shape, made up of 2 concentric cylinders, as shown:



Two `SlotCavity` features, SC1 & SC2, would be required to create the overall cavity for the coin to fit into. Each of these would require their own routing layer with a unique layer span. The tweo features would have the same location, and could reference the coin:

Layer definition of Route_SC2:

```
<Layer name="Route_ SC2" layerFunction="ROUT" side="NONE" polarity="POSITIVE">
        <Span fromLayer="D2" toLayer="D2"/>
</Layer>
```

Layer Feature defined on layer Route_ SC2:

```
<LayerFeature layerRef="Route_ SC2">
    <Set componentRef="CD1">
        <SlotCavity name="SC2" platingStatus="NONPLATED" plusTol="0.05" minusTol="0.05">
            <Location x="15.894" y ="18.713"/>
            <Circle diameter="40"/>
        </ SlotCavity >
    </Set>
</LayerFeature>
```

Layer definition of Route_SC1:

```
<Layer name="Route_SC1" layerFunction="ROUT" side="NONE" polarity="POSITIVE">
        <Span fromLayer="L1" toLayer="D1"/>
</Layer>
```

Layer Feature defined on layer Route_ SC2:

```
<LayerFeature layerRef="Route_ SC2">
    <Set componentRef="CD1">
        <SlotCavity name="SC2" platingStatus="NONPLATED" plusTol="0.05" minusTol="0.05">
            <Location x="15.894" y ="18.713"/>
            <Circle diameter="20"/>
        </ SlotCavity >
    </Set>
</LayerFeature>
```

A `Model` could be defined to describe the shape and dimensions of the coin:

```
<Model name=”Coin”>
  <SpecRef id=”CoinMaterial”/>
  <Extrusion startingHeight=”0’ height=”5”>
      <Circle diameter="40"/>
      <Location x=”0.00” y=”0.00”/>
  </Extrusion>
  <Extrusion startingHeight=”5’ height=”3”>
      <Circle diameter="20"/>
      <Location x=”0.00” y=”0.00”/>
  </Extrusion>
</Model>
```

Then the `Component` could reference both the `Model` and the `SlotCavity`. It could be glued to layer D3, and would share the same location as both `SlotCavity` features. Only one needs to be referenced.

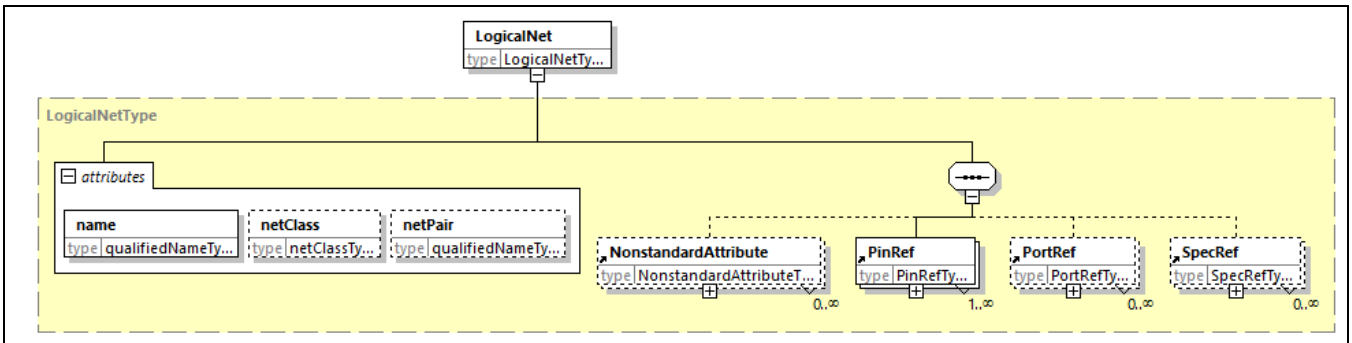
```
<Component refDes="CD1" layerRef="D3" part="Order#3521" mountType="GLUED" modelRef=”Coin”>
    <Location x=”15.894” y =”18.713”/>
    <SlotCavityRef id="SC1"/>
</Component>
```

Note that the component IC1 would not have a `SlotCavityRef` since it is not placed inside the cavity.

8.2.3.8 LogicalNet

The `LogicalNet` section is a list of `LogicalNet` elements, each with a name and a group of component/pin location(s). It enables the labeling of each pin with the net to which it belongs. The `PhyNetGroupList` is another representation of a netlist, using physical board locations instead of logical pins.

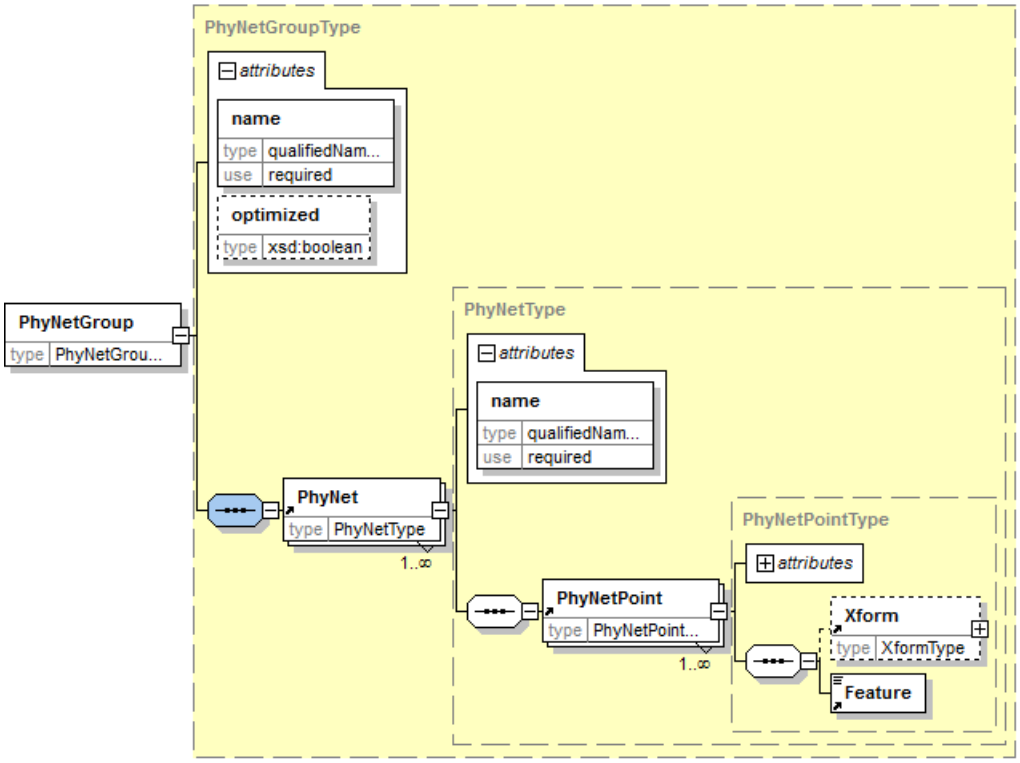
`LogicalNet` elements read from the CAD system in the form of component pins connectivity. Each `LogicalNet` contains the net name and a set of `LogicalNetPin`. Each `LogicalNetPin` points to a pin on a component.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
LogicalNet	LogicalNetType	The `LogicalNet` element consists of the characteristics that are all electrically common. These can include conductor, vias, or planes.	0-n
name	qualifiedNameType	The unique name assigned to the electrical description of the net.	1-1
netClass	netClassType	An enumerated string identifying one of the following net class types CLK | FIXED | GROUND | SIGNAL | POWER | UNUSED	0-1
netPair	qualifiedNameType	The unique name assigned to the net that forms a differential pair with the net name	0-1
NonstandardAttribute	NonstandardAttribute Type	A nested element that can be used to provide extra details regarding a particular “LogicalNet” description in terms of BOOLEAN, DOUBLE, INTEGER, or STRING data. The properties of the NonstandardAttribute become unique due to its’ Name.	0-n
PinRef	PinRefType	An individual `Pin` related to the place where a component attaches to the net. This description facilitates location of lands on the surface of the board or places where through-hole components are mounted.	1-n
PortRef	PortRefType	An optional reference to a unique ID of a port that interfaces to another `Step` with the file. The other `Step` would represent another design entity, such as a daughter card or custom IC.	0-n
SpecRef	SpecRefType	A unique reference to a specification pertaining to the `LogicalNet`	0-n

8.2.3.9 PhyNetGroup

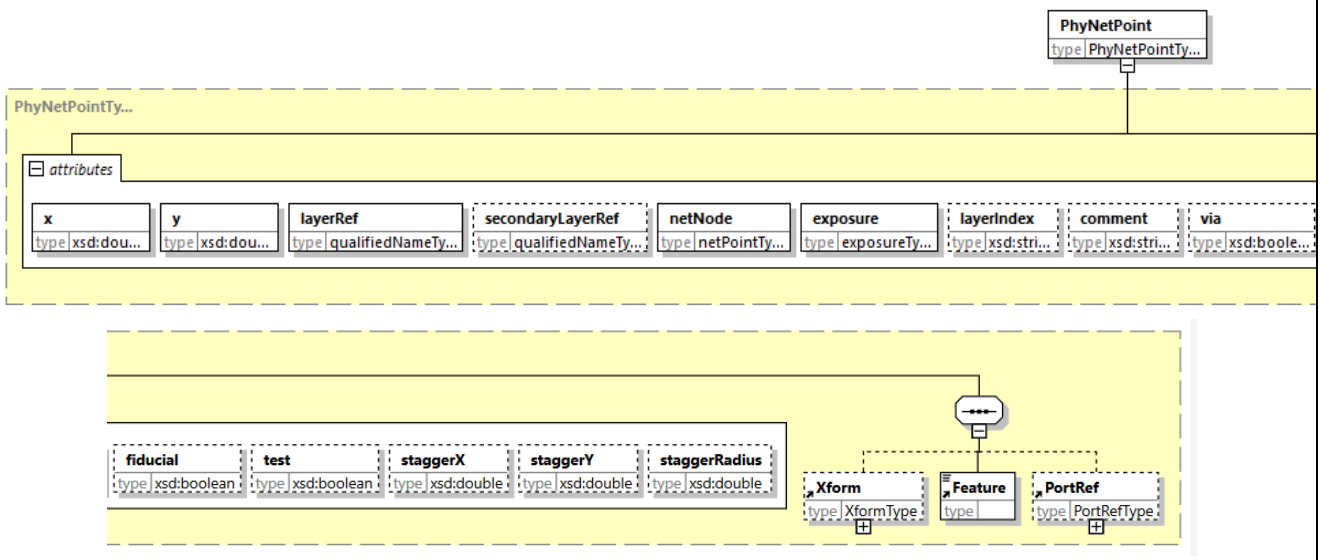
The `PhyNetGroup` element consists of various physical electrical connections. The group of nets may be combined from individual layers and submitted to a netlist analyzer or read from netlist files. Each `PhyNetGroup`, contains a set of one to many physical nets (`PhyNets`).



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
PhyNetGroup	PhyNetGroupType	A group of various physical nets combined in specific format and named accordingly.	0-n
name	qualifiedNameType	A unique name assigned to the PhyNetGroup.	1-1
optimized	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lower case). True equals that the `PhyNetGroup` has been optimized by combining all PhyNets into a convenient description under the `PhyNetGroup` element; False indicates that the `PhyNetGroup` has not been optimized. If the attribute is not present the optimization condition is unknown.	1-1
PhyNet	PhyNetType	An embedded element that provides all the characteristics of a `PhyNet` describing the characteristics needed to interconnect components in the electronic product.	1-n
name	qualifiedNameType	A unique name assigned to a specific `PhyNet`.	1-1
PhyNetPoint	PhyNetPointType	An embedded element that provides the details for the `PhyNet` location and characteristics.	1-n

8.2.3.9.1 PhyNet

The `PhyNet` element consists of one to many points that are essentially the nodes for the physical description of all the conductive elements that become a part of the `Net` on a particular surface of the board. The `PhyNetPoint` is only available on either top or bottom or both and, may also pertain to the concepts used for embedded passive or active component pin description.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
PhyNet	PhyNetType	An embedded element that provides all the characteristics of a `PhyNet` describing the characteristics needed to interconnect components in the electronic product.	1-n
name	qualifiedNameType	A unique name assigned to the `PhyNet`.	1-1
PhyNetPoint	PhyNetPointType	An embedded element that provides the details for the `PhyNet` location and characteristics.	1-n
x	double	The x-location for the `PhyNetPoint`.	1-1
y	double	The y-location for the `PhyNetPoint`.	1-1
layerRef	qualifiedNameType	The reference to the layer to which the physical net pertains as identified by the layer name including defining a reference to an internal layer for embedded component net relationships.	1-1
secondaryLayerRef	qualifiedNameType	The reference to a secondary layer to which the physical net pertains as identified by the layer name usually to the opposite side of the board and includes defining a reference to an internal layer for embedded component net relationships.	0-1
netNode	NetPointType	A `NetPointType` may be one of END | MIDDLE to indicate where the PhyNet should be probed defining the end of the Net or a conductor at midpoint.	1-1
exposure	exposureType	The exposure attribute indicates whether the `NetPoint` is accessible for probing. The enumerated strings consist of: EXPOSED | COVERED_PRIMARY | COVERED_SECONDARY | COVERED	1-1
layerIndex	string	An identification related to inner layer testing prior to multilayer lamination.	0-1
comment	string	Any comment pertaining to the probing of PhyNetPoints.	0-1
via	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lowere case). True equals that the `via` is being used as the probe point; False indicates that the `via` is not available. If the attribute is not present the `via` probing condition is unknown.	0-1

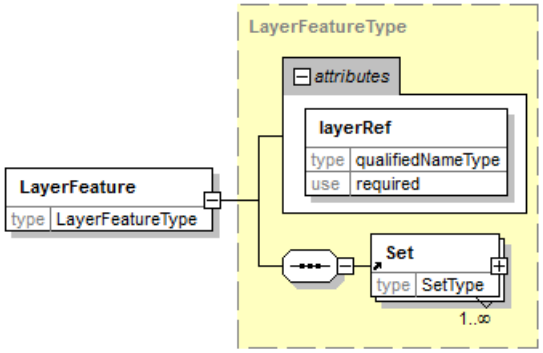
fiducial	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lower case). True equals that the `fiducial` is being used as the probe point; False indicates that the `fiducial` is not available. If the attribute is not present the `fiducial` probing condition is unknown	0-1
test	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lower case). True equals that the `PhyNetPoint` is part of the test sequence allowing for full nodal access, partial nodal access, or functional testing; False indicates that the `PhyNetPoint` is not part of the test sequence. If the attribute is not present the test condition is unknown.	0-1
staggerX	double	An X dimension that differs from the original X of the `PhyNetPoint` to indicate a probing location that varies from that specific point.	0-1
staggerY	double	A y-dimension that differs from the original x of the `PhyNetPoint` to indicate a probing location that varies from that specific point.	0-1
staggerRadius	double	A numerical value that indicates a radius taken from the original x-y point description in the direction of an open conductor that may be probed at its center.	0-1
Xform	XformType	An element that provides the ability to reset the point of origin of a predefined feature that can be scaled, mirror imaged or rotated.	0-1
Feature	ABSTRACT	An embedded element that defines a substitution group of any `StandardShape` or `UserShape` that may be instantiated as a part of the `pad` description. A predefined `StandardShape or UserShape` may also be instantiated by its unique “id” when the feature is contained in the `DictionaryStandard or User`. When a reference is made to the dictionary predefined primitive the `Units` must match.	1-1
PortRef	PortRefType	A reference to the unique ID of a `Port`, of type `Wirebond`, `ConnectrMate`,or `ComponentPad`	0-1

8.2.3.10 LayerFeature

The `LayerFeature` element contains all the physical features located on all layers. These features reference `StandardPrimitive` or `UserPrimitive` under the substitution group identification of `StandardShape` or `UserShape`. All shapes may be identified through a reference to predefined primitives contained in `DictionaryStandard` or `DictionaryUser`. Shapes may also be instantiated in the file by substitution of the `Feature` element with the shape name. All characteristics of any shape must be present when the substitution takes place.

An individual `LayerFeature` can be thought of as artwork and these two-dimensional descriptions become the main body of the `Step` data. The information is contained in `LayerFeature` elements and includes several different elements, each corresponding to a layer defined earlier in the `Layer` element.

The `Set` element defines modal attributes (attributes are in effect for all subsequent graphics contained in the set until changed). The only one important characteristic for the set graphic is the `polarity` attribute that can be POSITIVE (draw) or NEGATIVE (erase). The existence of negative features is the reason for the importance of the order.

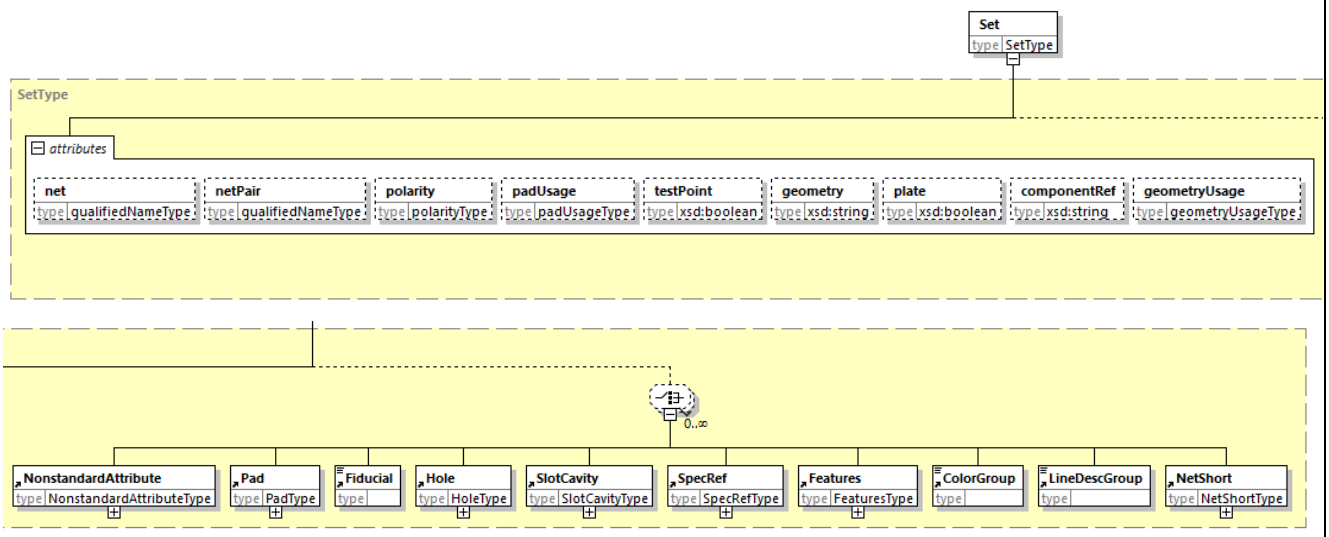


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
LayerFeature	LayerFeatureType	An element that contains various sets of features that are grouped into various segments, each with applicability to define the features on a specific layer. The conditions relate to the type of features that are contained within the `LayerFeature` element and these are grouped according to the usage that the two-dimensional graphics have for providing graphical images on printed boards documentation or assembly descriptions. These conditions are defined through `layerRef`.	0-n
layerRef	qualifiedNameType	The reference to the unique layer name to which the `LayerFeature`(s) pertain.	1-1
Set	SetType	The multiple `Set` elements and attributes used to define specific features associated with a specific layer as identified by the layer and determined by the LayerFunction characteristics	1-n

```
<Step name = "KarensFabricationPanel">
   <Datum x = "0.00" y = "0.00"/>
      <Profile>
         <Polygon>
            <PolyBegin x = "-305.00" y = "-230.00"/>
            <PolyStepSegment x = "-305.00" y = "230.00"/>
            <PolyStepSegment x = "305.00" y = "230.00"/>
            <PolyStepSegment x = "305.00" y = "-230.00"/>
            <PolyStepSegment x = "-305.00" y = "-230.00"/>
         </Polygon>
      </Profile>
      <LayerFeature layerRef = "KarensMultilayer">
         <Set>
                        SlotCavity name = "Tooling Slots" platingStatus = "NONPLATED" plusTol = "0.02" minusTol = "0.00">
<Outline>
                  <Polygon>
                     <PolyBegin x = "1.59" y = "209.29"/>
                     <PolyStepSegment x = "1.59" y = "210.71"/>
                     <PolyStepCurve x = "-1.59" y = "210.71" centerX = "0.00" centerY = "210.71" clockwise = “false”/>
                     <PolyStepSegment x = "-1.59" y = "209.29"/>
                     <PolyStepCurve x = "1.59" y = "209.29" centerX = "0.00" centerY = "209.29" clockwise = “false”/>
               </Outline>
               <Outline>
                  <Polygon>
                     <PolyBegin x = "1.59" y = "-209.29"/>
                     <PolyStepSegment x = "1.59" y = "-210.71"/>
                     <PolyStepCurve x = "-1.59" y = "-210.71" centerX = "0.00" centerY = "-210.71" clockwise = “true”/>
                     <PolyStepSegment x = "-1.59" y = "-209.29"/>
                     <PolyStepCurve x = "1.59" y = "-209.29" centerX = "0.00" centerY = "-209.29" clockwise = “true”/>
                  </Polygon>
               </Outline>
               <Outline>
                  <Polygon>
                     <PolyBegin x = "289.29" y = "1.59"/>
                     <PolyStepSegment x = "290.71" y = "1.59"/>
                     <PolyStepCurve x = "290.71" y = "-1.59" centerX = "290.71" centerY = "0.00" clockwise = “true”/>
                     <PolyStepSegment x = "289.29" y = "-1.59"/>
                     <PolyStepCurve x = "289.29" y = "1.59" centerX = "289.29" centerY = "0.00" clockwise = “true”/>
                  </Polygon>
               </Outline>
                  <Outline>
                     <Polygon>
                        <PolyBegin x = "-289.29" y = "1.59"/>
                        <PolyStepSegment x = "-290.71" y = "1.59"/>
                        <PolyStepCurve x = "-290.71" y = "-1.59" centerX = "-290.71" centerY = "0.00" clockwise = “false”/>
                        <PolyStepSegment x = "-1.59" y = "-289.29"/>
                        <PolyStepCurve x = "-289.29" y = "1.59" centerX = "-289.29" centerY = "0.00" clockwise = “false”/>
                     </Polygon>
                  </Outline>
               </Slot>
               <LineDesc lineEnd = "NONE" lineWidth = "0.00"/>
            </Set>
         </LayerFeature>
```

8.2.3.10.1 Set

A specific set of graphical descriptions for a particular set of graphical shapes. These shapes are applied defining the conductive pattern of the printed board.

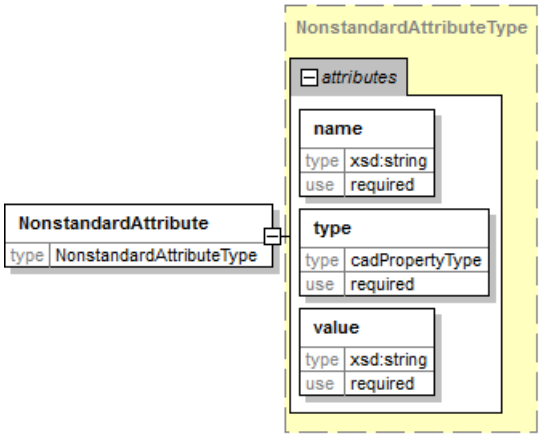


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Set	SetType	The multiple Set elements and attributes defined in 8.3.1 used to define specific features associated with a conductive layer.	1-n
net	qualifiedNameType	The electrical relationship of any feature, through the name of the `PhyNet`, when the `Set` feature has conductivity checked in the `PhyNetPoint` descriptions. This attribute is left blank if the `Set` descriptions are for other than printed board fabrication or assembly conductivity.	0-1
netPair	qualifiedNameType	The unique name assigned to the net that forms a differential pair with the net	0-1
polarity	polarityType	`Polarity` indicates whether the information described in the `Set` is POSITIVE | NEGATIVE. A NEGATIVE connotation can be used to describe the removal of a dark field to the specific dimensions described for another attribute. Thus, a surface that contains islands may have the islands described in a negative format.	0-1
padUsage	padUsageType	An indication as to the usage of any pad that becomes a part of the `LayerFeature Set`. The descriptions are enumerated strings and must be one of the following: TERMINATION | VIA | PLANE | TOOLING_HOLE | FIDUCIAL | MASK | THIEVING | THERMAL_RELIEF | NONE.	0-1
testPoint	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lower case). True indicates that the feature is a candidate for a test point used for either in-circuit or functional testing. False indicates that it is not.	0-1
geometry	string	An identification to describe the overall geometry of the features contained in the `Set` and their particular application to the electronic product.	0-1
plate	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lower case). True indicates that the feature is plated in a secondary operation. False indicates that it is not.	0-1
componentRef	string	A reference to the component instance associated with the reference designation that links the information together so that graphic representation in the Set can be accurately described in Assembly or Silk screen documentation.	0-1

geometryUsage	geometryUsageType	An indication as to the usage of any geometry that becomes a part of the LayerFeature Set. The descriptions are enumerated strings and must be one of the following: THIEVING | THERMAL_RELIEF | TEXT | TEARDROP | GRAPHIC | NONE.	0-1
NonstandardAttribute	NonstandardAttribute Type	A nested element that can be used to provide extra details regarding a particular "Set" description in terms of BOOLEAN, DOUBLE, INTEGER, or STRING data. The properties of the NonstandardAttribute become unique due to its' Name.	0-n
Pad	PadType	A series of pads that are associated with the `LayerFeature Set`.	0-n
Fiducial	ABSTRACT	A substitution that consists of four elements that may be used to replace the fiducial element. When the `Fiducial` element is substituted it **shall** be by a `Global`, `Local`, `BadBoardMark`, or `GoodPanelMark`.	0-n
Hole	HoleType	A series of holes associated with the `LayerFeature Set`.	0-n
SlotCavity	SlotCavityType	A series of slots or cavities associated with the `LayerFeature Set`.	0-n
SpecRef	SpecRefType	The identification of a particular specification(s) from the SpecificationType that is able to provide additional details or instructions that apply to the Hole, SlotCavity or Features of the Set descriptions.. The linkage is provided through the specific "specificationId" (spec Name)	0-n
Features	FeaturesType	An embedded element that defines a substitution group of any predefined `StandardShape` or `UserShape` that may be instantiated as a part of the `LayerFeature Set`.	0-n
ColorGroup	ABSTRACT	A substitution group that permits assigning a particular color through instantiating the three basic colors or by providing a reference to a predefined `Color` in `DictionaryColor`.	0-n
LineDescGroup	ABSTRACT	A substitution group that specifies the `LineWidth` and `LineEnd` characteristics of a `Feature` that requires that description. If a predefined feature is instantiated the presents of a `LineDescGroup` will override the previously defined `LineDesc`.	0-n
NetShort	NetShortType	The identification of two or more nets that are intentionally shorted	0-n

8.2.3.10.2 NonstandardAttribute

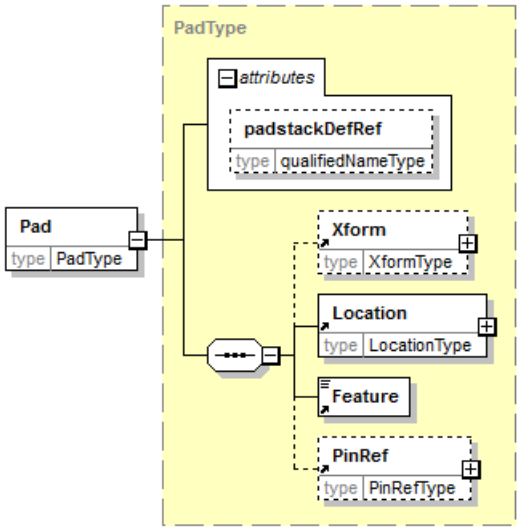
The NonstandardAttribute element consists of various conditions that may be used in association with the `Set` element. The NonstandardAttribute element has several attributes which include a name, type and value thus making each NonstandardAttribute unique. The type attribute is a cadPropertyType and identifies the "value" attribute as being BOOLEAN, DOUBLE, INTEGER or STRING data



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Nonstandard Attribute	NonstandardAttribute Type	A nested element that can be used to provide extra details regarding a particular "Set" description in terms of BOOLEAN, DOUBLE, INTEGER, or STRING data. The properties of the NonstandardAttribute become unique due to its' Name.	0-n
name	qualifiedNameType	A unique name assigned by the user to the NonstandardAttribute. The name must be unique and should match the purpose of the actions to be taken.	1-1
type	cadPropertyType	A set of enumerated string descriptions that identify the condition of the value requirement. They consist of BOOLEAN | DOUBLE | INTEGER or STRING data.	1-1
value	string	The detail requirements as they pertain to the particular Step description and interpreted according to the type enumeration that is part of the particular NonstandardAttribute.	1-1

8.2.3.10.3 Pad

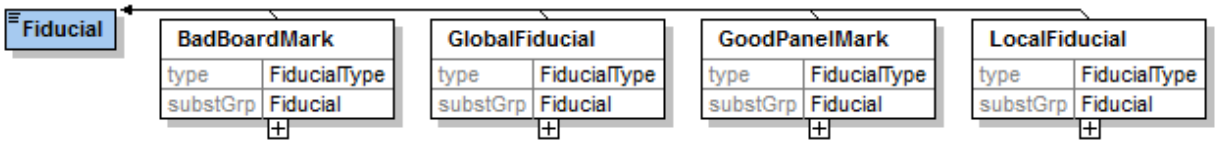
The `Pad` element represents an individual pad. `Pads` are features with a center (`x, y`), a standard primitive shape or User defined shape either pre-defined in the `DictionaryStandard`, or `DictionaryUser` instanced at the time the `Set` is defined. The `Pad` may be changed through the `Xform` element (located, rotated, mirrored or scaled). `Rotation` is any number of degrees, although 90º multiples is the usual angle; positive rotation is always counter-clockwise as viewed from the board TOP (primary side). When `mirror` is set to MIRROR it indicates that all x dimensions are set to a–x value. For scaling the `Pad`, all `x` and `y` dimensions of a `geometry` are multiplied by the `scale` attribute. The scale factor does not apply to angular values. The `Pad` may have an appropriate `pin` attribute.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Pad	PadType	A series of pads associated with the `LayerFeature Set`.	0-n
padstackDefRef	qualifiedName	A reference to the name of the padstackDef element that provides the characteristics of individual padstacks in the design, as established by the CAD system, and provides the link to the layer distribution of the padstack's individual characteristics.	0-1
Xform	XformType	An element that provides the ability to reset the point of origin of a predefined area that can be scaled, mirror imaged or rotated. The image may also by located by the `Xform`. See 3.3.	0-1
Location	LocationType	The location of the image defined by the standard shape or Usershape or a pre-defined standard shape, or Usershape of the pad. The image may have been reorientated by the Xform.	1-1
Feature	ABSTRACT	An embedded element that defines a substitution group of any `StandardShape` or `UserShape` that may be instantiated as a part of the `pad` description. A predefined `StandardShape or UserShape` may also be instantiated by its unique "id" when the feature is contained in the `DictionaryStandard or User`. When a reference is made to the dictionary predefined primitive the `Units` must match.	1-1
PinRef	PinRefType	An individual `Pin` related to the place where a component attaches to the net. This description facilitates location of lands on the surface of the board or places where through-hole components are mounted.	0-n

8.2.3.10.4 Fiducial

A specific set of fiducials used by the board fabricator to distinguish between those boards, in a panel, that passed inspection or electrical test, and those that did not pass.



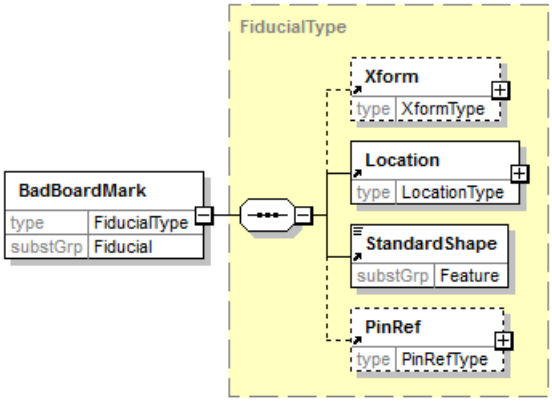
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
BadBoardMark	FiducialType	A specific set of fiducials used by the board fabricator to distinguish between those boards, in a panel, that passed inspection or electrical test, and those that did not pass . The `Fiducial` is positioned near each board in the assembly panel array.	0-n
GlobalFiducial	FiducialType	An individual `Set` of fiducials used in the description and arrangement of features on a board, assembly, or panel. This symbol is also used as a good board mark	0-n
GoodPanelMark	FiducialType	A single Standard Shape used to define a panel where all boards on the panel are good. The fiducial is positioned once on the panel and enables reduction in inspection time.	0-n
LocalFiducial	FiducialType	An individual `Set` of fiducials used in the description and arrangement of features on a board, assembly, or panel which represent component location positioning.	0-n

❖ BadBoardMark

The BadBoardMark element provides a list of images intended to represent a symbol known as a fiducial which works with equipment vision systems to identify whether the board in the array is good or not. The determination is usually made by the board fabricator and he covers the fiducial (BadBoardMark) to indicate that the board should not be assembled with components.

These images are usually described in the form of a StandardShape and may appear on any Layer as a LayerFeature. The BadBoardMark may also be identified as a separate Set, thus arranging all of the BadBoardMark elements that identify good and bad boards on an assembly array, or manufacturing panel.

If treated individually, BadBoardMark elements may appear multiple times within the LayerFeature Set.

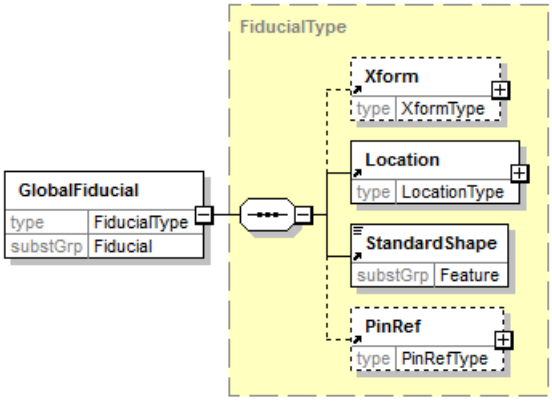


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
BadBoardMark	FiducialType	An individual set of fiducials used as an aid to the board fabricator to be used to define those boards in a panel that did not pass inspection or electrical test. The Fiducial is positioned near each board in the assembly panel array.	0-n
Xform	XformType	An element that provides the ability to reset the point of origin of a predefined area that can be scaled, mirror imaged or rotated. The image may also by located by the Xform. See 3.3.	0-1
Location	LocationType	The location of the image defined by the standard shape or a pre-defined standard shape of the fiducial. The image may have been reorientated by the Xform.	1-1
StandardShape	ABSTRACT	A substitution group that may call for any StandardPrimitive, instantiated by describing their unique features or by referencing a predefined primitive contained in the DictionaryStandard.	1-1
PinRef	PinRefType	An optional reference to a pin, which is not applicable in this case	0-n

❖ GlobalFiducial

The `GlobalFiducial` element provides a list of images intended to represent a symbol known as a fiducial which works with assembly equipment vision systems to improve the positioning of the board or panel. These images are described in the form of a `StandardShape` and may appear on any `Layer` as a `LayerFeature`. The `GlobalFiducial` may also be identified as a separate `Set`, thus arranging all of the fiducials that position boards, assemblies, and assembly arrays in a panel format are considered in one `LayerFeature Set`.

If treated individually, `GlobalFiducial` may appear multiple times within the `LayerFeature Set`.

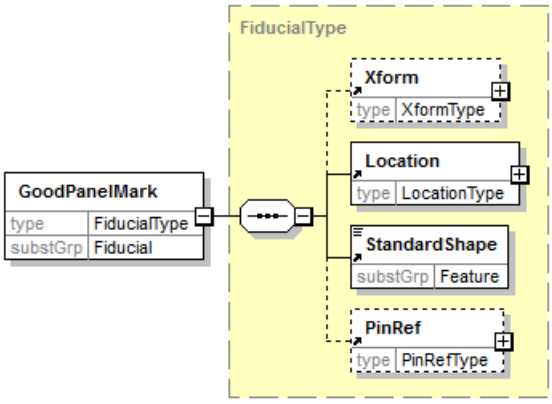


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
GlobalFiducial	FiducialType	An individual `Set` of fiducials used in the description and arrangement of features on a board, assembly, or panel.	0-n
Xform	XformType	An element that provides the ability to reset the point of origin of a predefined area that can be scaled, mirror imaged or rotated. The image may also by located by the `Xform`. See 3.3.	0-1
Location	LocationType	The location of the image defined by the standard shape or a pre-defined standard shape of the fiducial. The image may have been reorientated by the Xform.	1-1
StandardShape	ABSTRACT	A substitution group that may call for any `StandardPrimitive`, instantiated by describing their unique features or by referencing a predefined primitive contained in the `DictionaryStandard`.	1-1
PinRef	PinRefType	An optional reference to a pin, which is not applicable in this case	0-n

❖ GoodPanelMark

The `GoodPanelMark` element provides a single image intended to represent a symbol known as a fiducial which works with equipment vision systems to identify that all the boards in an array are good. The determination is usually made by the board fabricator. He makes sure that the `GoodPanelMark` fiducial is clearly visible to avoid having to check to see if there are any bad boards.

These images are usually described in the form of a `StandardShape` and may appear on any `Layer` as a `LayerFeature`.

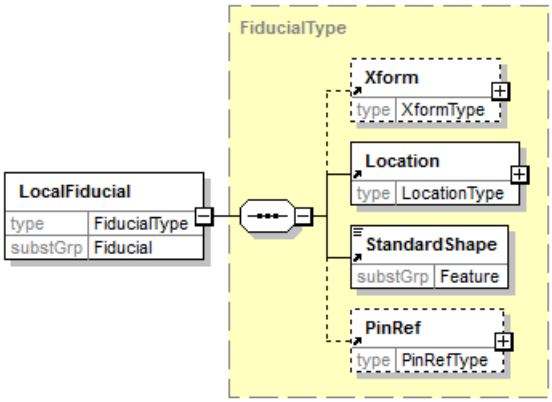


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
GoodPanelMark	FiducialType	An individual fiducial or set of fiducials used in the description of features on a panel that identifies that all properties within the panel periphery meet the appropriate performance specification.	0-n
Xform	XformType	An element that provides the ability to reset the point of origin of a predefined area that can be scaled, mirror imaged or rotated. The image may also by located by the `Xform`. See 3.3.	0-1
Location	LocationType	The location of the image defined by the standard shape or a pre-defined standard shape of the fiducial. The image may have been reorientated by the Xform.	1-1
StandardShape	ABSTRACT	A substitution group that may call for any `StandardPrimitive`, instantiated by describing their unique features or by referencing a predefined primitive contained in the `DictionaryStandard`.	1-1
PinRef	PinRefType	An optional reference to a pin, which is not applicable in this case	0-n

❖ LocalFiducial

The LocalFiducial element provides a list of images intended to represent a symbol known as a fiducial which works with specific components that require the additional precision of assembly equipment vision systems to improve the positioning of the component during the assembly operation. These images are usually described in the form of a StandardShape and may appear on any Layer as a LayerFeature. The LocalFiducial may also be identified as a separate Set, thus arranging all of the fiducials that position components or other specific features on a board, assembly array, or manufacturing panel.

If treated individually, the LocalFiducial may appear multiple times within the LayerFeature Set.

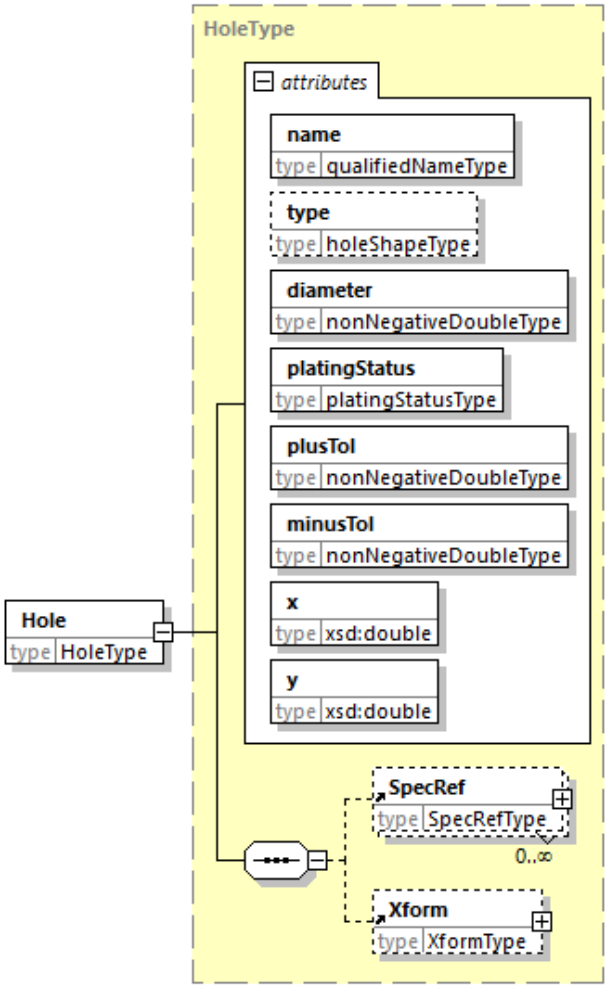


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
LocalFiducial	FiducialType	An individual Set of fiducials used in the description and arrangement of features on a board, assembly, or panel which represent component location positioning.	0-n
Xform	XformType	An element that provides the ability to reset the point of origin of a predefined area that can be scaled, mirror imaged or rotated. The image may also by located by the Xform. See 3.3.	0-1
Location	LocationType	The location of the image defined by the standard shape or a pre-defined standard shape of the fiducial. The image may have been reorientated by the Xform.	1-1
StandardShape	ABSTRACT	A substitution group that may call for any StandardPrimitive, instantiated by describing their unique features or by referencing a predefined primitive contained in the DictionaryStandard.	1-1
PinRef	PinRefType	An optional reference to a pin of an associated component	0-n

8.2.3.10.5 Hole

The Hole element describes the characteristics of a particular hole, including naming the hole description with a unique name that may be reused. The main purpose of including hole in the Set means that specific information can be described as all the particular holes in one set of data. In this instance, the layerRef of LayerFeature is to the Layer/Stackup element which describes the overallThickness for those holes that go entirely through the board. For those holes that are buried or blind vias, the appropriate Stackup reference **shall** be used as a part of the layerRef of the LayerFeature descriptions of holes. This concept permits a replacement of the Drill file that usually accompanies a data transfer transaction.

The Hole element can occur multiple times within the LayerFeature element.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Hole	HoleType	A series of holes associated with the `LayerFeature Set`.	0-n
name	qualifiedNameType	A unique identification of a particular hole.	1-1
type	holeShapeType	The type of hole shape. Legal values are CIRCLE and SQUARE. Default type is CIRCLE.	0-1
diameter	nonNegativeDoubleType	The nominal diameter of the hole in the as-finished state.	1-1
platingStatus	platingStatusType	The type of hole defined as an enumerated string indicating PLATED | NONPLATED | VIA | VIA_CAPPED	1-1
plusTol	nonNegativeDoubleType	The plus tolerance variation permitted from the nominal hole diameter.	1-1
minusTol	nonNegativeDoubleType	The minus tolerance variation permitted from the nominal hole diameter.	1-1
x	double	The x-location of the hole.	1-1
y	double	The y-location of the hole.	1-1
SpecRef	SpecRefType	The identification of a particular specification(s) from the `SpecificationType` that is able to provide additional details or instructions that apply to the board or assembly descriptions of the Layer. The linkage is provided through the specific `specificationId` (spec Name). Example specifications are `Backdrill and SecondaryDrill`. Other specifications may also be attached if appropriate.	0-n

Xform	XformType	Optional element used for square hole rotation	0-1

```
<Spec name="BD_1B">
     <Backdrill type="START_LAYER" >
        <Property layerOrGroupRef="BOTTOM"/>
     </Backdrill>
     <Backdrill  type="MUST_NOT_CUT_LAYER" >
        <Property layerOrGroupRef="IL4"/>
     </Backdrill>
    <Backdrill type="MAX_STUB_LENGTH" >
      <Property Value="0.008" unit="INCH"/>
    </Backdrill>
</Spec>
<Spec name="BD_1A">
       <Backdrill type="START_LAYER" >
         <Property layerOrGroupRef="TOP"/>
       </Backdrill>
       <Backdrill type="MUST_NOT_CUT_LAYER">
         <Property layerOrGroupRef="IL-3"/>
      </Backdrill>
      <Backdrill type="MAX_STUB_LENGTH" >
        <Property Value="0.008" unit="INCH"/>
      </Backdrill>
</Spec>

<Set>
   <Hole name="H122" diameter="0.017" platingStatus="VIA" plusTol="0.001" minusTol="0.001000" x="3.125" y="5.025" >
   <SpecRef Id =”BD_1A”/>
   <SpecRef Id=”BD_1B”/>
   </Hole>
   <Hole name="H123" type=”SQUARE” diameter="0.05"  platingStatus="PLATED" plusTol="0.001" minusTol="0.001" x="3.5" y="5.525">
       <Xform rotation="45.000"/>
   </Hole>
</Set>
```

```
<Spec name="BD_1B">
    <Backdrill type="START_LAYER" >
       <Property layerOrGroupRef="BOTTOM"/>
    </Backdrill>
    <Backdrill  type="MUST_NOT_CUT_LAYER" >
       <Property layerOrGroupRef="IL4"/>
    </Backdrill>
   <Backdrill type="MAX_STUB_LENGTH" >
     <Property Value="0.008" unit="INCH"/>
   </Backdrill>
</Spec>
<Spec name="BD_1A">
      <Backdrill type="START_LAYER" >
        <Property layerOrGroupRef="TOP"/>
     </Backdrill>
     <Backdrill type="MUST_NOT_CUT_LAYER">
       <Property layerOrGroupRef="IL-3"/>
    </Backdrill>
    <Backdrill type="MAX_STUB_LENGTH" >
      <Property Value="0.008" unit="INCH"/>
    </Backdrill>
</Spec>

<Set>
  <Hole name="H122" diameter="0.017000" platingStatus="VIA" plusTol="0.001000" minusTol="0.001000" x="3.125000" y="5.025000">
      <SpecRef Id ="BD_1A"/>
      <SpecRef Id="BD_1B"/>
  </Hole>
</Set>
```

8.2.3.10.6 SlotCavity

The `SlotCavity` element describes a feature created by a machining operation that removes material from a bare board within a given shape. The shape is defined by the substitution group `Feature`, which can be either a user defined shape or a standard primitive shape. The feature can be plated or nonplated. The `SlotCavity` element can occur multiple times within the `LayerFeature Set` of a layer. There are two types of features that it can define, as described below.

8.2.3.10.6.1 Cut To Depth/Thickness

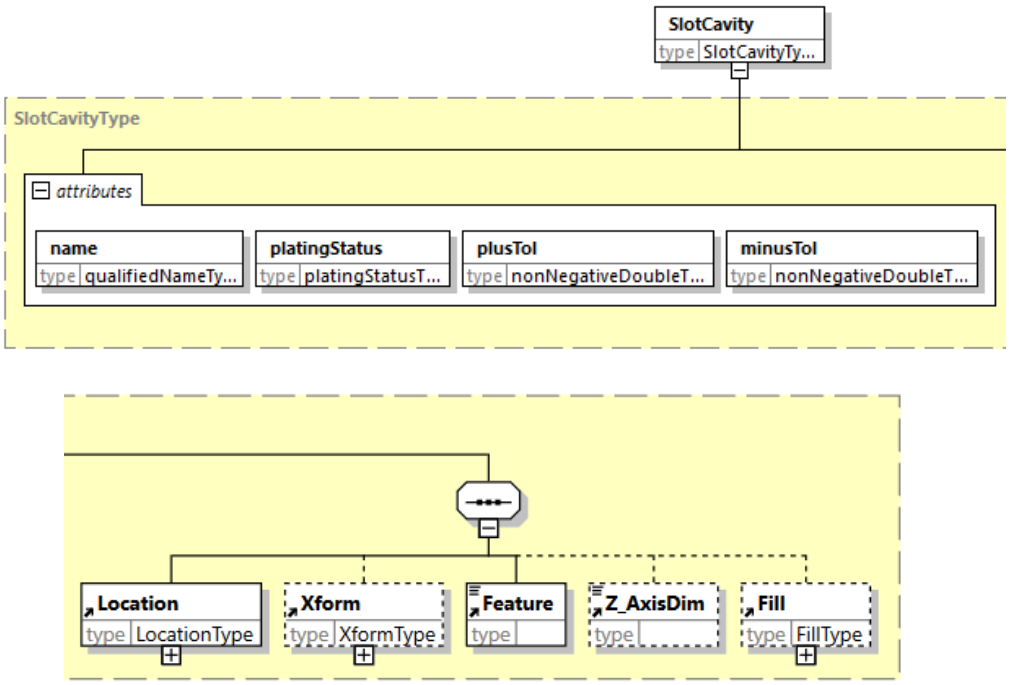
This operation is identified by the presence of the optional child substitution group `Z_AxisDim`, which can be either `MaterialCu`t, which specifies a cut depth, or `MaterialLeft`, which specifies a thickness remaining. It can be performed on the full layer set that has completed the lamination process, or on a single layer or subset of layers prior to lamination. The cut can start from the top or bottom of the layer set, but not cut all the way through. If the cut is performed on a single layer then the optional attribute `direction` is required to specify whether the cut is DOWN (from the top) or UP (from the BOTTOM). The resulting open cavity has a flat floor that is parallel to the board surface, and sides that are perpendicular to the board surface. The location of the feature is typically planned so that internal copper etch is avoided when cutting. There may be a through hole drill or drills anywhere within the feature, added either before or after its creation. Note however that for a counterbore, which is specifically a circular cutout concentric to a single drill, the `SecondaryDrill` Specification should be used (see section 8.1.1.14). Examples are partial depth slots or grooves (but not v-grooves/v-scores – see `V_Cut` Specificaton, section 8.1.1.12).

8.2.3.10.6.2 Cut To Span

The absence of the optional substitution group `Z_AxisDim` denotes a feature created by a machining operation that cuts a given shape completely through either all layers, or a subset of layers of a bare board. A cut through a subset of layers implies sequential lamination, and may impact the order of construction of the layer stack. If the cut laminate then has additional layers added on either side then the cut feature becomes an embedded cavity. If additional layers are added to just one side then the feature becomes an open cavity.

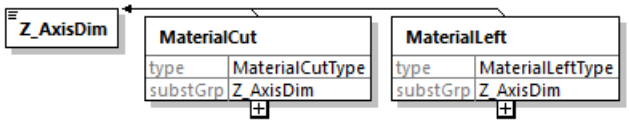
The extent of the cut through the layer stack is controlled by a `Span` child element of the routing layer containing the `SlotCavity` elements. The `fromLayer` attribute of `Span` denotes the first layer to be cut. The `toLayer` attribute denotes the last layer to be cut. If a `Span` child element is not present then the cut defaults to all the way through the

board. Note that if a SlotCavity element with a Z_AxisDim child element is defined on a layer that has a Span child element then the Z_AxisDim shall take precedence, and the Span shall be ignored.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
SlotCavity	SlotCavityType	A series of slots or cavities associated with the LayerFeature Set.	0-n
name	qualifiedNameType	A unique identification of a particular slot or cavity	1-1
platingStatus	platingStatusType	The type of slot defined as an enumerated string indicating PLATED | NONPLATED | VIA.	1-1
plusTol	nonNegativeDouble Type	The plus tolerance variation permitted from the nominal dimensions	1-1
minusTol	nonNegativeDouble Type	The minus tolerance variation permitted from the nominal dimensions	1-1
Location	LocationType	The location of the feature. The feature may have been reorientated by the Xform.	1-1
Xform	XformType	An element that provides the ability to reset the point of origin of a feature that can be scaled, mirror imaged or rotated.	0-1
Feature	ABSTRACT	A substitution group to define either a StandardShape or a UserShape	1-1
Z_AxisDim	ABSTRACT	A substitution group to define Z axis dimensions for the slot or cavity in terms of MaterialCut or MaterialLeft.	0-1
Fill	FillType	A child element that describes any full or partial fill material in the SlotCavity. The fill material is assumed to completely surround any component(s) in the SlotCavity.	0-1

8.2.3.10.6.3 Z_AxisDim

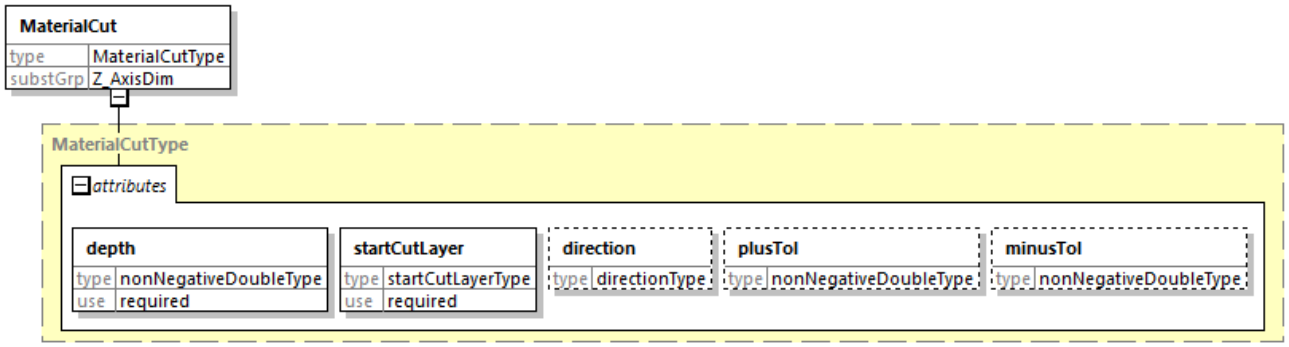


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
MaterialCut	MaterialCutType	The descriptions of the material that is being removed	0-1
MaterialLeft	MaterialLeftType	The description of the material that is to remain from a specified thickness after some material is removed.	0-1

The depth is an example of the Z_AxisDim and is identified as MATERIALCUT.

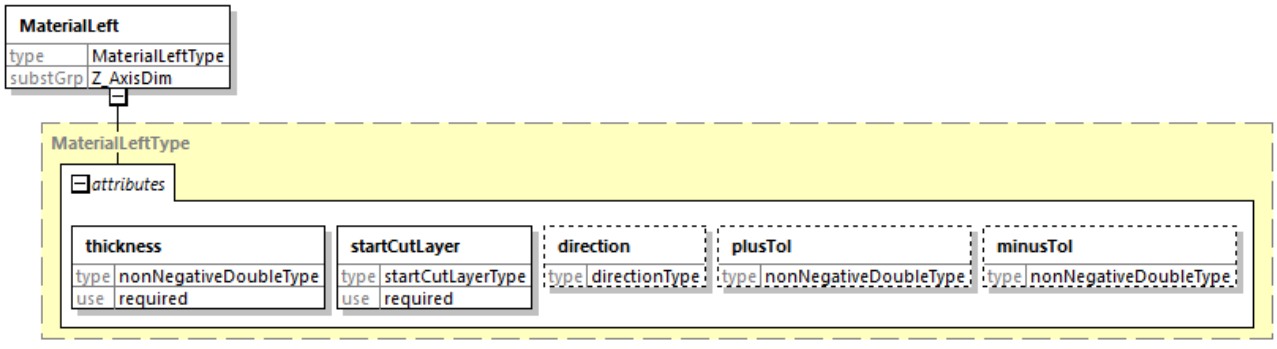
The thickness is an example of the Z_AxisDim and is identified as MATERIALLEFT

8.2.3.10.6.3.1 MaterialCut



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
MaterialCut	MaterialCutType	The descriptions of the material that is being removed	0-1
depth	nonnegativeDoubleType	The nominal dimension of the depth of the cut in material removal from a specified thickness	1-1
startCutLayer	qualifiedNameType	The unique name of the layer, from the list of `Layer` definitions, which is the first to be cut. It is the name of the layer in the final stackup, after all laminations processes are complete. If an inner layer is specified then sequential lamination is implied, with that layer being an outer layer of the layer subset to be cut.	1-1
direction	directionType	An optional enumerated string with values DOWN|UP that denotes the direction of the cut, Referring to Figure 2, DOWN means in the direction of moving from the primary (top) side of the board to the secondary (bottom) side of the board, and UP means in the opposite direction. If cutting more than one layer then the cut direction can possibly be inferred from `startCutLayer`, but if cutting just one layer, or the direction is ambiguous, then `direction` is required.	0-1
plusTol	nonnegativeDoubleType	The plus tolerance variation permitted from the nominal depth dimension	0-1
minusTol	nonnegativeDoubleType	The minus tolerance variation permitted from the nominal depth dimension	0-1

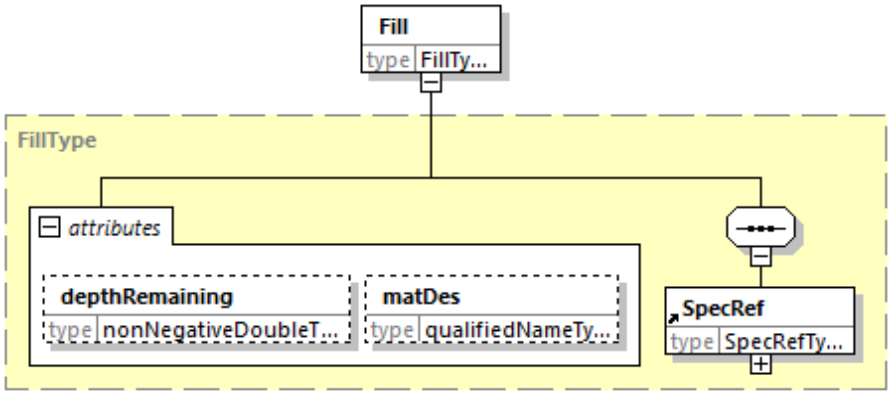
8.2.3.10.6.3.2 MaterialLeft



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
MaterialLeft	MaterialLeftType	The description of the material that is to remain from a specified thickness after some material is removed.	0-1
thickness	nonnegativeDoubleType	The nominal dimension of the thickness of the remaining material.	1-1

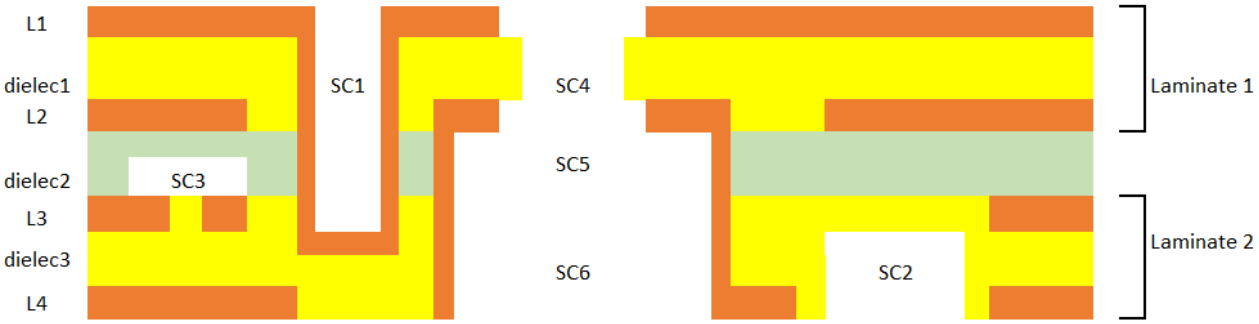
startCutLayer	qualifiedNameType	The unique name of the layer, from the list of `Layer` definitions, which is the first to be cut. It is the name of the layer in the final stackup, after all laminations processes are complete. If an inner layer is specified then sequential lamination is implied, with that layer being an outer layer of the layer subset to be cut.	1-1
direction	directionType	An optional enumerated string with values DOWN|UP that denotes the direction of the cut, Referring to Figure 2, DOWN means in the direction of moving from the primary (top) side of the board to the secondary (bottom) side of the board, and UP means in the opposite direction. If cutting more than one layer then the cut direction can be inferred from `startCutLayer`, but if cutting just one layer then `cutDirection` is required.	0-1
plusTol	nonnegativeDoubleType	The plus tolerance variation permitted from the nominal depth dimension	0-1
minusTol	nonnegativeDoubleType	The minus tolerance variation permitted from the nominal depth dimension	0-1

8.2.3.10.6.4 Fill



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Fill	FillType	An element that describes any full or partial fill material in the `SlotCavity`. The fill material is assumed to completely surround any component(s) in the `SlotCavity`.	0-1
depthRemaining	nonnegativeDoubleType	The depth measured (in design units) from the surface entrance of the `SlotCavity` to the uppermost surface of the fill material. A value of zero, or an absence of the attribute, indicates the `SlotCavity` is completely filled	0-1
matDes	qualifiedNameType	A reference to a fill material used that is defined in the `Bom` section	0-1
SpecRef	SpecRefType	The identification of a particular specification(s) from the SpecificationType that is able to provide additional details or instructions that apply to the Fill. For example a definition of the Fill's material, or instructions on how or when to apply the fill.	0-1

Examples



Cut To Depth Example

```
<SlotCavity name="SC1" platingStatus="PLATED">
      <Location x="345.200" y ="45.832"/>
      <Oval width="2.3" height="0.5"/>
 <MaterialCut depth="0.6" startCutLayer="TOP"  plusTol="0.06" minusTol="0.06"/>
</SlotCavity>
```

Cut To Thickness Examples

```
<SlotCavity name="SC2" platingStatus="NONPLATED" plusTol="0.03" minusTol="0.03">
      <Location x="0.0" y ="0.0"/>
      <Outline>
          <LineDesc lineEnd="ROUND" lineWidth="0.000000"/>
          <Polygon>
              <PolyBegin x="-0.0050" y="-0.2750"/>
              <PolyStepSegment x="1.0450" y="-0.2750"/>
              <PolyStepSegment x="1.0450" y="0.2750"/>
              <PolyStepSegment x="-0.0050" y="0.2750"/>
              <PolyStepSegment x="-0.0050" y="-0.2750"/>
           </Polygon>
      </Outline>
 <MaterialLeft thickness="2.5" startCutLayer="BOTTOM" plusTol="0.05" minusTol="0.05"/>
</SlotCavity>
```

Cut To Span Examples

Layer definition of Route_Laminate1:

```
      <Layer name="Route_Laminate1" layerFunction="ROUT" side="NONE" polarity="POSITIVE">
                <Span fromLayer="L1" toLayer="L2"/>
      </Layer>
```

Layer Feature defined on layer Route_Laminate1:

```
      <SlotCavity name="SC4" platingStatus="NONPLATED" plusTol="0.05" minusTol="0.05">
                <Location x="15.894" y ="18.713"/>
                <Circle diameter="2.5"/>
      </ SlotCavity >
```

Layer definition of Route_Laminate2:

```
      <Layer name="Route_Laminate2" layerFunction="ROUT" side="NONE" polarity="POSITIVE">
                <Span fromLayer="L3" toLayer="L4"/>
      </Layer>
```

Layer Feature defined on layer Route_Laminate2:

```
      < SlotCavity name="SC6" platingStatus="PLATED" plusTol="0.03" minusTol="0.03">
                 <Location x="15.894" y ="18.713"/>
                 <Circle diameter="5.0"/>
      </ SlotCavity>
```

Combination Example For Layer Dielec 2

Layer definition of Route_Dielec2:

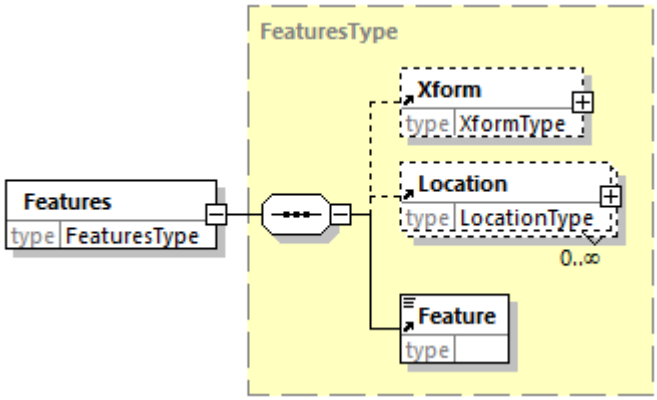
```
<Layer name=”Route_Dielec2” layerFunction=”ROUT” side=”NONE” polarity=”POSITIVE”>
        <Span fromLayer=”dielec2” toLayer=”dielec2”/>
</Layer>
```

Layer Features defined on layer Route_Dielec2:

```
< SlotCavity name=”SC5” platingStatus=”PLATED” plusTol=”0.03” minusTol=”0.03”>
        <Location x=”15.894” y =”18.713”/>
        <Circle diameter=”5.0”/>
</ SlotCavity>
<SlotCavity name=”SC3” platingStatus=”NONPLATED”>
        <Location x=”3.2” y =”2.6”/>
        <RectCenter width="2.3" height="1.5"/>
    <MaterialCut depth=”0.15” startCutLayer=”dielec2”  direction=”UP” plusTol=”0.02” minusTol=”0.02”/>
</SlotCavity>
```

8.2.3.10.7 Features

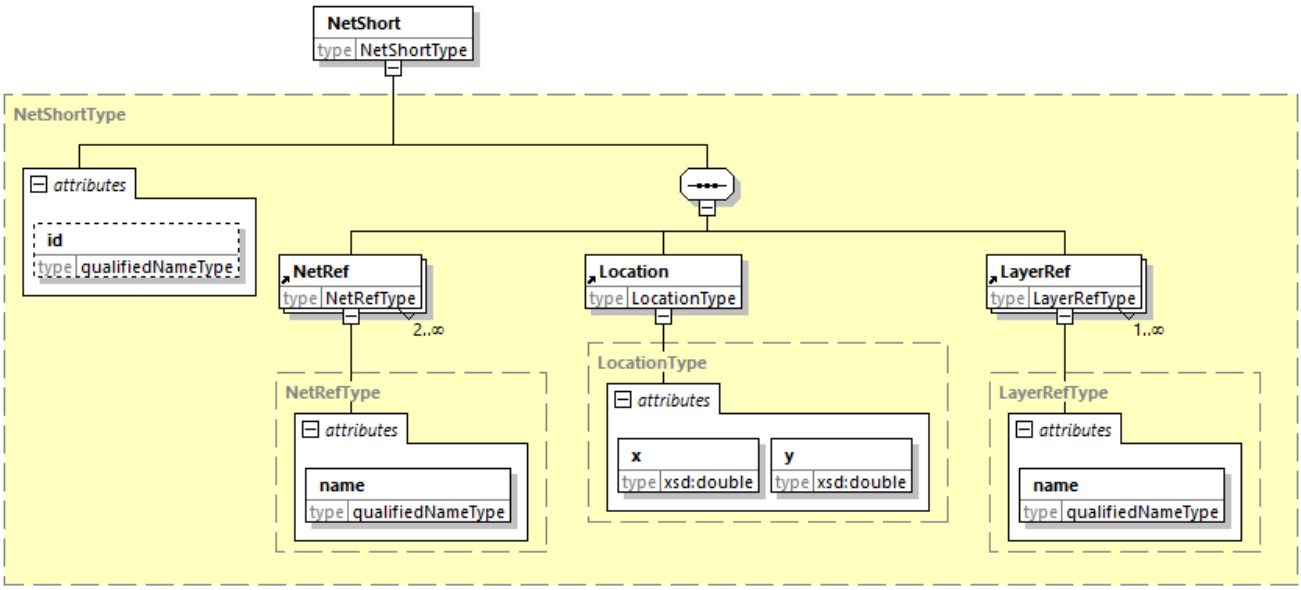
An embedded element that defines a substitution group, whose characteristics are used to identify any `StandardShape` or `UserShape`. The description may come from a predefined stored element contained in `DictionaryStandard` or `DictionaryUser` or instantiated at the time a feature is described.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Features	FeaturesType	An embedded element that defines a substitution group of any predefined `StandardShape` or `UserShape` that may be instantiated as a part of the `LayerFeature Set`.	0-n
Xform	XformType	An element that provides the ability to reset the point of origin of a predefined area that can be scaled, mirror imaged or rotated. The image may also by located by the `Xform`. See 3.3.	0-1
Location	LocationType	The location(s) of the `Feature` if it is a `StandardShape`. If the Feature has its own built in co-ordinates, such as `Line` or `Polyline` then a `Location` is not required, but then there can be only one instance.	0-n
Feature	ABSTRACT	An element that identifies the substitution group of any pre-defined `StandardShape` or `UserShape` that can be instanced by the user at the time the layer feature is described.	0-n

8.2.3.10.8 NetShort

An embedded element that defines an intentional physical and electrical short between two or more nets. Each `NetShort` represents a single refenced location, so that if the same nets are shorted in other locations then a `NetShort` element should be used to define each of those locations. A `LayerRef` child element defines the layer on which the short occurs. If the short is caused by a plated hole then it may span more than one layer, so each of those layers shall be referenced.

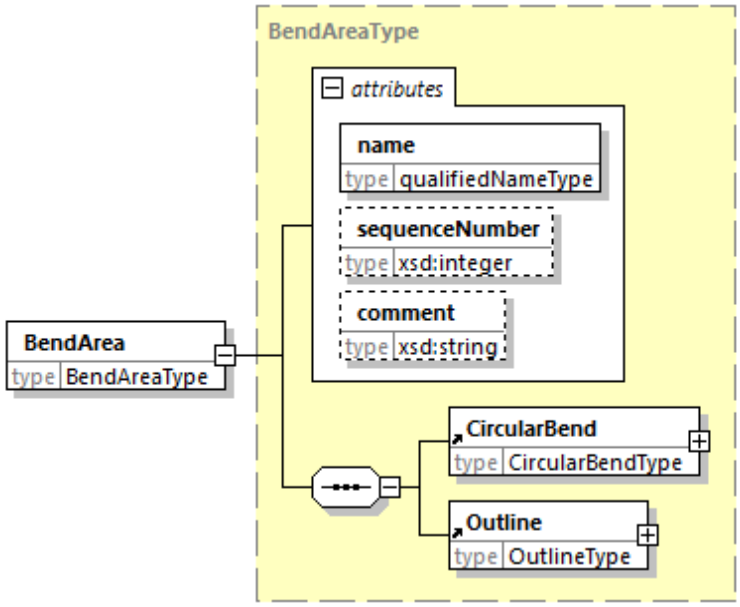


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
NetShort	NetShortType	The declaration of 2 or more nets that are intentionally shorted	0-n
id	qualifiedNameType	Optional unique identification of the net short, or collective/group name of the shorted nets	0-1
NetRef	NetRefType	The identification of a net that is shorted	2-n
name	qualifiedNameType	A reference to a unique net name in the design	1-1
Location	LocationType	A single x y location to identify where the short occurs, such as a point of contact along the touching surfaces of two layer features, or the center of a via. If the same nets are shorted at another location then another `NetShort` element should be used	1-1
LayerRef	LayerRefType	Reference to the layer or layers on which the short occurs. If the short is caused by a plated hole then it may span more than one layer	1-n

```
<NetShort id=”ShortedGrounds”>
   <NetRef name=”AGND”/>
   <NetRef name=”DGND”/>
   <NetRef name=”EARTH”/>
   <Location x=”123.45” y=”346.76”/>
   <LayerRef=”BOTTOM”/>
<NetShort>
```

8.2.3.11 BendArea

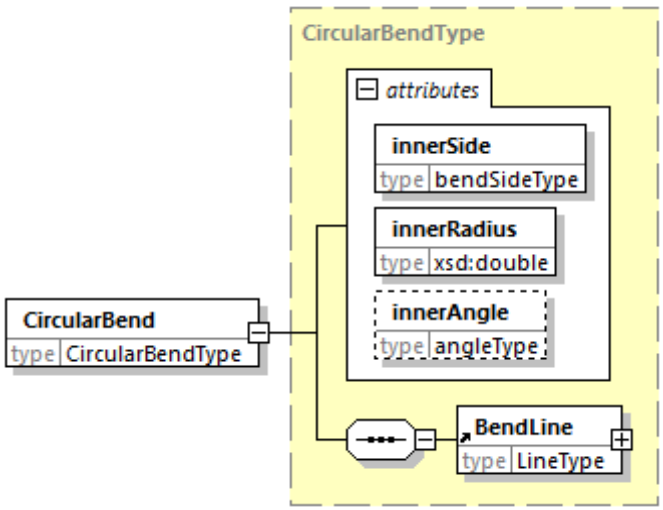
An embedded element that defines either a single bend, or one of a sequence of bends to a flex circuit.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
BendArea	BendAreaType	A nested element containing the different layer information represented in the CAD data base which describes the electronic assembly to be manufactured.	0-n
name	qualifiedNameType	The identification of the bend area. Each bend area must be a unique name.	1-1
sequenceNumber	integer	An integer for which order in sequence a bend should take place in reference to other bends.	0-1
comment	string	A description of the Bend Area as provided in the original CAD data	0-1
CircularBend	CircularBendType	An element that contains the bending information	1-1
Outline	OutlineType	A nested element that defines the physical outline of the bend area as viewed from the top.	1-1

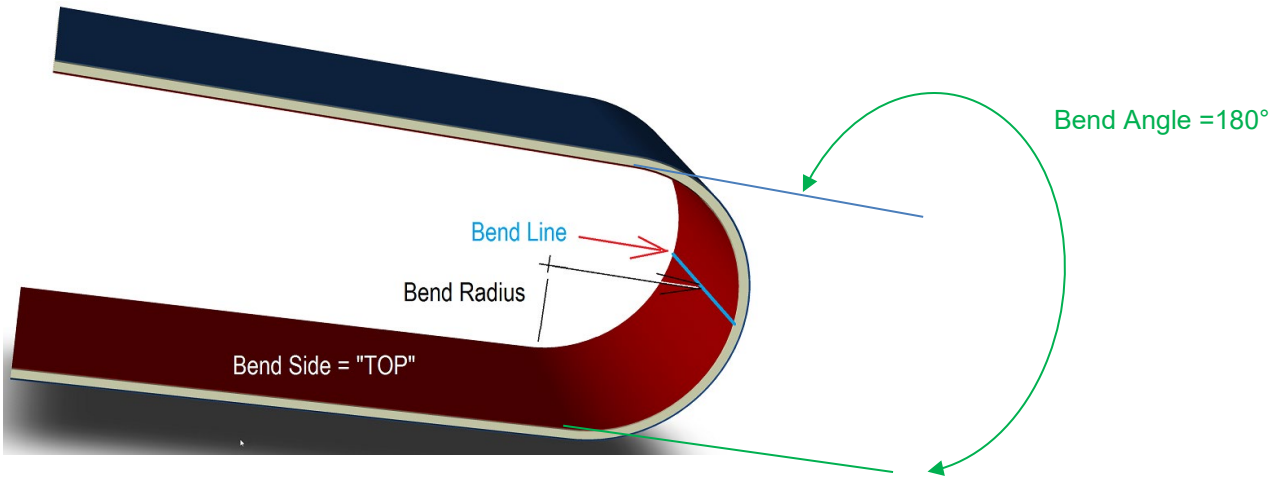
8.2.3.11.1 CircularBend

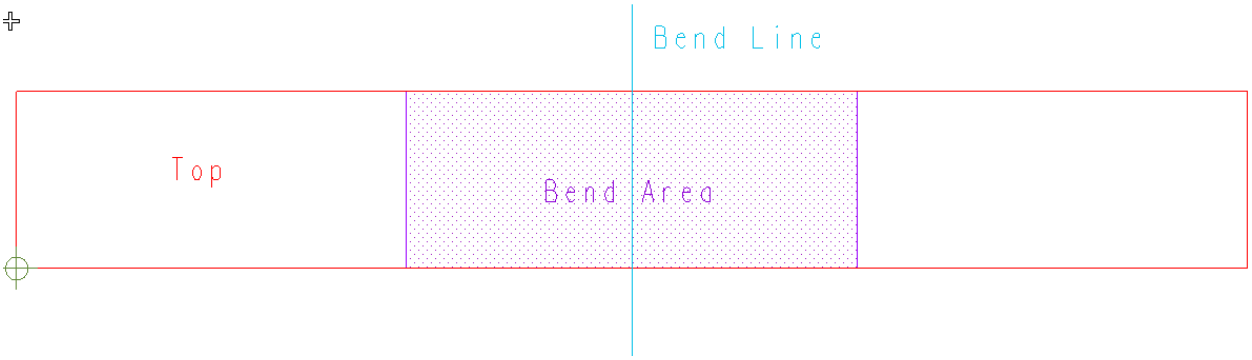
The bend type description for a bend that is a circular form.



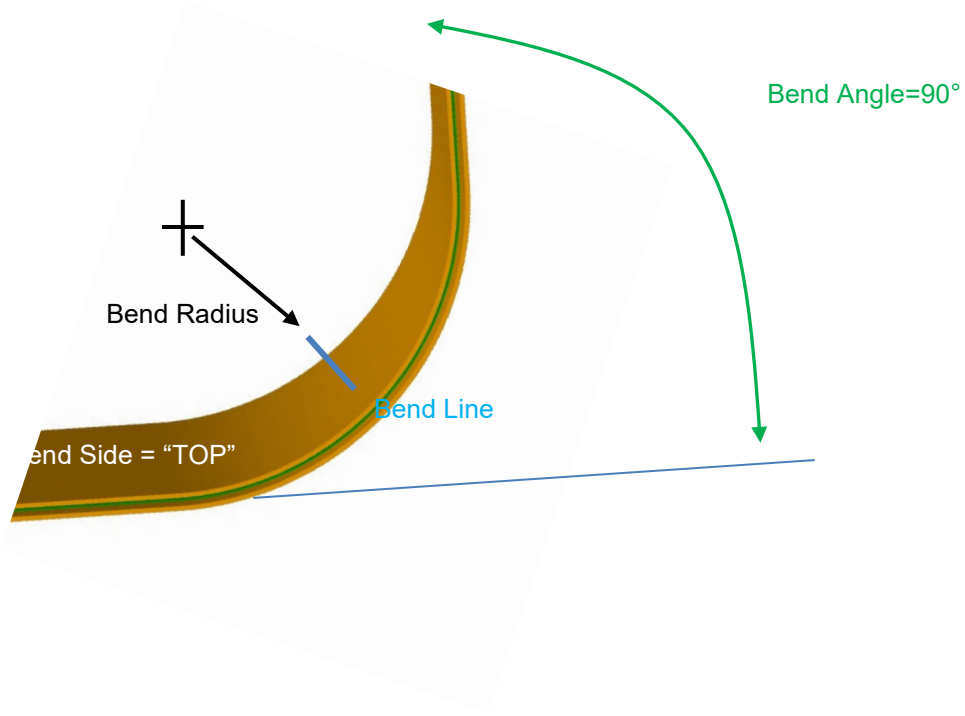
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
CircularBend	CircularBendType	The bend type description for a bend that is a circular form	0-1
innerSide	bendSideType	Side name for the inner side of the surface that is bending as viewed from the Top of the design. TOP indicate the outermost extents of the bend move upward. BOTTOM indicates the outermost extents of the bend move downward.	1-1
innerRadius	double	The radius of the bend	1-1
innerAngle	angleType	The final angle of the bend from the opposite side extents of the bend area outline and parallel to the bend line.	0-1
BendLine	LineType	The apex axis of the bend position on the inside surface of the bend.	1-1

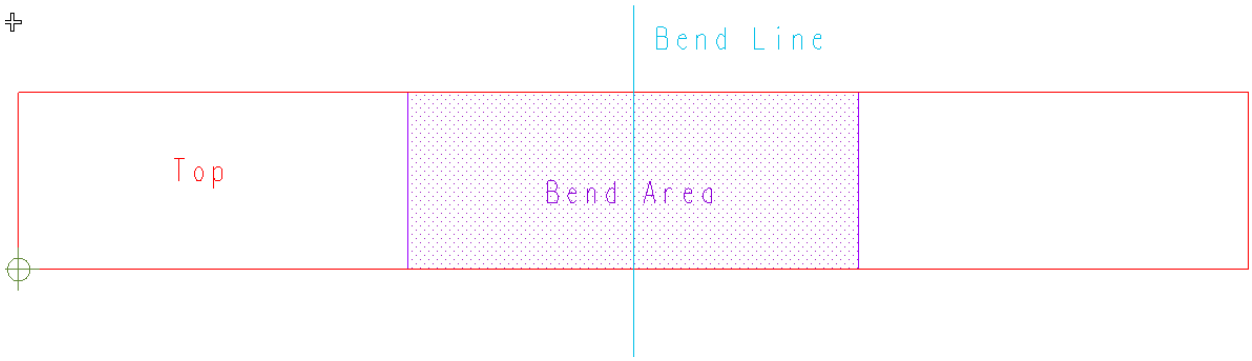
Examples:






```
<BendArea name="BEND_1" sequenceNumber="1" >
        <Outline>
           <LineDesc lineEnd="ROUND" lineWidth="0.000000"/>
            <Polygon>
               <PolyBegin x="443.551000" y="0.000000"/>
               <PolyStepSegment x="443.551 0000" y="200.000000"/>
               <PolyStepSegment x="956.488000" y="200.000000"/>
               <PolyStepSegment x="956.448000" y="0.000000"/>
               <PolyStepSegment x="443.551000" y="0.500000"/>
            </Polygon>
        </Outline>
        <CircularBend innerSide = “TOP” innerRadius = “500.000000” innerAngle=”180”>
            <BendLine startX=”700.000000” startY=”0.000000” endX=”700.000000” endY=”300.000000”>
                  <LineDesc lineEnd="ROUND" lineWidth="0.000000"/>
            </BendLine >
        </CircularBend>
 </BendArea>
```

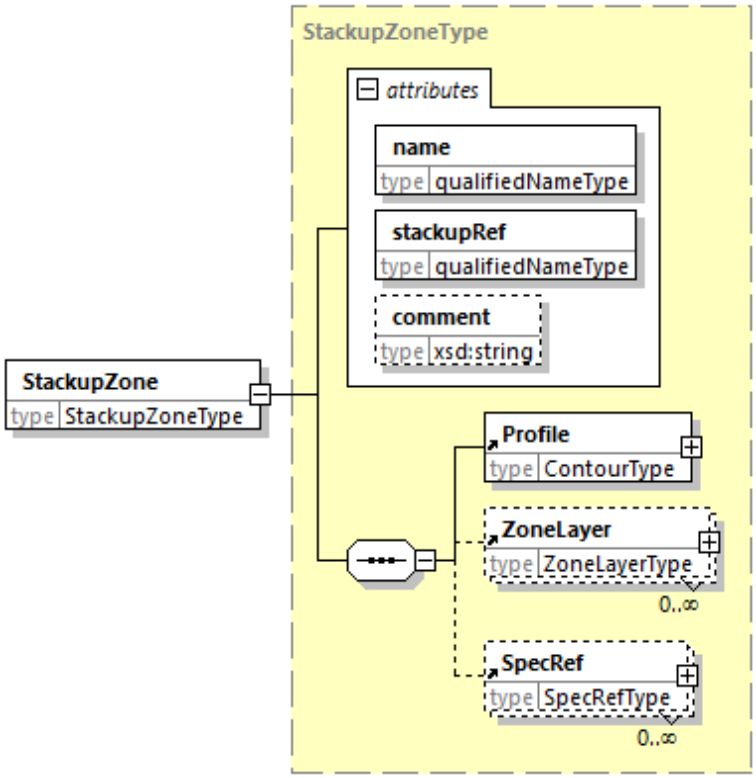





```
<BendArea name="BEND_2" sequenceNumber="2" >
      <Outline>
        <LineDesc lineEnd="ROUND" lineWidth="0.000000"/>
        <Polygon>
           <PolyBegin x="443.551000" y="0.000000"/>
           <PolyStepSegment x="443.551 0000" y="200.000000"/>
           <PolyStepSegment x="956.488000" y="200.000000"/>
           <PolyStepSegment x="956.448000" y="0.000000"/>
           <PolyStepSegment x="443.551000" y="0.500000"/>
         </Polygon>
        </Outline>
        <CircularBend innerSide = “TOP” innerRadius = “500.000000” innerAngle=”90”>
  <BendLine startX=”700.000000” startY=”0.000000” endX=”700.000000” endY=”300.000000”>
     <LineDesc lineEnd="ROUND" lineWidth="0.000000"/>
  </ BendLine >
        </CircularBend>
      </BendArea>
```

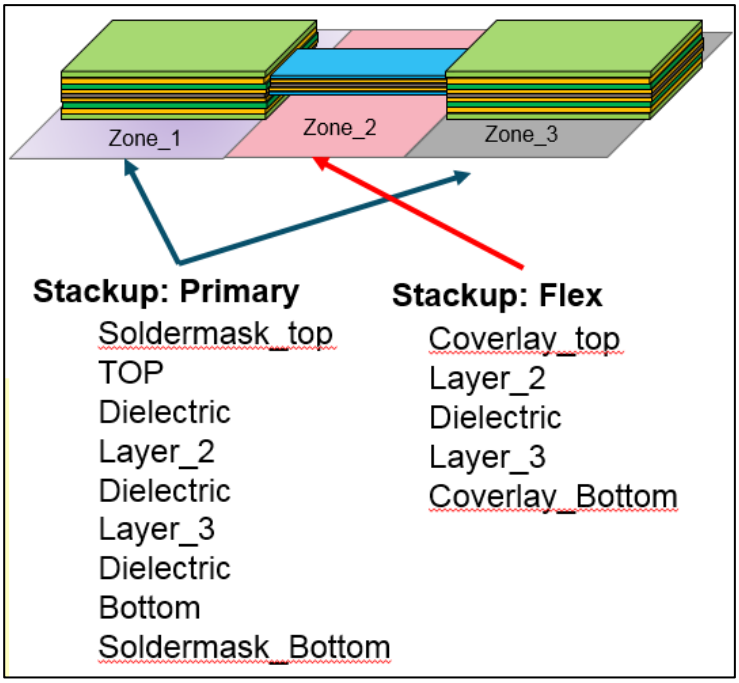
8.2.3.12 StackupZone

A nested element defining stackup zones in a rigid flex design. The zones are created by planes ‘slicing’ vertically, that is perpendicular, to the plane of the board. They occur at the boundary of a stackup change between rigid and flex sections of the board.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
StackupZone	StackupZoneType	The definition of the area and properties of a stackup zone	1-n
name	qualifiedNameType	If more than one stack up zone, it is recommended to have a unique name for each stackup zone. Duplicate names are not allowed.	0-1
stackupRef	qualifiedNameType	A reference to the stackup layer structure that is contained within this zone structure	1-1
Profile	ContourType	The profile of the stackup zone.	1-1
SpecRef	SpecRefType	A reference to flex or other properties of the zone	0-1

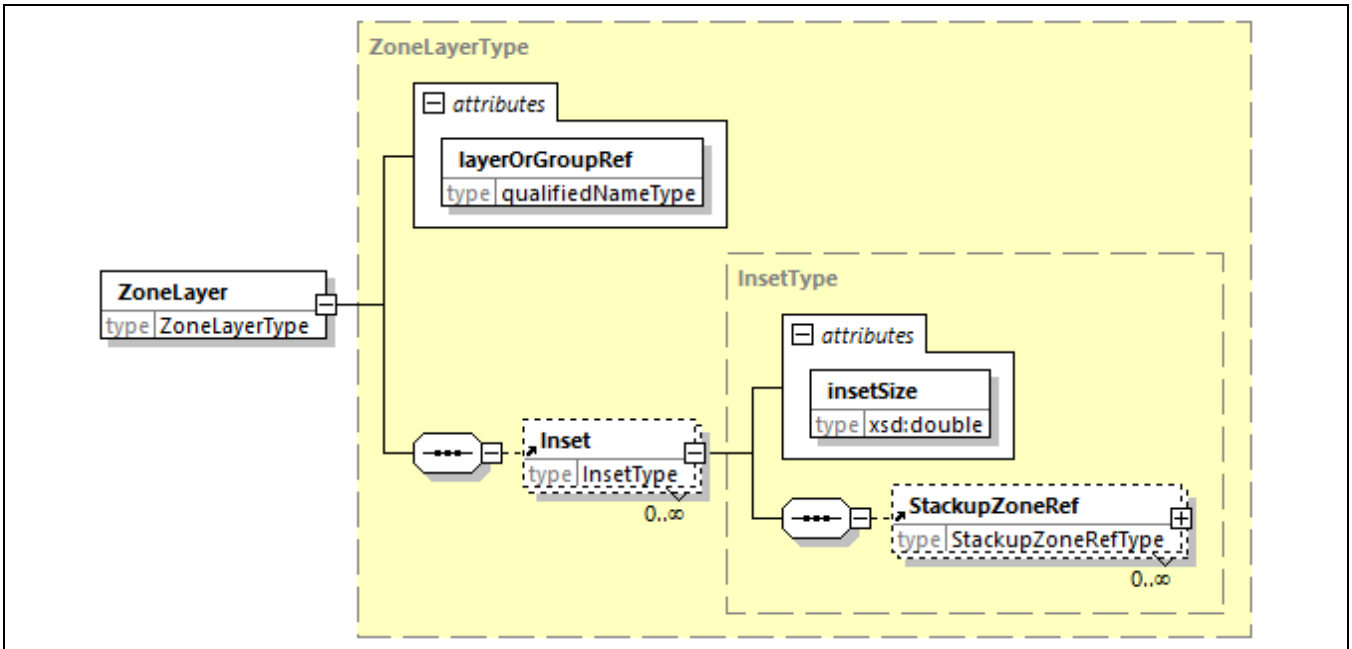
Example:




```
<StackupZone name="Zone_1">
        <Profile>
            <Polygon>
   <PolyBegin x= "-.250" y="-.250"/>
   <PolyStepSegment x= "2.750 y= ".-250"/>
   <PolyStepSegment x= "2.750" y= ".2.750"/>
   <PolyStepSegment x= "-.250" y= ".2.750"/>
   <PolyStepSegment x= "-.250" y= "-250"/>
            </Polygon>
          </Profile>
         <StackupRef name="Primary"/>
</StackupZone>

<StackupZone name="Zone_2">
        <Profile>
            <Polygon>
   <PolyBegin x= "2.750" y="0.000"/>
   <PolyStepSegment x= "5.750 y= "0.000"/>
   <PolyStepSegment x= "5.750" y= "2.000"/>
   <PolyStepSegment x= "2.750" y= "2.000"/>
   <PolyStepSegment x= "2.750" y= "0.000"/>
            </Polygon>
          </Profile>
         <ZoneLayer layerRef="CoverLay" InsetSize="0.010">
          <StackupRef name="Flex"/>
</StackupZone>
<StackupZone name="Zone_1">
        <Profile>
            <Polygon>
   <PolyBegin x= "5.750" y="-.250"/>
   <PolyStepSegment x= "7.750 y= ".-250"/>
   <PolyStepSegment x= "7.750" y= ".2.750"/>
   <PolyStepSegment x= "5.750" y= ".2.750"/>
   <PolyStepSegment x= "5.750" y= "-250"/>
            </Polygon>
          </Profile>
         <StackupRef name="Primary"/>
</StackupZone>
```

8.2.3.12.1 ZoneLayer

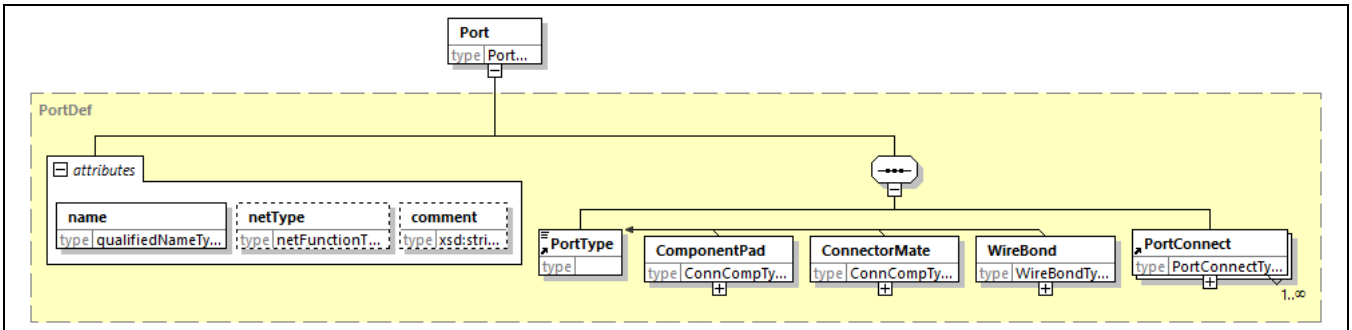
Layer specific attributes for a stackup layer in a zone.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
ZoneLayer	ZoneLayerType	Layer specific attributes for a stackup layer in a zone	0-1
layerOrGroupRef	qualifiedNameType	A reference to a layer in the stackup layer structure that is contained within this zone structure.	1-1
InsetSize	Double	The dimension of the inset oversize from the coverlay or other flexible layer into the rigid zone. InsetSize is intended to be used for enlarging a flexible zone where it is inserted into an adjacent rigid zone.	1-1
StackupZoneRef	StackupZoneRefType	The names of adjacent zones to which the inset zone applies. This allows different inset values for different adjacent rigid zones. If no stackup zone references are specified then the InsetSize attribute is applied to all adjacent rigid zones for the specified flexible zone.	0-n

8.2.3.13 Port

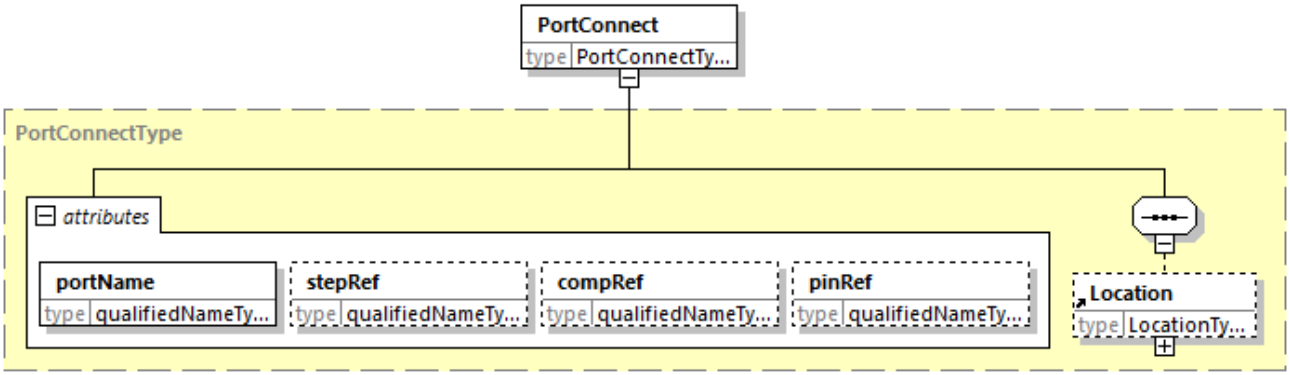
An embedded element to describe a physical and electrical connection between a point within the board `Step` and a point outside the board `Step`. The destination point could be within another Step in the same file, which represents a daughter board or a custom IC, or it could be a pin/pad of an off-the-shelf component. The port represents one of three types of connection: a wire bond, a connector pin mating, or a custom component pin to pad direct connection when the custom component is descried in another Step in the same file.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Port	PortType	An element that signifies that the PhyNetPoint is physically and electrically connected to another object, such as a component by wire bond, or a daughter card by connector	0-n
name	qualifiedNameType	A unique name assigned to the `Port`	1-1
netType	netFunctionType	An enumerated list to specify the direction (or other function) of the electrical signal carried by the port, in relation to the `Step` that the port is in. Possible values are INPUT | OUTPUT | BIDIRECTIONAL | VOLTAGE_REF | UNSPECIFIED | NO_CONNECT | OTHER	0-1
PortType	ABSTRACT	A substitution group that is replaced by one of 3 elements to define the port type: WireBond – the port represents a wire bond that connects from the board to the pin of a component, which is either custom (defined in its own `Step`) or off the shelf ConnectorMate, - the port represents the mating of pins between the connector on the board and the connector on another board (`Step`) ComponentPad – the port represents the connection between the board and the component pad of a custom IC which is defined in its own `Step`	1-1
comment	string	Text that describes a netType of OTHER or any other details about the port	0-1

8.2.3.13.1 PortConnect

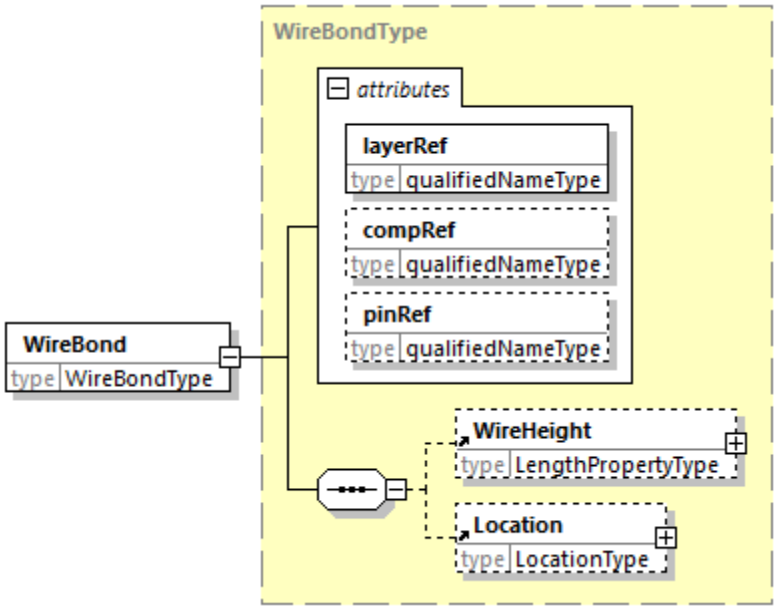
A nested element that defines the end point of a physical connection that the port represents.



PortConnect	PortConnectType	An element that signifies the end point of a physical connection that the port represents. There can be multiple end points for a single start point	1-n
name	qualifiedNameType	A unique name assigned to the `PortConnect`	1-1
stepRef	qualifiedNameType	An optional reference if the port connects to another `Step`, representing another board or custom IC	0-1
compRef	qualifiedNameType	Reference to any component that is instantiated on the `Step` being connected to	0-1
pinRef	qualifiedNameType	Reference to any component that is instantiated on the `Step` being connected to	0-1
Location	LocationType	X y coordinates of the end point of the port. If the end point is in another `Step` then the coordinates shall be relative to that `Step`. This location **shall** match the location of a pad in the `PhyNe`t section that has a `PortRef` attribute with a value equal to the name of this Port	0-1

8.2.3.13.2 WireBond

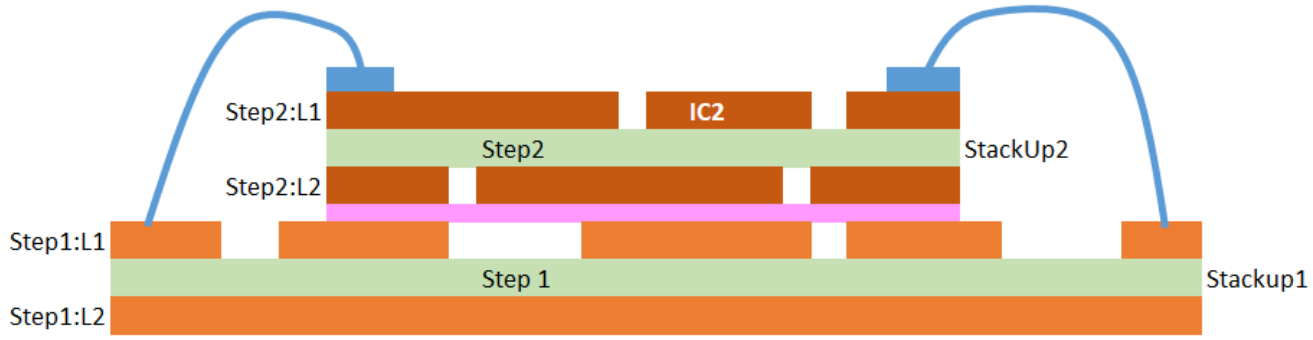
A nested element that defines a wire bond.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence

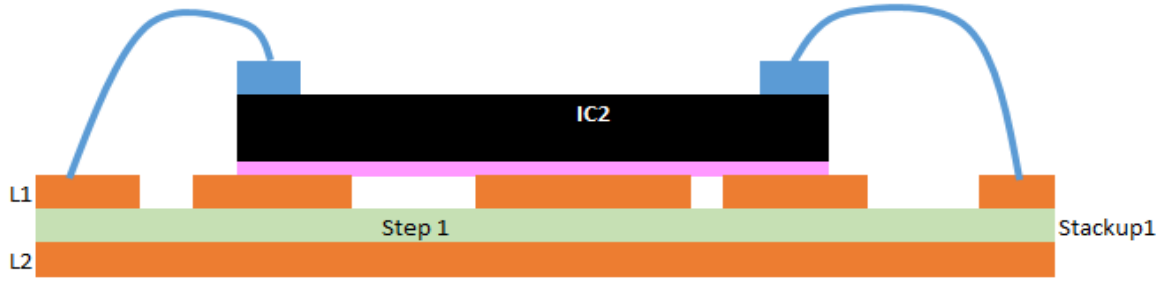
WireBond	WireBondType	The definition of a wire bond	1-1
layerRef	qualifiedNameType	A reference to the layer that the wire bond starts from	1-1
compRef	qualifiedNameType	Reference to a component that is instantiated in the current `Step` which relates to the port	0-1
pinRef	qualifiedNameType	Reference to any component pin that is instantiated in the current `Step` which relates to the port	0-1
WireHeight	LengthPropertyType	The height relative to the parent board (Step) which defines a region that the bonding wire is restricted to. The attribute `constraintType` of `LengthPropertyType` can be set to indicate a MAX or MIN value	0-1
Location	LocationType	X y coordinates of the start point of the port. This location **shall** match the location of a pad in the `PhyNe`t section that has a `PortRef` attribute with a value equal to the name of this Port	0-1

8.2.3.13.2.1 Custom IC Example




```
<Step name="Step1" type=”BOARD” stackupRef=”Stackup1”>
 <Component refDes="IC2" packageRef="Cust2" layerRef="Step1:L1" part="W54" mountType="WIRE_BONDED">
     <Location x="320.0" y="184.0" />
 </Component>
   <Port name=”IC2_TX” netType=”OUTPUT”>
     <WireBond compRef=”IC2” pinRef=”A1>
        <WireHeight=”6” unit=”mm” tolPlus=”0.5” constraintType=”MAX”/>
        <Location x=”315.0” y=”184.0”/>
     <WireBond/>
     <PortConnect portName=”TX” stepRef=”Step2” pinRef=”A1”/>
   </Port>
   <Port name=”IC2_RX” netType=”INPUT”
     <WireBond compRef=”IC2” pinRef=”X7”>
        <WireHeight=”6” unit=”mm” tolPlus=”0.5” constraintType=”MAX”/>
        <Location x=”334.0” y=”184.0”/>
     <WireBond/>
     <PortConnect portName=”RX” stepRef=”Step2” pinRef=”X7”/>
    </Port>

<Step name="Step2" type=”IC” stackupRef=”Stackup2”>
   <Port name=”TX” netType=”INPUT”>
     <WireBond pinRef=”A1”>
         <Location x=”1.00” y=”1.00”/>
     <WireBond/>
  <PortConnect portName=”IC2_TX” stepRef=”Step1” compRef=”IC2” pinRef=”A1”/>
   </Port>
   <Port name=”RX” netType=”OUTPUT”
     <WireBond pinRef=”X7”>
         <Location x=”50.00” y=”1.00”/>
     <WireBond/>
     <PortConnect portName=”IC2_X7” stepRef=”Step1” compRef=”IC2” pinRef=”X7”/>
      </Port>
</Step>
```

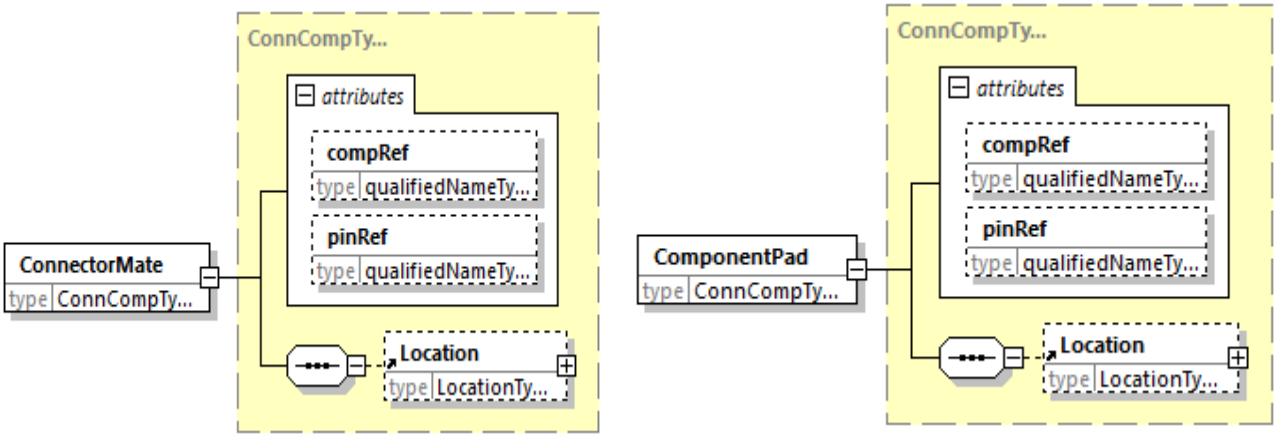
8.2.3.13.2.2 Off-The-Shelf Component Example




```
<Step name="Step1" type=”BOARD” stackupRef=”Stackup1”>
 <Component refDes="IC2" packageRef="Cust2" layerRef="L1" part="ABC" mountType="WIRE_BONDED">
      <Location x="320.0" y="184.0" />
  </Component>
   <Port name=”IC2_TX” netType=”OUTPUT”>
     <WireBond compRef=”IC2” pinRef=”A1>
        <WireHeight=”6” unit=”mm” constraintType=”MAX”/>
        <Location x=”315.0” y=”184.0”/>
     <WireBond/>
   <PortConnect portName=”TX” pinRef=”A1”/>
    </Port>
    <Port name=”IC2_RX” netType=”INPUT”
     <WireBond compRef=”IC2” pinRef=”X7”>
        <WireHeight=”6” unit=”mm” constraintType=”MAX”/>
        <Location x=”325.0” y=”184.0”/>
     <WireBond/>
     <PortConnect portName=”RX” pinRef=”X7”/>
      </Port>
</Step>
```

8.2.3.13.3 ConnectorMate & ComponentPad

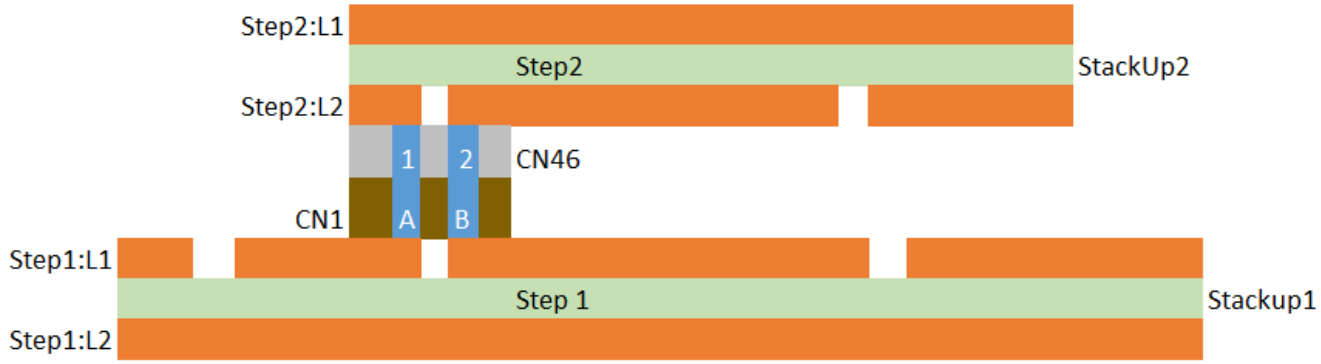
Nested elements `ConnectorMate`, for describing a port between two connectors, and `ComponentPad`, for describing a port between a custom IC pin and a pad on the board, share the same Element Type `ConnCompType`.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
ConnectorMate / ComponentPad	ConnCompType	The definition of a port formed by mating connectors, or by a pin pad of a custom IC that is defined by another `Step`	1-1
stepRef	qualifiedNameType	An optional reference if the port connects to another `Step`, representing another board or custom IC	0-1

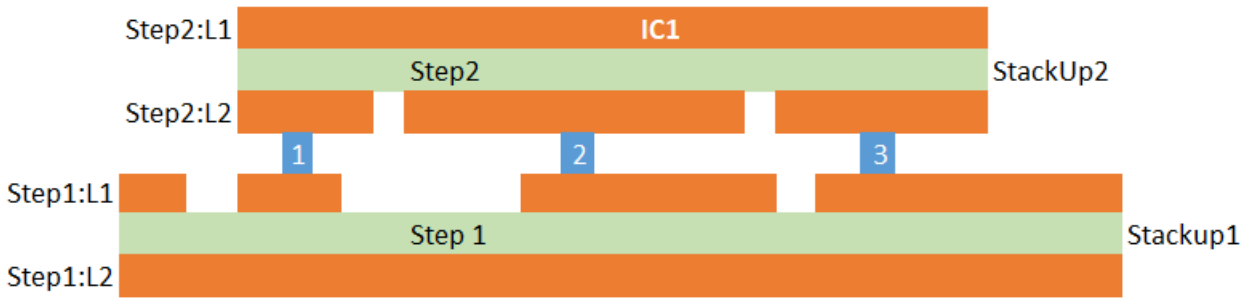
compRef	qualifiedNameType	Reference to any component that is instantiated on the `Step` being connected to	0-1
Location	LocationType	X y coordinates of the start point of the port. This location **shall** match the location of a pad in the `PhyNet` section that has a `PortRef` attribute with a value equal to the name of this Port	0-1

Connector Example




```
<Step name="Step1" type=”BOARD” stackupRef=”Stackup1”>
   <Component refDes="CN1" packageRef="PinArray" layerRef="Step1:L1" part="CDE" mountType="SMT">
              <Location x="10.0" y="30.0" />
   </Component>
   <Port name=”SUB_PWR” netType=”POWER”>
            <ConnectorMate compRef=”CN1” pinRef=”A”/>
           <PortConnect portName=”PWR” stepRef=”Step2” compRef=”CN46” pinRef=”1”/>
   </Port>
   <Port name=”SUB_GND” netType=”POWER”>
            <ConnectorMate compRef=”CN1” pinRef=”A”/>
            <PortConnect portName=”GND” stepRef=”Step2” compRef=”CN46” pinRef=”2”/>
    </Port>
</Step>

<Step name="Step2" type=”BOARD” stackupRef=”Stackup2”>
   <Component refDes="CN46" packageRef="Socket" layerRef="L4" part="XYZ" mountType="SMT">
              <Xform mirror="true"/>
              <Location x="1.0" y="1.0"/>
   </Component>
   <Port name=”PWR” netType=”POWER”>
            <ConnectorMate compRef=”CN46” pinRef=”1”>
           <PortConnect portName=”SUB_PWR” stepRef=”Step1” compRef=”CN1” pinRef=”A”/>
   </Port>
   <Port name=”GND” netType=”POWER”>
            <ConnectorMate compRef=”CN46” pinRef=”1”>
           <PortConnect portName=”SUB_GND” stepRef=”Step1” compRef=”CN1” pinRef=”B”/>
    </Port>
</Step>
```

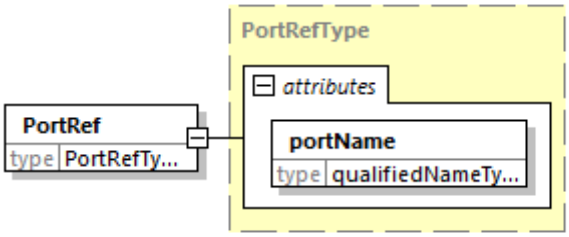
Custom IC Example




```
<Step name="Step1" type=”BOARD” stackupRef=”Stackup1”>
  <Component refDes="IC1" packageRef="CustomPackageName" layerRef="Step1:L1" part="ABC" mountType="SMT">
            <Location x="105.0" y="105.0" />
  </Component>
  <Port name=”IC1_EN” netType=”OUTPUT”>
        <ComponentPad compRef=”IC1” pinRef=”1>
              <Location x=”100.00 y=”100.00”/>
        <ComponentPad/>
        <PortConnect portName=”ENABLE” stepRef=”Step2” pinRef=”1”/>
  </Port>
  <Port name=”IC1_A1” netType=”OUTPUT”>
        <ComponentPad compRef=”IC1” pinRef=”2>
              <Location x=”105.00 y=”100.00”/>
        <ComponentPad/>
        <PortConnect portName=”ADDR1” stepRef=”Step2” pinRef=”2”/>
    </Port>
    <Port name=”IC1_D1” netType=”INPUT”>
        <ComponentPad compRef=”IC1” pinRef=”3>
              <Location x=”110.00 y=”100.00”/>
        <ComponentPad/>
        <PortConnect portName=”DATA1” stepRef=”Step2” pinRef=”3”/>
    </Port
</Step>


<Step name="Step2" type=”IC” stackupRef=”Stackup2”>
  <Port name=”ENABLE” netType=”INPUT”>
        <ComponentPad pinRef=”1>
              <Location x=”0.00 y=”0.00”/>
        <ComponentPad/>
        <PortConnect portName=”IC1_EN” stepRef=”Step1” compRef=”IC1” pinRef=”1”/>
  </Port>
  <Port name=”ADDR1” netType=”INPUT”>
        <ComponentPad pinRef=”2>
              <Location x=”5.00 y=”0.00”/>
        <ComponentPad/>
        <PortConnect portName=”IC1_A1” stepRef=”Step1” compRef=”IC1” pinRef=”2”/>
    </Port>
  <Port name=”DATA1” netType=”INPUT” type=”COMPONENT”>
        <ComponentPad pinRef=”3>
              <Location x=”10.00 y=”0.00”/>
        <ComponentPad/>
        <PortConnect portName=”IC_D1” stepRef=”Step1” compRef=”IC1” pinRef=”3”/>
   </Port>
</Step>
```

8.2.3.13.4 PortRef

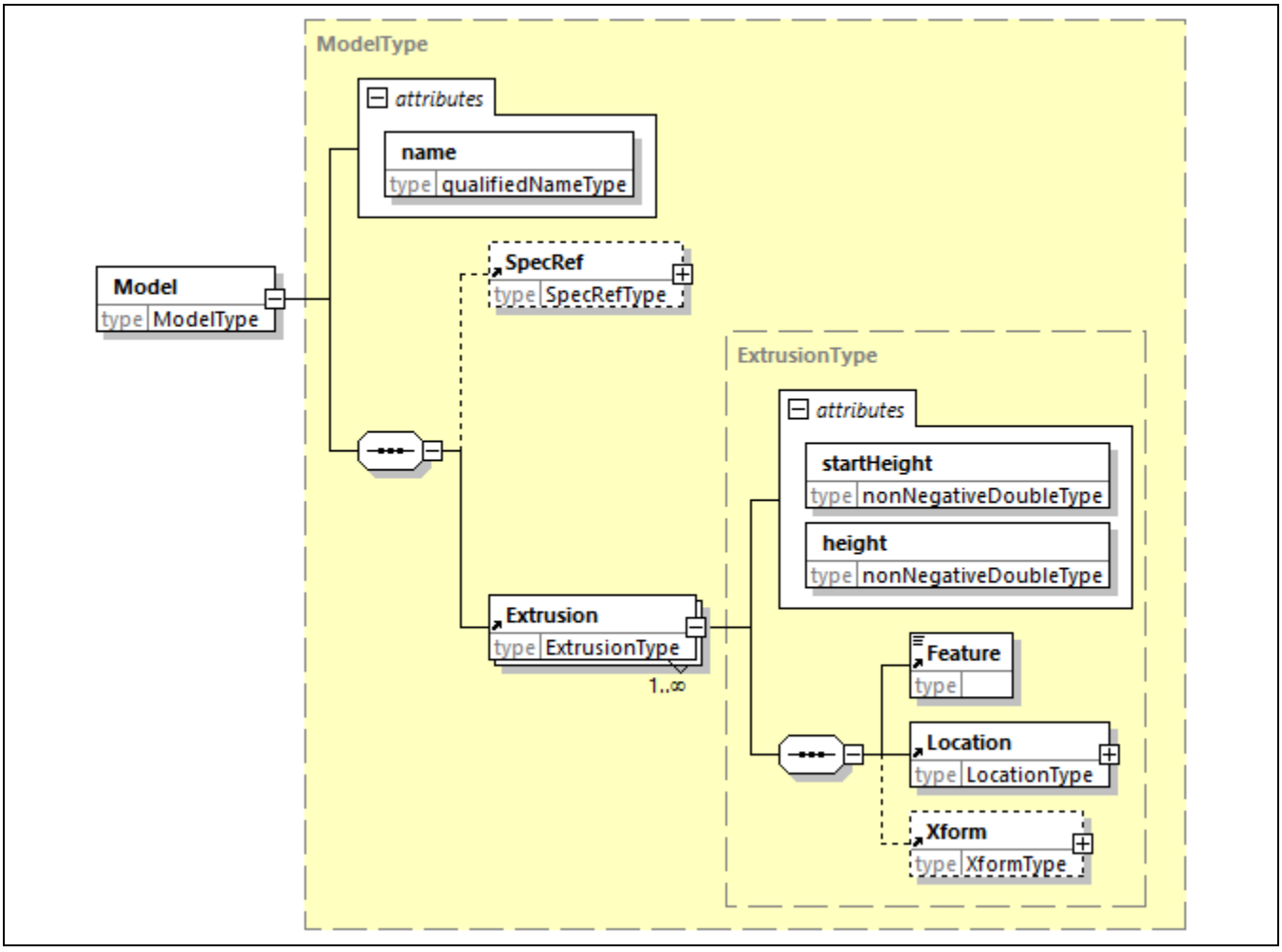
A nested element to reference a port from another element.



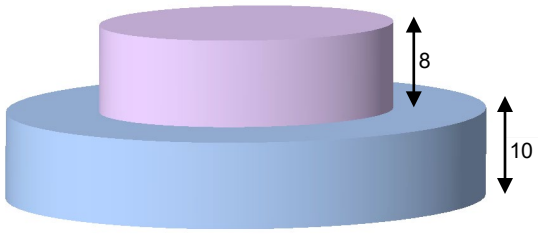
Attribute / Element Name	Attribute / Element Type	Description	Occurrence
PortRef	PortRefType	A nested element to reference a port from another element.	1-1
portName	qualifiedNameType	A reference to a predefined port name	1-1

8.2.3.14 Model

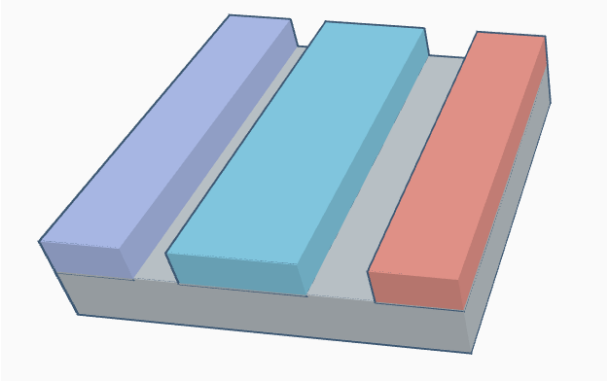
A 3D definition of an item, consisting of the combination of one or more 3D shapes. The origin of the `Model` definition is the placement origin in the `Step` it is instantiated in. All units of length are per `CadHeader` units.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Model	ModelType	A 3D definition of an item, consisting of the combination of one or more 3D shapes. The origin of the `Model` definition is the placement origin in the `Step` it is instantiated in. All units of length are per `CadHeader` units.	0-n
name	qualifiedNameType	A unique name for the model	1-1
SpecRef	SpecRefType	Reference to a unique specification that defines the Model's material and/or other properties. Alternatively if the model is referenced by a `modelRef` attribute of a `Component` instance then that `Component` could have a `matDes` attribute that references a `MatDes` child of a `BomItem` which has an overarching `SpecRef`.	0-1
Extrusion	ExtrusionType	A nested element that describes a simple 3D shape, that is formed by extruding a given 2D shape to a given height	1-n
startingHeight	nonNegativeDoubleType	The starting height of the extrusion, relative to zero height of the Model definition. The `startingHeight` **shall** be either zero (as the first shape) or equal to the `height` value of another shape, such that the shape is not floating in mid air.	1-1
height	nonNegativeDoubleType	The height to which the 2D shape is extruded above the `startingHeight`	1-1
Feature	ABSTRACT	A substitution group that can be a StandardShape or UserShape	1-1
Location	LocationType	A nested element with attributes x & y to define the location of the `Feature` in the local coordinate system of the model (relative to the origin of the Model)	1-1
Xform	Xform type	An optional nested element that can define a rotation or scale in the XY plane of the 2D shape to be extruded.	0-1




```
<Model name="coin">
  <SpecRef id="CoinMaterial"/>
  <Extrusion startingHeight="0' height="10">
      <Ellipse width="50" height="25"/>
      <Location x="0.00" y="0.00"/>
  </Extrusion>
  <Extrusion startingHeight="10' height="8">
      <Ellipse width="30" height="15"/>
      <Location x="0.00" y="0.00"/>
  </Extrusion>
</Model>
```



```
<Model name="heatsink">
  <SpecRef id="Copper"/>
  <Extrusion startingHeight="0' height="10">
      <RectCenter width="50" height="50"/>
      <Location x="0.00" y="0.00"/>
  </Extrusion>
  <Extrusion startingHeight="10' height="8">
      <RectCenter width="10" height="50"/>
      <Location x="5.00" y="0.00"/>
  </Extrusion>
  <Extrusion startingHeight="10' height="8">
      <RectCenter width="20" height="50"/>
      <Location x="24.00" y="0.00"/>
  </Extrusion>
  <Extrusion startingHeight="10' height="8">
      <RecCenter width="10" height="50"/>
      <Location x="45.00" y="0.00"/>
  </Extrusion>
</Model>
```

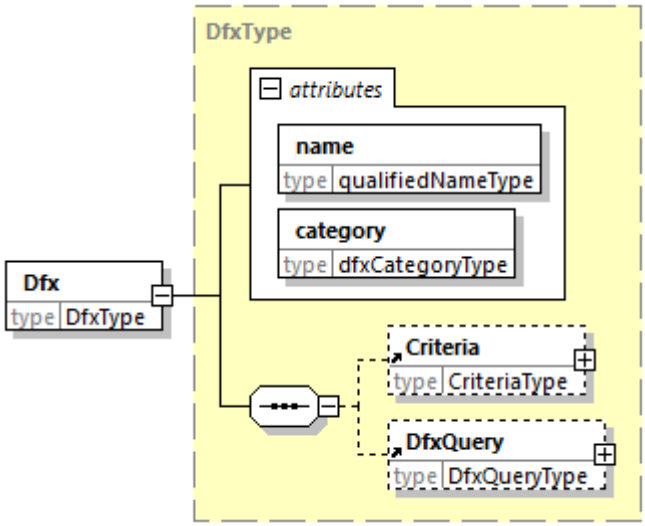
8.2.3.15 Dfx

Many design and manufacturing tools have the ability to analyze the details of a data file and make a determination as to whether all the design rules have been met or if the parts are manufacturable within the capability of the board fabricator or assembler. The results of these analyses need to be retained so that future users of the data contained in the IPC-2581 file are aware of the improvements or risks which are apparent within the manufacturing domain.

The `Dfx` element consists of a variety of criteria and queries. Criteria can also include DfxMeasurement elements indicating locations which violate the criteria properties. Each of these lists is identified separately so that the design for manufacturing analysis can be grouped according to their particular characteristic. The granularity of this grouping is dependent on the desire of the designer, fabricator, or assembler to capture the details of the Dfx analysis.

Just as there may be several `Dfx` elements, each such element may contain many measurements (`DfxMeasurement`) that are described in order to indicate where the conditions afford a risk or need improvement.

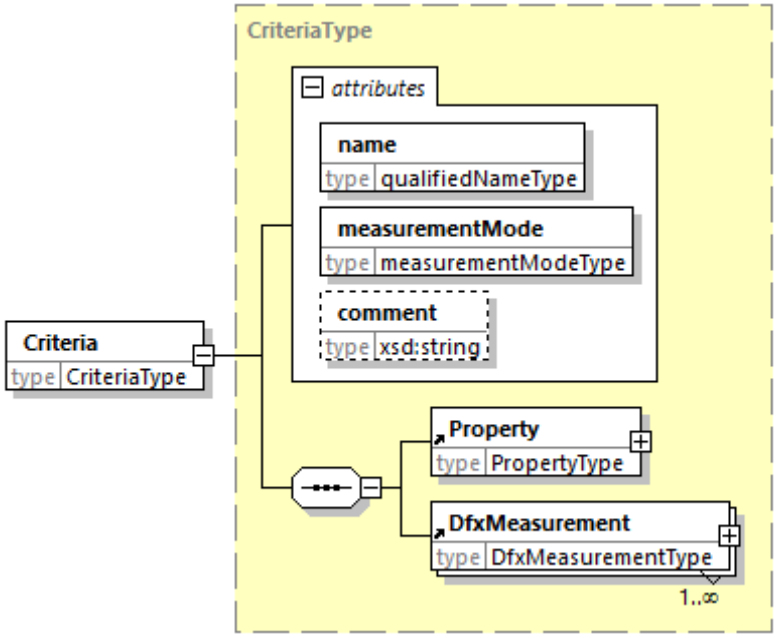
Since Dfx is designed for bidirectional communications, the DfxQuery element allows for one or more DfxResponses to the query or the DfxMeasurements listed under Criteria. This allows a vendor to indicate the items of concern and allows the OEM to respond to each of those items individually or group all items in a Dfx list under one response.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Dfx	DfxType	An element consisting of a `Critera` for the design and/or a `DfxQuery` relating to the design, with reference to a specific category.	0-n
name	qualifiedNameType	A unique name assigned to the `Dfx` element.	1-1
category	dfxCategoryType	A unique description defined by the user which relates to the products contained in the particular IPC-2581 file intended to provide identification for the group of the `DfxMeasurement` characteristics. The category type is an enumerated string that may be any of the following groupings COMPONENT | BOARDFAB | ASSEMBLY | STACKUP | TESTING | DATAQUALITY	1-1
Criteria	CriteraType	An element used in defining the criteria used in evaluating the measurement characteristics in order to determine whether the design as supplied by the user meets the manufacturing criteria.	0-1
DfxQuery	DfxQueryType	An element used to ask any question about the design, whether related to data in the IPC-2581 file or not. The question may refer to a `DfxMeasurement`, but if not then the Criteria element need not be present.	0-1

8.2.3.15.1 Criteria

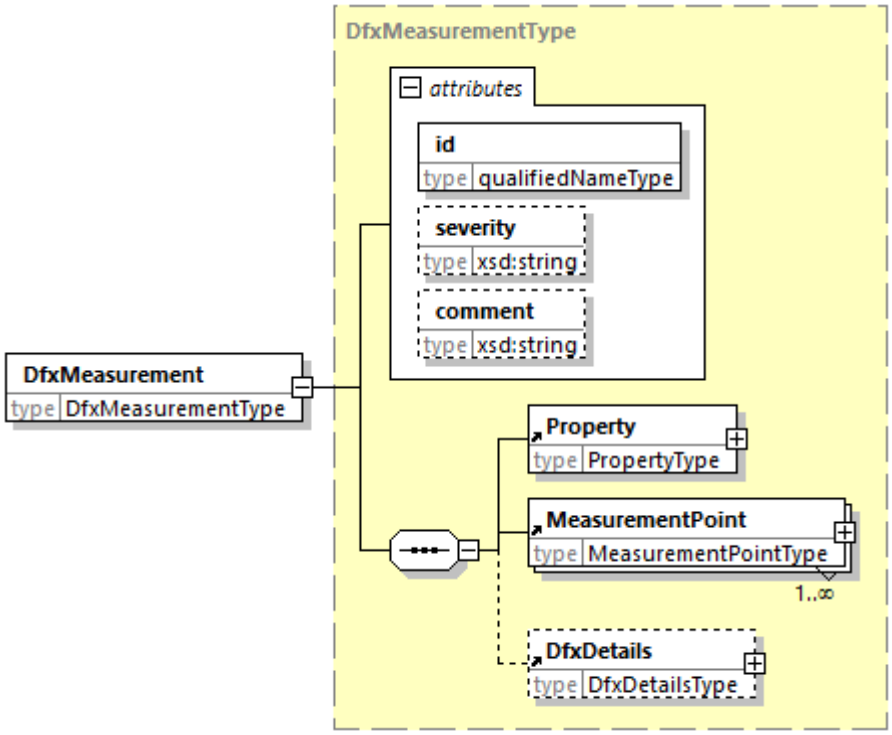
The purpose of this element is to define the measurement criteria that was used for DFx Analysis and to list the measurements that violate the defined criteria. The criteria may range from any design for manufacturing validation rules, process rules, note details, or instruction details. The Property element is used as the descriptor for the values.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Criteria	CriteriaType	An element consisting of the definition of a particular requirement of the product, and a list of all the measurements made in the design for manufacturing analysis that do not meet that requirement.	1-1
name	qualifiedNameType	A descriptive name that describes the specific measurement or detail	1-1
measurementMode	measurementModeType	The mode of measurement depending on the characteristic being evaluated, indicated as an enumerated string using: DISTANCE | AREA | RESISTANCE | CAPACITANCE | IMPEDANCE | INDUCTANCE | VOLTAGE_DC | VOLTAGE_AC | FREQUENCY | CURRENT | POWER | PERCENTAGE | SIZE | NONE | TEXT | TOLERANCE | OTHER	1-1
comment	string	Any comment to assist in the definition or description of the unit details. Mandatory when `measurementMode` value is the enumerated type OTHER	0-1
Property	PropertyType	An element that provides the definition of the measurement details as the base definition	1-1
DfxMeasurement	DfxMeasurementType	An element consisting of a measurement made in the design which violates its parent `Criteria`	1-n

8.2.3.15.1.1 DfxMeasurement

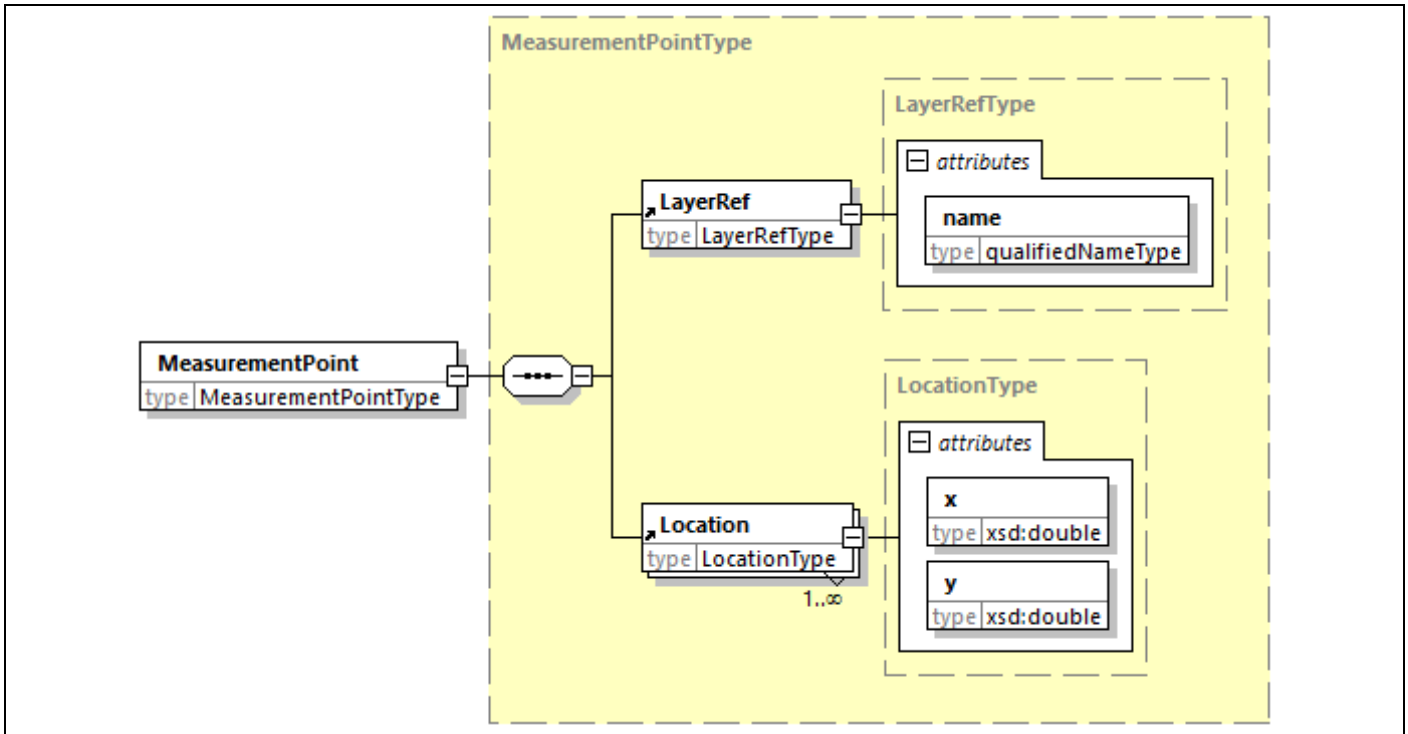
The `DfxMeasurement` element describes details of a measurement that violates its parent `Criteria`. The measurement may be made on a one layer or may span more than one layer.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
DfxMeasurement	DfxMeasurementType	An embedded element that provides the specific details of the measurements that are made on individual layers or product to determine design for manufacturing characteristics.	1-n
id	qualifiedNameType	A unique identifier that can be referenced by a DfxResponse	1-1
severity	string	A user defined indication of the seriousness of the violation.	0-1
comment	string	Any comment to assist in the definition or description of the unit details, required when OTHER is used.	0-1
Property	PropertyType	An element used in defining the actual value of the measurement to which the `Critera` property characteristics is compared to.	1-1
MeasurementPoint	MeasurementPointType	An element used in defining the specific point, or points, where the measurement was taken, with reference to a given layer in tre design.	1-n
DfxDetails	DfxDetailsType	An optional nested element containg any details pertaining to the measurement, such as a location, marker, or feature description, or reference(s) to any design object(s) or external data. See section 8.2.3.15.2.1 DfxDetails	0-1

8.2.3.15.1.1.1 Measurement Point

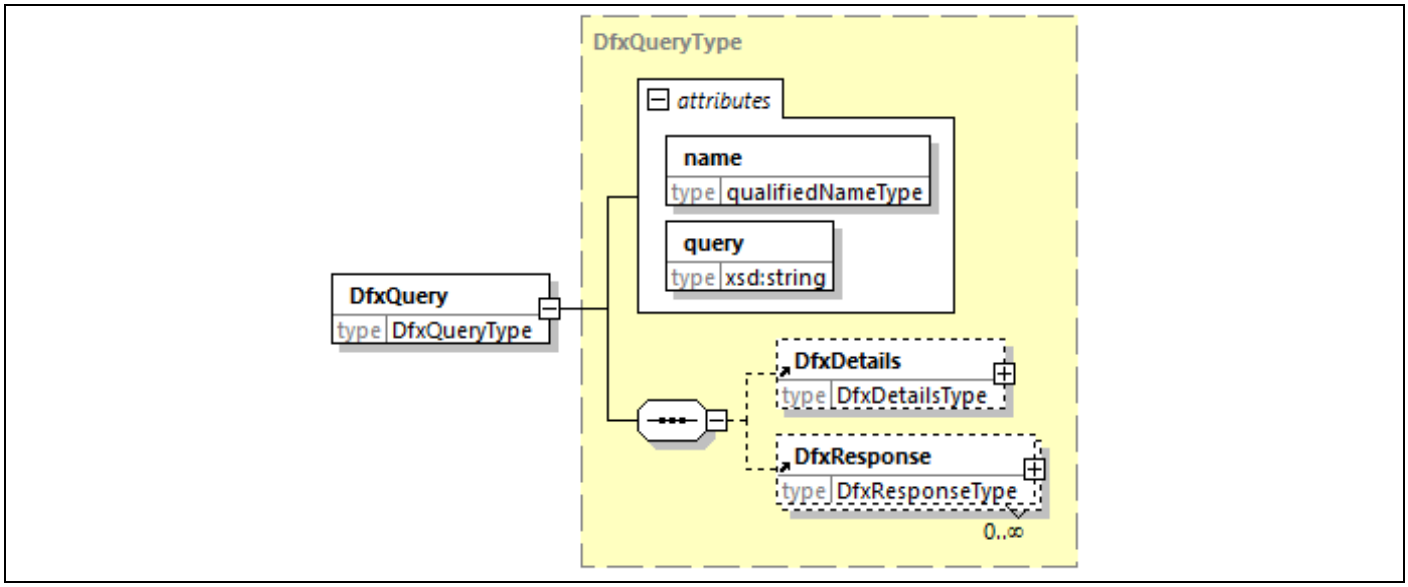
The `MeasurementPoint` element is used to define the location of a single point, multiple points or a boundary that represent the location of the concern being reported. These locations are associated to a specific layer where the measurement was taken. If there are measurement points on other layers then additional `MeasurementPoint` elements shall be added per layer.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
MeasurementPoint	MeasurementPointType	The element used to define one or more points, or a boundary from which the measurement was taken	1-n
LayerRef	LayerRefType	A reference to the layer in the Ecad layer section that pertains to the specifics of the DfxMeasurement.	1-1
Location	LocationType	The X:Y position defined in the same units as defined in the dataset	1-n

8.2.3.15.2 DfxQuery

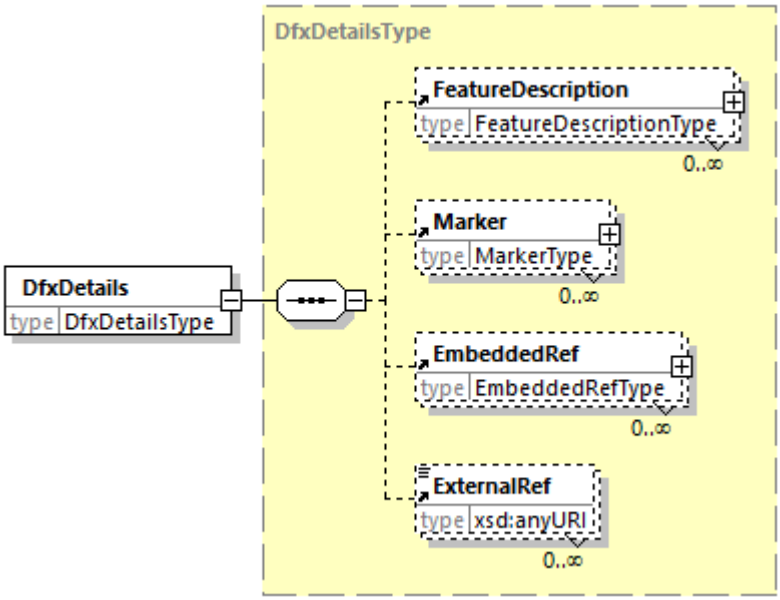
The `DfxQuery` element is used to communicate DFx issues that may not necessarily have a specific location or criteria or DFx issues that need more descriptive clarification than what can be indicated with the `Criteria` alone. This element also allows for a receiver of the DFx information to respond to the DFx items.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
name	qualifiedNameType	A descriptive name that describes the specific DfxQuery	1-1
query	string	A string describing the specific issue causing the query	1-1
DfxDetails	DfxDetailsType	An optional nested element containg any details pertaining to the measurement, such as a location, marker, or feature description, or reference(s) to any design object(s) or external data. See section 8.2.3.15.2.1 DfxDetails	0-1
DfxResponse	DfxResponseType	An optional nested element containing an enumerated response type and an optional comment about the response. Multiple responses are allowed to allow for different responses to multiple measurements per DfxQuery or Criteria DfxMeasurement elements	0-n

8.2.3.15.2.1 DfxDetails

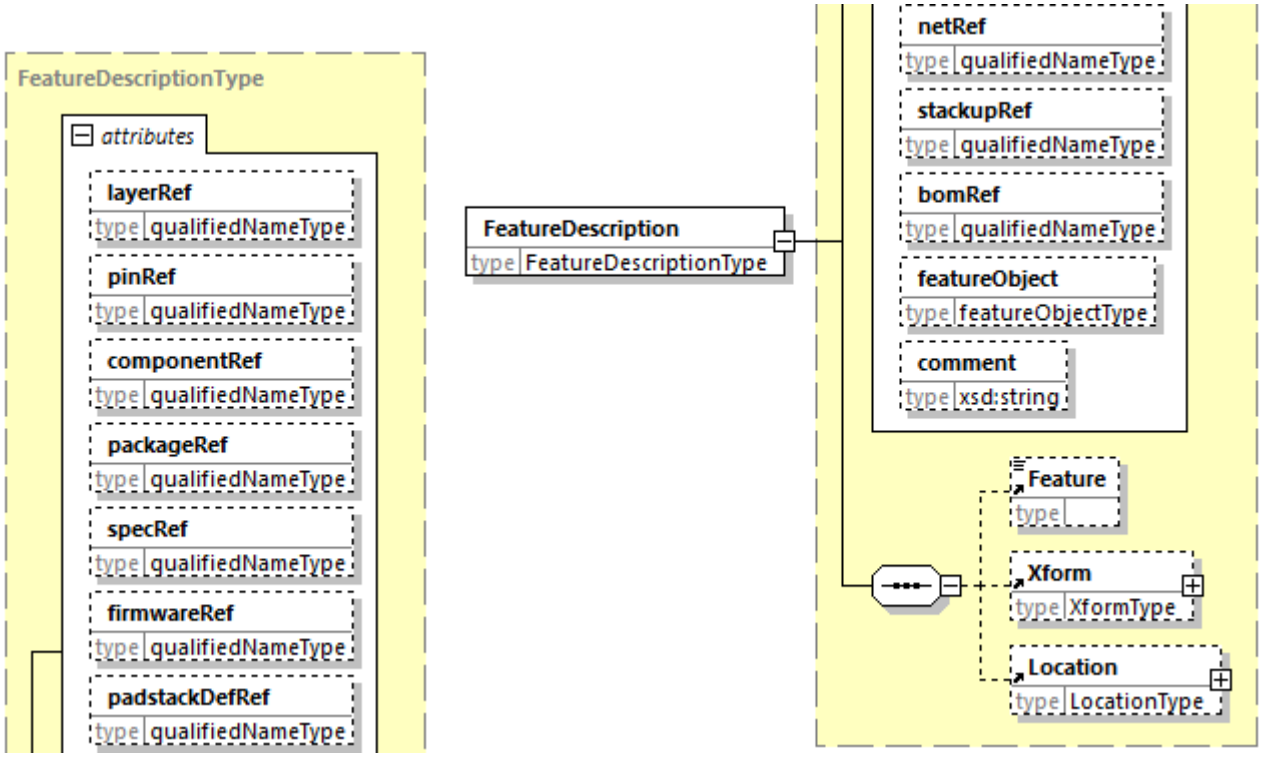
The `DfxDetails` element is used to specify any feature, layer, spec, image, or any other attribute found in `FeatureDescription` needed to identify a DFx issue.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
DfxDetails	DfxDetailsType	A nested element containg any details pertaining to a measurement, query, or response	0-1
FeatureDescription	FeatureDescriptionType	An optional element describing the feature, location, image, or reference of interest for Dfx	0-n
Marker	MarkerType	An element that contains data to define a graphical marking to identify the point(s) of interest. Uses of `Marker` include a boundary around the point(s) of interest or an identifting flag, such as a bow tie	0-n
EmbeddedRef	EmbeddedRefType	The `EmbeddedRef` element holds the file is base64 encoded. The data is encoded using base64. (see IETF RFC 1421 for the base64 algorithm and http://www.w3.org/TR/xmlschema-2/#base64Binary)	0-n
ExternalRef	anyURI	An attribute that defines a reference to an external or source of data such as a *URL*, specific file or document location. See the examples below.	0-n
<ExternalRef>file:///C:/testmethod.avi</ExternalRef> <ExternalRef>http://ipc2581.com</ExternalRef>			

8.2.3.15.2.1.1 FeatureDescription

The FeatureDescription element is used to define the geometry and location that represents the object being referenced. The location is associated to a specific layer where the object exists.

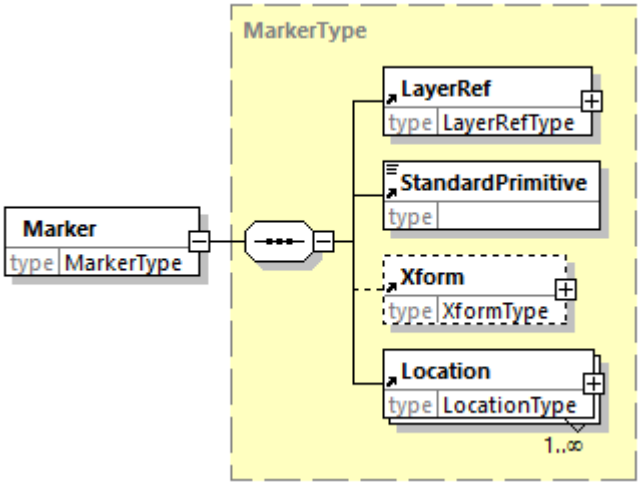


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
FeatureDescription	FeatureDescriptionType	A nested element that can be 1 or 2 feature descriptions related to the definitions used in the Simple graphic descriptions. The second FeatureDescription may be either a solution or a second feature that is in conflict with the first feature.	0-n
layerRef	qualifiedNameType	A reference to the specific layer in the Ecad layer section that pertains to the specifics of the DfxMeasurement	0-1
pinRef	qualifiedNameType	An individual `Pin` related to the place where a component attaches to the net. This description facilitates location of lands on the surface of the board or places where through-hole components are mounted.	0-1
componentRef	qualifiedNameType	Identifies the reference designator used as the attribute `refDes` of the `Component` element in `Step`. It is the reference to the component that is connected by the particular Pin and becomes a part of the electrical description of the net. The componentRef attribute is not required when used for a PinRef element under Pad elements which are under LandPattern elements. Otherwise, the componentRef attribute is required for PinRef elements under Pad, LayerPad and LogicalNet elements.	0-1
packageRef	qualifiedNameType	The name assigned to the package describing the physical outlines, documentation, and land patterns features related to package pin assignment.	0-1
specRef	qualifiedNameType	The identification of a particular specification(s) from the SpecificationType that is able to provide additional details or instructions that apply to the board or assembly descriptions of the Layer. The linkage is provided through the specific “specificationId” (spec Name)	0-1
firmwareRef	qualifiedNameType	The specific reference to firmware previously defined and contained in the DictionaryFirmware.	0-1
padstackDefRef	qualifiedNameType	A reference to the name of the padstackDef element that provides the characteristics of individual padstacks in the design, as established by the CAD system, and provides the link to the layer distribution of the padstack’s individual characteristics.	0-1
netRef	qualifiedNameType	A reference to a logical net whose reference value is a logicalNet name.	0-1
stackupRef	qualifiedNameType	A reference to the stackup where the feature resides, if there are multiple stackupds in the file.	0-1
bomRef	qualifiedNameType	A reference to a specific `Bom` element. Required if there is more than one BOM in the file	0-1

featureObject	featureObjectType	A brief description of the type of object referenced as an enumerated attribute whose values are: PAD | SHAPE | POLYGON | COMPONENT| DRILL/HOLE | CUTOUT | OUTLINE | TEXT | SLOT | CAVITY | ZONE | BEND | PORT | MILLING | OTHER (requires comment)	0-1
comment	string	To describe a `featureObject` type of OTHER, and/or any other information about the object.	0-1
Feature	ABSTRACT	An embedded element that defines a feature (`StandardShape` or `UserShape`) that may be used to describe the shape of the Dfx object.	0-1
Xform	XformType	An optional element that provides the ability to offset the origin, scale, mirror, or rotate the `Feature`.	0-1
Location	LocationType	The location of an object referenced in the design, such as pinRef or componentRef, or the location of a shape defiend by `Feature`. If `Feature` has its own built-in co-ordinates, such as `Line` or `Polyline`, then `Location` is not required.	0-1

8.2.3.15.2.1.2 Marker

The `Marker` element is used to define the location of a single point, multiple points or boundary that may be displayed graphically to represent the location of a point or area of interest. The marker is a geometry as defined in the `StandardPrimitive` substitution group. The geometry is repeated at each specified location. These locations are associated to a specific layer where the points of interest are positioned in the design data.

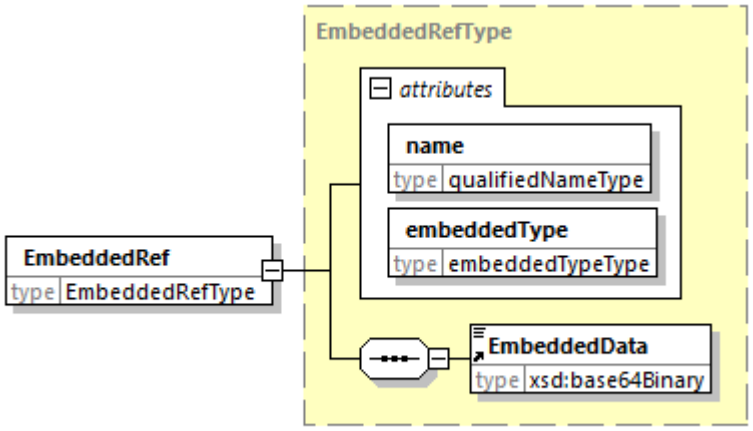


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Marker	MarkerType	A substitution group of any `StandardPrimitive` used to display a graphical shape, to flag or enclose the point(s) of interest.	0-1
LayerRef	LayerRefType	The specific layer containing the object(s) that the marker(s) refer to	1-1
StandardPrimitive	ABSTRACT	A substitution group that permits the substitution of the `StandardPrimitive` element with any of the sixteen standard primitive types.	1-1
Xform	XformType	An optional element that provides the ability to offset the origin, scale, mirror, or rotate the `StandardPrimitive`.	0-1
Location	LocationType	The X:Y position defined in the same units as defined in the dataset, which is the point of origin for each marker geometry	1-n

```
<Marker>
  <LayerRef name=”ASSEMBLY_TOP”/>
  <Diamond width=”40.00” height=”40.00”>
      <FillDescRef id=”SOLID_FILL”/>
  </Diamond>
   <Xform rotation=”90.00”/>
   <Location x=”1255.00” y=”845.00”>
</Marker>
```

8.2.3.15.2.1.3 EmbeddedRef

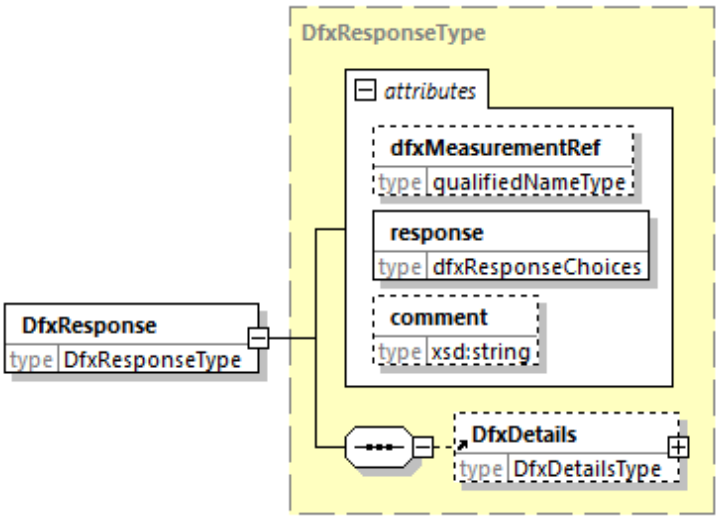
The `EmbeddedRef` element contains embedded data for an image, documents, firmware, or executable in a base64ascii format. The embedded data can be used to define, illustrate, or supplement the `DfxMeasurement`.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
EmbeddedRef	EmbeddedRefType	An Element that contains embedded data for an image, documents, firmware or executable in a base64ascii format.	0-1
name	qualifiedNameType	The qualified name type of the file name and extension.	1-1
embeddedType	embeddedTypeType	The EmbeddedType attribute value is an enumerated string that identifies the embedded object type IMAGE | FIRMWARE | EXECUTABLE | DOCUMENT	1-1
EmbeddedData	base64Binary	The EmbeddedData Attribute holds the file that is base64 encoded. The data is encoded using base64. (see IETF RFC 1421 for the base64 algorithm and http://www.w3.org/TR/xmlschema-2/#base64Binary)	1-1

8.2.3.15.2.2 DfxResponse

The DfxReponse element is used to describe a response to a parent `DfxQuery`.

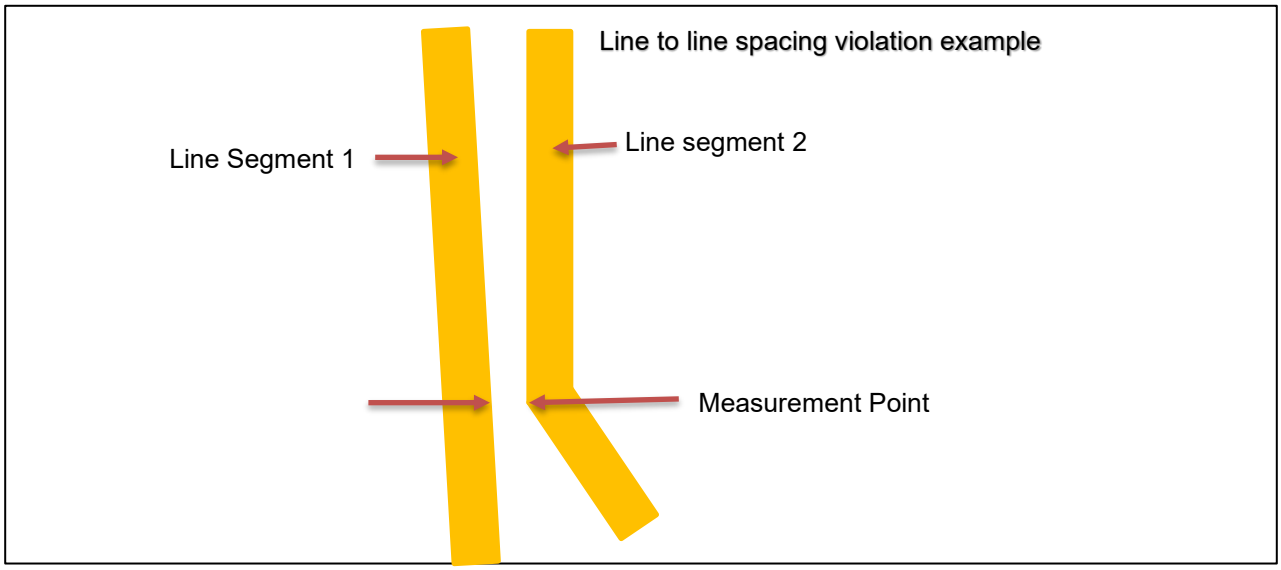


Attribute / Element Name	Attribute / Element Type	Description	Occurrence
DfxResponse	DfxResponse Type	An Element that describes a response to a parent `DfxQuery`	0-n
dfxMeasurementRef	qualifiedNameType	An optional reference to the ID of a `DfxMeasurement` element. Refer to section 8.2.3.15.1.1 DfxMeasurement. This allows each response for a parent DfxQuery to reference a specific measurement listed under Criteria.	0-1

response	dfxResponseChoices	An enumerated string containing the following response choices: YES | NO | APPROVED | CONDITIONALLY_APROVED | REJECTED | WAIVED | IGNORE | OPEN | CLOSED | OTHER	1-1
comment	string	An optional comment **shall** be used to explain a `response` choice of OTHER, and/or for any additional comments about the `response`.	0-1
DfxDetails	DfxDetailsType	An optional nested element containg any details pertaining to the response, such as a location, marker, or feature description, or reference(s) to any design object(s) or external data	0-1

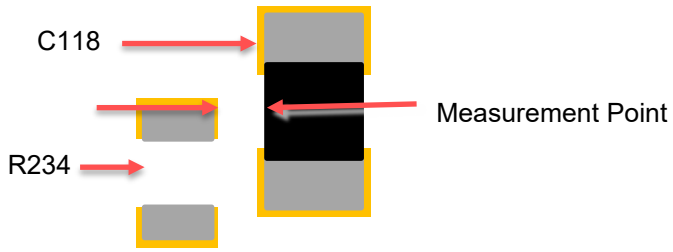
8.2.3.15.3 Dfx Examples

8.2.3.15.3.1 Line to Line Violation



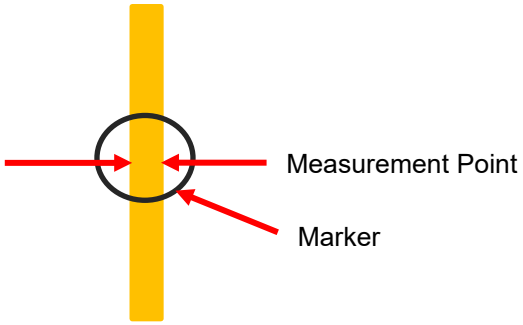

```
<Dfxt name=”FAB DRCs” category=”BOARDFAB“>
      <Criteria name=”CopperSpacing” measurementMode=”DISTANCE”>
          <Property name=”LineToLine” value=”4.0” unit=”MILS” refUnit=”MIN”/>
              <DfxMeasurement measurementMode=”DISTANCE”>
                    <Property name="Line to Line" value=“3.75" unit="MILS"/>
                    <MeasurementPoint>
                        <LayerRef  name= “Int_1”/>
                        <Location x=“300.00” y=“1200.00”/>
                        <Location x=“303.75” y=“1200.00”/>
                    </MeasurementPoint>
                    <DfxDetails>
                        <FeatureDescription layerRef=“Int_1 featureObject=”LINE”>
                              <Polyline>
                                    <PolyBegin x=“297.46” y=“1137.48”/>
                                    <PolyStepSegment x=“298.05” y=“1199.85”/>
                                    <PolyStepSegment x=“284.40” y=“1214.65”/>
                              </Polyline>
                        </FeatureDescription>
                        <FeatureDescription layerRef= “Int_1” featureObject=”LINE”>
                              <Polyline>
                                    <PolyBegint x=“305.79” y=“1137.89”/>
                                    <PolyStepSegment x=“305.79” y=“1265.68”/>
                              </Polyline>
                        </FeatureDescription>
                    </DxfDetails>
               </DfxMeasurement>
        </Criteria>
</Dfx>
```

8.2.3.15.3.2 Component Spacing Violation




```
<Dfxt name=”DFX2” category=”ASSEMBLY”>
      <Criteria name=”ComponentSpacing” measurementMode=”DISTANCE”>>
            <Property name=”packagetopackage” value=”30.00” unit=”MILS” refUnit=”MIN”>
            <DfxMeasurement measurementMode=”DISTANCE”>
                  <Property name="packagetopackage" value=“20.00" unit="MILS"/>
                  <MeasurementPoint>
                        <LayerRef  name= “Assembly_TOP”/>
                        <Location x=“1125.00” y=“1500.00”/>
                        <Location x=“1130.75” y=“1520.00”/>
                  </MeasurementPoint>
                  <DfxDetails>
                        <FeatureDescription layerRef=“Assembly_TOP” componentRef=”R234”>
                              <Location x=“1110.00” y=“1485.00”/>
                        </FeatureDescription>
                        <FeatureDescription layerRef=“Assembly_TOP” componentRef =”C118”>
                              <Location x=“1220.00” y=“1665.00”/>
                        </FeatureDescription>
             </DfxMeasurement>
      </Criteria>
</Dfx>
```

8.2.3.15.3.3 Minimum Line Width




```
<Dfx name=”comps” category=” BOARDFAB”>
      <Criteria name=”MinCopperWidth” measurementMode=”DISTANCE”>
            <Property name=”MinLineWidth” value=”4.00” unit=”MILS”>
            <DfxMeasurement measurementMode=”DISTANCE”>
                  <Property Name="MinLineWidth" value=“3.95" unit="MILS"/>
                  <MeasurementPoint>
                        <LayerRef  name= “TOP”/>
                        <Location x=“1125.00” y=“1500.00”/>
                        <Location x=“1130.75” y=“1520.00”/>
                  </MeasurementPoint>
```

```
                <DfxDetails>
                    <FeatureDescription>
                        <Line startX=”310.00” startY=”1250.00 endX=”310.00” endY=”1350”>
                            <LineDesc lineEnd=”ROUND” lineWidth=”3.95”/>
                        </Line>
                    </FeatureDescription>
                    <Marker>
                        <LayerRef  Name= “TOP”/>
                        <Circle Diameter=“15.00”>
                            <FillDesc fillProperty=”HOLLOW”/>
                            <LineDesc lineEnd=”ROUND” lineWidth=”4.00”/>
                        </Circle
                        <Location x=“1220.00” y=“1665.00”/>
                    </Marker>
                </DfxDetails>
            </DfxMeasurement>
        </Criteria>
</Dfx>
```

8.2.3.15.3.4 DfxQuery and DfxResponse with Criteria

Assembly component spacing violation example

```
  <Dfx name="Bottom_side_wave_spacing-2" category="ASSEMBLY">
    <Criteria measurementMode="DISTANCE" name="Bottom_side_wave_spacing-2">
      <Property refUnit="MILS" refValue="125" name="Component_to_wave_pin_spacing"></Property>
      <DfxMeasurement id="Bottom_side_wave_spacing-2" severity="WARM">
        <Property unit="MILS" value="98"></Property>
        <MeasurementPoint>
          <LayerRef name="BOTTOM" />
          <Location x="1.2" y="1.7" />
        </MeasurementPoint>
        <DfxDetails>
          <FeatureDescription componentRef="C138"></FeatureDescription>
          <FeatureDescription componentRef="C140"></FeatureDescription>
          <FeatureDescription componentRef="T1"></FeatureDescription>
          <FeatureDescription componentRef="L18"></FeatureDescription>
          <Marker>
            <LayerRef name="BOTTOM" />
            <RectCenter height="0.25" width="0.3"></RectCenter>
            <Location x="1.2" y="1.7"></Location>
          </Marker>
        </DfxDetails>
      </DfxMeasurement>
    </Criteria>
    <DfxQuery name="Bottom_side_wave_spacing-2" query="Current height are 228-531mil, DFM spec is 125mil min in bottom
side for wave solder fixture design. Pls check and move them to top side. C138, C140, T1, L18">
      <DfxResponse response="IGNORE" comment="This board will be processed as Paste in Hole."></DfxResponse>
    </DfxQuery>
  </Dfx>
```

8.2.3.15.3.5 DfxQuery and DfxResponse without Criteria

Component missing data example

```
  <Dfx name="Missing_datasheets" category="COMPONENT">
    <DfxQuery name="Missing_datasheets" query="No CPN or no datasheet for this location. Please provide for further DFM
analysis. T6, P10,CR58, CR57, CR56, CR55, CR39, CR38, CR37, CR36">
      <DfxDetails>
        <FeatureDescription layerRef="TOP" componentRef="T6"></FeatureDescription>
        <FeatureDescription layerRef="TOP" componentRef="P10"></FeatureDescription>
        <FeatureDescription layerRef="BOTTOM" componentRef="CR58"></FeatureDescription>
      </DfxDetails>
```

```
      <DfxResponse response="OTHER" comment="CPNs are in preliminary status and will be released to prototype as quickly as possible.  In the mean time, we will supply datasheets for the components via email."></DfxResponse>
    </DfxQuery>
  </Dfx>
```

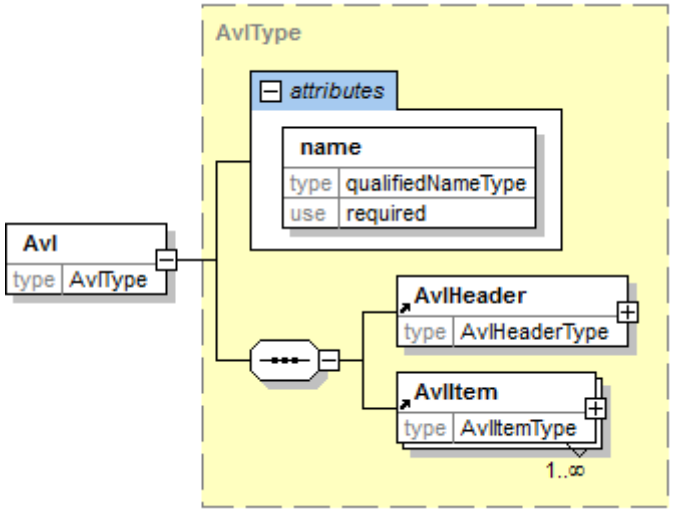
8.2.3.15.3.6 Board fab thieving required example

```
  <Dfx name="Add_dummy_pad" category="BOARDFAB">
    <Criteria measurementMode="NONE" name="Thieving-1">
      <Property name="Unbalanced copper"></Property>
      <DfxMeasurement id="RecommendedThieving-1">
        <Property name="Unbalanced copper" layerOrGroupRef="T03"/>
        <MeasurementPoint>
          <LayerRef name="T03" />
          <Location x="5.1" y="4.3" />
        </MeasurementPoint>
        <DfxDetails>
          <Marker>
            <LayerRef name="T03" />
            <Ellipse height="1.2" width="0.5"></Ellipse>
            <Location x="5.1" y="4.3"></Location>
          </Marker>
        </DfxDetails>
      </DfxMeasurement>
      <DfxMeasurement id="RecommendedThieving-2">
        <Property></Property>
        <MeasurementPoint>
          <LayerRef name="T03"></LayerRef>
          <Location x="1.1" y="1.6"></Location>
        </MeasurementPoint>
        <DfxDetails>
          <Marker>
            <LayerRef name="T03" />
            <Ellipse height="0.25" width="1.0"></Ellipse>
            <Location x="1.1" y="1.6" />
          </Marker>
        </DfxDetails>
      </DfxMeasurement>
    </Criteria>
    <DfxQuery name="Thieving" query="We can add thieving on internal layers except area of 14a. Thieving pad 40 mil squares spaced on 65 mil centers and keep 100 mil clearance from pattern on adjacent signal layer, no affect for impedance control ">
      <DfxResponse response="APPROVED"></DfxResponse>
    </DfxQuery>
  </Dfx>
```

9 APPROVED VENDOR LIST (AVL)

The AVL element defines the mapping between the OEM part or item number (OEMDesignNumber) and the vendor manufacturers part number (VMPN). When an OEM part or item number is multi-sourced this can become a one to many mapping between a single OEMDesignNumber and a list of VMPNs that the OEM has approved to be equivalent. Alternatively the mapping can be defined by an external specification, which is referenced by a SpecRef element. The AVL is created by the OEM, but can be modified, if allowed and necessary, by the board fabricator or the board assembler to reflect the materials and components used in the final electronic assembly.

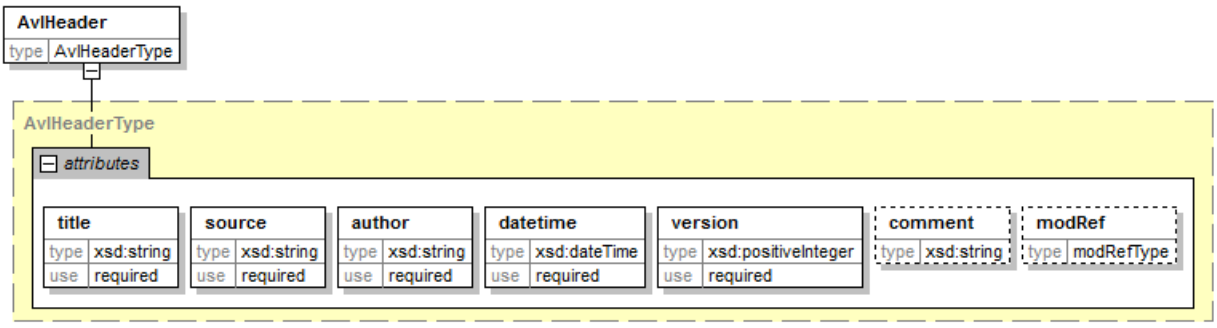
Each BomItem element in the BOM **shall** have an OEMDesignNumberRef attribute pointing to a OEMDesignNumber attribute of an AvlItem element in the AVL. Although there may be several Bill of Materials (BOM's) there is only ever one approved vendor list.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
Avl	AvlType	The element that identifies the approved suppliers for the parts listed in the `Bom` or those that have been identified from other sources such as Internal or External Vendor Libraries.	0-n
name	qualifiedNameType	A unique name assigned to a group of approved sources of supply for the materials used in building the electronic assembly.	1-1
AvlHeader	AvlHeaderType	An embedded element that defines the characteristics of the Avl file, describing the source of the information and who has the responsibility for its creation and update.	1-1
AvlItem	AvlItemType	An embedded element that indicates the details of the approved supplier information and specifically indicates the relationship to all items in the file contained within every qualified named `Bom` element.	1-n

9.1 AvlHeader

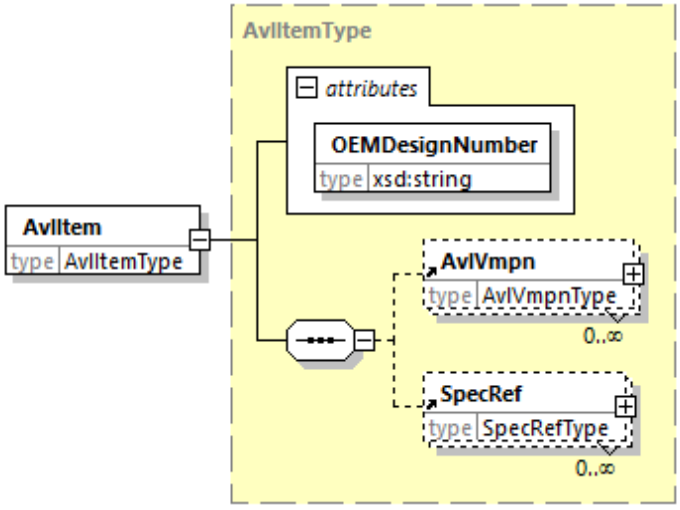
The `AvlHeader` element defines the characteristics of the Avl information contained in the specific Avl file. Its occurrence is related to the name associated with the Avl file and may have different source information based on the purpose of the specific Avl. The `dateTime` attribute is used to keep account of changes that may take place in updating the information in the Avl file.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
AvlHeader	AvlHeaderType	An embedded element that defines the characteristics of the Avl file, describing the source of the information and who has the responsibility for its creation and update.	1-1
title	string	A unique title provided to the AvlHeader that distinguishes the schema instance from the `AvlDataHeader`.	1-1
source	string	The source of the information contained in the Avl file. If multiple sources exist they **shall** be so indicated with the relationship to the Avl data identified.	1-1
author	string	The individual responsible for the header creation and update. If the responsibility changes the information for this attribute must also change. It is required the person and their enterprise be contained in the `LogisticHeader` instances and the string name be identical to the id of the `Person` instance.	1-1
datetime	dateTime	The date and time that the header was created or modified.	1-1
version	positiveInteger	A positive number that defines the revision level of the AvlHeader.	1-1
comment	string	Any comment to assist in the interpretation of the Avl data.	0-1
modRef	modeType	A reference to the mode intended for the file transfer. An enumerated string consisting of: USERDEF | BOM | STACKUP | ASSEMBLY | FABRICATION | TEST | STENCIL	0-1

9.2 AvlItem

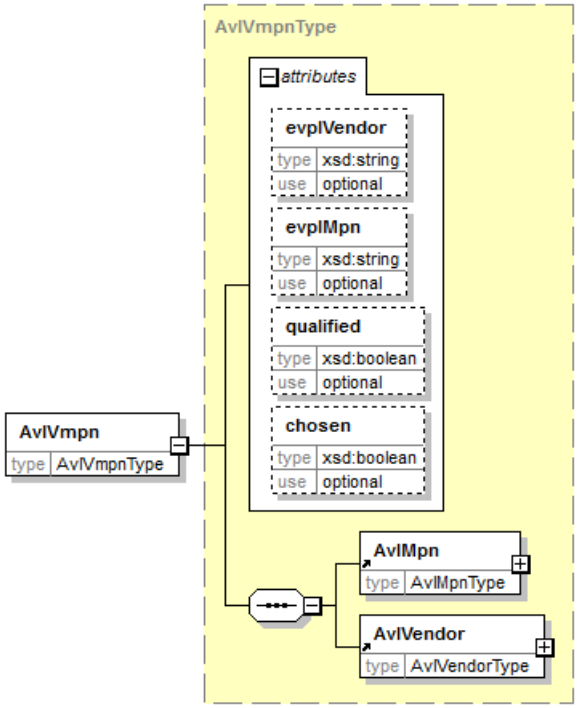
The `AvlItem` element consists of specific approved vendor information related the Bom data items and the part numbers (`OEMDesignNumber`) specified by the originator of the IPC-2581 file. Each `AvlItem` instance starts with its own `AvlDataHeader` in order to establish the relationship with the appropriate Bom. The grouping of AvlItem's provides the information on the individual relationship to vendor manufacturing part numbers (`AvlVmpnList`).



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
AvlItem	AvlItemType	An embedded element that indicates the details of the approved supplier information and specifically indicates the relationship to all items in the file contained within every qualified named Bom element.	1-n
OEMDesignNumber	string	The identification of the unique part number used by the OEM customer in the Component XML instance.	1-1
AvlVmpn	AvlVmpnType	An embedded element that describes an individual approved vendor part number which corresponds to the part selected by the customer and contained in the group of Component's or, if approved, the `VplComponent` grouping.	0-n
SpecRef	SpecRefType	The identification of a particular specification(s) from the SpecificationType that is able to provide additional details or instructions that apply to the AvlItem. The linkage is provided through the specific "specificationId" (spec Name)	0-n

9.2.1 AvlVmpn

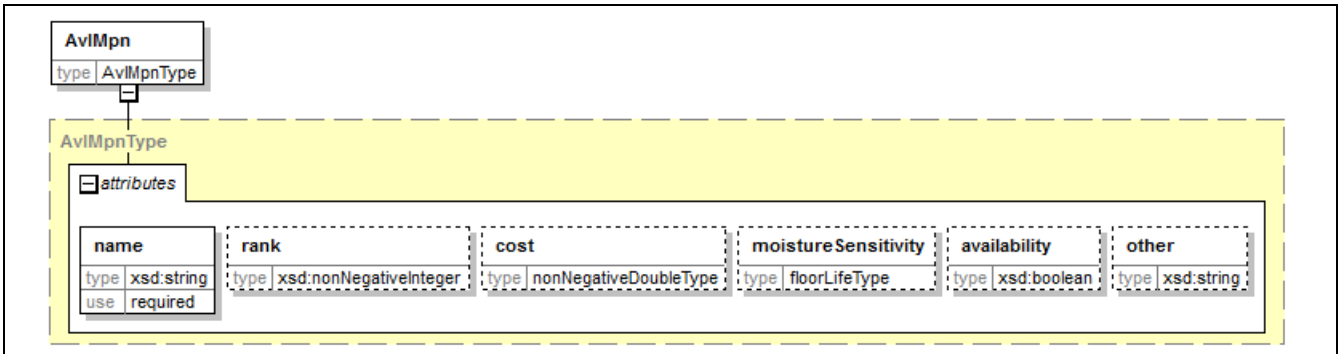
The `AvlVmpn` element represents the approved vendor part number substitution that might be implemented as an alternate to the `OEMDesignNumber` supplied by the user. The details of the information may be extracted from an external vendor part library, and if this technique is used the attributes of “qualified” and “chosen” must be included in the file.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
AvlVmpn	AvlVmpnType	An embedded element that describes an individual approved vendor part number which corresponds to the part selected by the customer and contained in the group of Component’s or, if approved, the VplComponent grouping.	1-n
evplVendor	string	The identification of an external vender part library (evpl) which may include a URL to the total library domain.	0-1
evplMpn	string	The manufactures part number in the evpl.	0-1
qualified	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lower case). True equals that the external vendor part library has been qualified; False indicates that it has not been qualified. If the attribute is not present the qualification is unknown.	0-1
chosen	boolean	An enumerated string is either “true” or “false” (part of the 3WC standard – must be lower case). True equals that the external vendor part library has been chosen; False indicates that it has not been chosen. If the attribute is not present the fact as to whether the vendor part library has been chosen or not is unknown.	0-1
AvlMpn	AvlMpnType	An embedded element that defines the name and rank of the particular part identified for possible selection.	1-n
AvlVendor	AvlVendorType	An embedded element that defines the vendor(s) who produce the part in question as extracted from the external vendor part library and added to the enterprise information in the `LogisticHeader`.	1-n

9.2.1.1 AvlMpn

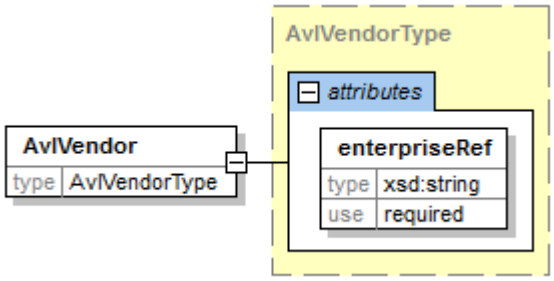
The `AvlMpn` element defines the name and ranking of the particular approved vendor part. Information is provided that identifies the characteristics of the substitution part in order to help the selection process.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
AvlMpn	AvlMpnType	An embedded element that defines the name and rank of the particular part identified for possible selection.	1-n
name	string	The name of the part being suggested as an appropriate substitution	1-1
rank	nonNegativeInteger	The appropriateness of the part for its ability to serve as the same form fit and function of the original part identified as the OEMDesignNumber. The number 1 is the best ranking.	0-1
cost	nonNegativeDouble Type	The cost of the part when purchased in a reasonable quantity	0-1
moistureSensitivity	floorLifeType	An identification of the parts' ability to resist moisture penetration. It is an enumerated string that matches the requirements of J-STD-020 and is one of the following: UNLIMITED | 1_YEAR | 4_WEEKS | 168_HOURS | 72_HOURS | 48_HOURS | 24_HOURS | BAKE	0-1
availability	boolean	An enumerated string is either "true" or "false" (part of the 3WC standard – must be lower case). True equals that the part is readily available; False indicates that it is not. If the attribute is not present the fact as to the parts availability is unknown.	0-1
other	string	Any other information pertinent to the information about the manufacturers part number	0-1

9.2.1.2 AvlVendor

The AvlVendor element is the linkage back to the Enterprise information defining the location of the part manufacturer, distributor or other source.



Attribute / Element Name	Attribute / Element Type	Description	Occurrence
AvlVendor	AvlVendorType	An embedded element that defines the vendor(s) who produce the part in question as extracted from the external vendor part library and added to the enterprise information in the `LogisticHeader`.	1-n
enterpriseRef	string	A reference to the `Enterprise id` attribute identifying the company that is able to deliver the required part. The information must be available in the `LogisticHeader`	1-1

10 GLOSSARY

Name or Acronym	Description
IPC-2581	Top level data structure
Avl	Approved Vendor List
Bom	Bill of Material
Ecad	Computer-Aided design information
Contents	Information about contents of the file
HistoryRef	Information about order and supply data
LogisticHeader	File change information
Header	Header
AvlVmpnList	Manufacturer part number list
AvlVmpn	Manufacturer Part Number
AvlVendor	Vendor

10.1 Process Flow Descriptions

The detail shown in Figure 17 indicates the flow of data between design and manufacturing. Terminology may change as each domain performs their particular function, so the flow highlights the naming convention as the physical item moves through various steps in the process.

Usage of IPC-T-50 Terminology During the PCB Fabrication adn Component Population at an Assembly House
Flow Diagram

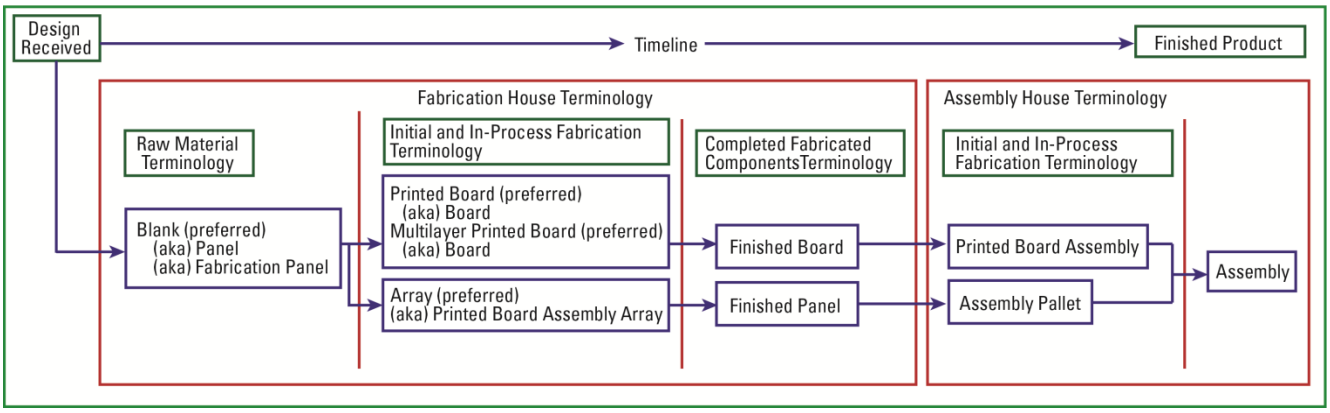


Figure 17 - Electronic printed board design, through fabrication, assembly and test process data flow

10.2 Terms and Definitions

The definition of all terms **shall** be in accordance with IPC-T-50 and the following. An asterisk (*) by the term indicates that it is a reproduction from IPC-T-50 and is provided to assist the reader in interpretation of this standard. The order of the terms is related to the IPC-2581 process flow shown in 10.1

Array* **22.0049**
A group of elements or circuits arranged in rows and columns on a base material.

Assembly* **80.1327**
A number of parts, subassemblies or combinations thereof joined together. (Note: This term can be used in conjunction with other terms listed herein, e.g., ‘‘Printed Board Assembly’’)

Assembly Pallet
The generic term for the assembly that uses a finished panel, as delivered from the board fabricator, of the same or different designs, for element and circuit component mounting and attachment to the board interconnections layers. The board arrangement on the pallet may be random or in the form of an array; the pallet may also include coupons for testing.

Blank * **41.1339**
An unprocessed or partially processed piece of base material or metal clad base material, that has been cut from a sheet or panel, that has the rough dimensions of a printed board. (See also “Panel.”)

Board * **60.0118**
see “Printed Board,” and “Multilayer Printed Board.”

Fabrication Panel
(See “Panel”)

Finished Board
see “Printed Board”

Finished Panel
A rectangular sheet of base material or metal-clad material of predetermined size that is used for the processing of one or more printed board designs and, when required, one or more test coupons which is extracted from the fabrication panel to deliver to the customer or to the next level of fabrication. (see Assembly Pallet)

Multilayer Printed Board * 60.1227
The general term for a printed board that consists of rigid or flexible insulation materials and three or more alternate printed wiring and/or printed circuit layers that have been bonded together and electrically interconnected.

Panel 41.1463
A rectangular sheet of base material or metal-clad material of predetermined size that is used for the processing of one or more printed boards and, when required, one or more test coupons. (See also ‘‘Blank.’’)

Printed Board (PB) * 60.1485
The general term for completely processed printed circuit and printed wiring configurations. (This includes single-sided, double-sided and multilayer boards with rigid, flexible, and rigid-flex base materials.)

Printed Board Assembly* 80.0911
The generic term for an assembly that uses a printed board for component mounting and interconnecting purposes.

Printed Board Assembly Array
A group of assemblies, all of the same design, arranged in rows and columns on a panel.

10.3 Enumerated strings of IPC-2581

The following enumerations exist for IPC-2581 data restrictions. The terms used in the started are intended to relate to the definitions shown in 10.2.

BOARD | BOARDPANEL | ASSEMBLY | ASSEMBLYPALLET | COUPON

11 REFERENCE INFORMATION

The following sections define reference documents that are useful in clarifying the products or process of the industry or provide additional insight into the subject of data modeling or released information models.

11.1 IPC

IPC-T-50 Terms and Definitions for Interconnecting and Packaging Electronic Circuits

IPC-D-310 Guidelines for Phototool Generation and Measurement Techniques

IPC-D-325 Documentation Requirements for Printed Boards

IPC-2220 series, Design Standards for Printed Boards and Printed Board Assemblies

IPC-2501 Definition for Web-Based Exchange of XML Data

IPC-2511 Generic Requirements for Implementation of Product Manufacturing Description Data and Transfer Methodology

IPC-2571 Generic Requirements for Electronics Manufacturing Supply Chain Communication – Product Data eXchange (PDX)

IPC-2576 Sectional Requirements for Electronics Manufacturing Supply Chain Communication of As-Built Product Data – Product Exchange (PDX)

IPC-2578 Sectional Requirements for Supply Chain Communication of Bill of Material and Product Design Configuration Data - Product Data eXchange (PDX)

IPC-2611 Generic Requirements for Electronic Product Documentation

IPC-2614 Sectional Requirements for Board Fabrication Documentation

IPC-2615 Printed Board Dimensions and Tolerances

IPC-4101 Specification for Base Materials for Rigid Board and Multilayer Printed Boards

IPC-4103 Specification for Base Materials for High Speed/ High Frequency Applications

IPC-4104 Specification for High Density Interconnect (HDI) and Microvia Materials

IPC-7351 Generic Requirements for Surface Mount Design of Land Pattern Standard

11.2 American National Standards Institute

ANSI X3/TR-1-77 *American National Dictionary for Information Processing*

ANSI X3.12 *Subroutine Record Format Standardization*

ANSI Y14.5 *Dimensioning and Tolerancing for Engineering Drawing*

ANSI Y32.1 *Logic Diagram Standards*

ANSI Y32.16 *Electrical and Electrical Reference Designators*

ANSI Z210.1 *Metric Practice Guide (ASTM 380-72)*

11.3 Department of Defense

DoD-STD-100 *Engineering Drawings*

11.4 Electronic Industries Association

EDIF 4 0 0 *Electronic Data Interchange Format*

11.5 International Organization for Standards (ISO)

ISO STEP Documentation:

ISO 10303-AP210 *Electronic Assembly, Interconnect, and Packaging Design*

ISO 10303-AP212 *Electrotechnical Design & Installation*

AP220 *Process Planning, Manufacturing, and Assembly of Layered Electronic Products*

AP221 *Process Plant Functional Data & Schematic Representation*

Appendix A Panel Instance File

The following is the full XML instance file for the panel shown in the illustration below. It has passed schema validation, so should load into any tool/viewer that supports IPC-2581 revision B1.

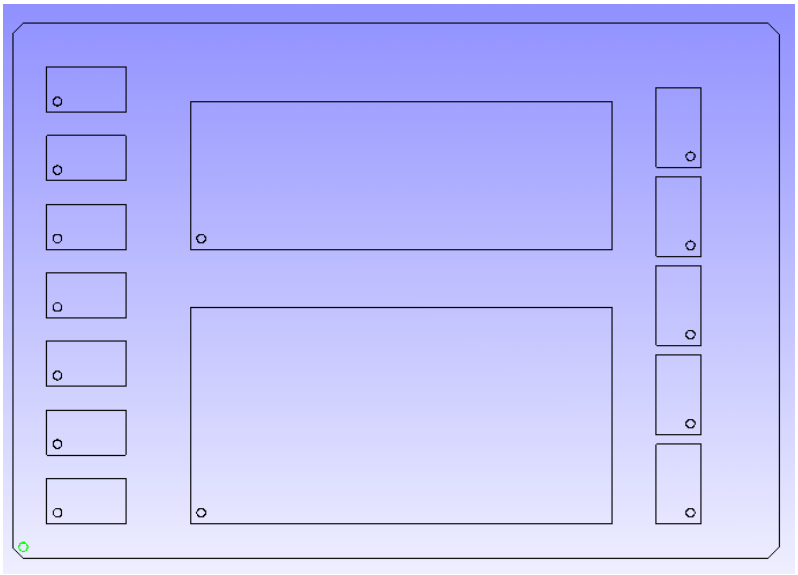


Figure 18 - Multiple board designs and coupon panel

```
<?xml version="1.0" encoding="UTF-8"?>
<IPC-2581 revision="B1" xmlns="http://webstds.ipc.org/2581" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://webstds.ipc.org/2581 http://www.ipc.org/2581/IPC-2581B1.xsd">
   <Content roleRef="DesignOwner">
      <FunctionMode mode="USERDEF" level="1" comment="Panel Test"/>
      <StepRef name="panel_1" />
      <StepRef name="basicboard_1" />
      <StepRef name="basicboard_2" />
      <StepRef name="basicboard_3" />
      <LayerRef name="OnlyLayer" />
   </Content>
   <LogisticHeader>
      <Role id="DesignOwner" roleFunction="OWNER" description="Design Owner" publicKey="" authority="" />
      <Enterprise id="A Design Company" code="NONE" />
      <Person name="A User" enterpriseRef="A Design Company" roleRef="DesignOwner" />
   </LogisticHeader>
   <HistoryRecord number="1" origination="2015-11-13T10:20:02" software="A Tool" lastChange="2015-11-13T10:20:02">
      <FileRevision fileRevisionId="1.0" comment="Initial Version">
         <SoftwarePackage name="A Tool" vendor="A Tool Company" revision="1.0">
            <Certification certificationStatus="BETA" />
         </SoftwarePackage>
      </FileRevision>
   </HistoryRecord>
   <Bom name="BOM_panel_1">
      <BomHeader assembly="Assembly1" revision="1.0">
         <StepRef name="panel_1" />
      </BomHeader>
      <BomItem OEMDesignNumberRef="orderNumber_brd1" quantity="12" category="MATERIAL">
         <Characteristics category="MATERIAL"/>
      </BomItem>
      <BomItem OEMDesignNumberRef=" orderNumber_brd2" quantity="1" category="MATERIAL">
        <Characteristics category="MATERIAL"/>
      </BomItem>
      <BomItem OEMDesignNumberRef=" orderNumber_brd3" quantity="1" category="MATERIAL">
        <Characteristics category="MATERIAL"/>
      </BomItem>
   </Bom>
   <Ecad name="Design">
```

```
<CadHeader units="MILLIMETER">
</CadHeader>
<CadData>
  <Layer name="OnlyLayer" layerFunction="DOCUMENT" side="NONE" polarity="POSITIVE">
  </Layer>
  <Stackup name="DummyStackup" overallThickness="0" tolPlus="0" tolMinus="0" whereMeasured="OTHER">
  </Stackup>
  <Step name="panel_1">
    <Datum x="0.0" y="0.0" />
    <Profile>
      <Polygon>
        <PolyBegin x="-10.0" y="680.0" />
        <PolyStepSegment x="-10.0" y="0.0" />
        <PolyStepSegment x="0.0" y="-10.0" />
        <PolyStepSegment x="980.0" y="-10.0" />
        <PolyStepSegment x="990.0" y="0.0" />
        <PolyStepSegment x="990.0" y="680.0" />
        <PolyStepSegment x="980.0" y="690.0" />
        <PolyStepSegment x="0.0" y="690.0" />
        <PolyStepSegment x="-10.0" y="680.0" />
      </Polygon>
    </Profile>
    <StepRepeat stepRef="basicboard_1" x="57.0" y="12.0" nx="1" ny="7" dx="0" dy="102.0" angle="0.00" mirror="false" />
    <StepRepeat stepRef="basicboard_2" x="260.0" y="467.0" nx="1" ny="1" dx="0" dy="0" angle="0.00" mirror="false" />
    <StepRepeat stepRef="basicboard_1" x="928.0" y="18.0" nx="1" ny="5" dx="0" dy="138.0" angle="90.00" mirror="false" />
    <StepRepeat stepRef="basicboard_3" x="260.0" y="110.0" nx="1" ny="1" dx="0" dy="0" angle="0.00" mirror="false" />
    <LayerFeature layerRef="OnlyLayer">
      <Set polarity="POSITIVE">
        <Color r="153" g="255" b="255" />
        <Features>
          <Arc startX="3" startY="3" endX="3" endY="3" centerX="0" centerY="0" clockwise="false">
            <LineDesc lineWidth="1.4" lineEnd="ROUND" />
          </Arc>
        </Features>
      </Set>
    </LayerFeature>
  </Step>
  <Step name="basicboard_1">
    <Datum x="0.0" y="0.0" />
    <Profile>
      <Polygon>
        <PolyBegin x="-17.0" y="-12.0" />
        <PolyStepSegment x="110.0" y="-12.0" />
        <PolyStepSegment x="110.0" y="54.0" />
        <PolyStepSegment x="-17.0" y="54.0" />
        <PolyStepSegment x="-17.0" y="-12.0" />
      </Polygon>
    </Profile>
    <LayerFeature layerRef="OnlyLayer">
      <Set polarity="POSITIVE">
        <Color r="153" g="255" b="255" />
        <Features>
          <Arc startX="3" startY="3" endX="3" endY="3" centerX="0" centerY="0" clockwise="false">
            <LineDesc lineWidth="0.025" lineEnd="ROUND" />
          </Arc>
        </Features>
      </Set>
    </LayerFeature>
  </Step>
  <Step name="basicboard_2">
    <Datum x="0.0" y="0.0" />
    <Profile>
      <Polygon>
        <PolyBegin x="-25.0" y="-10.0" />
        <PolyStepSegment x="484.0" y="-10.0" />
        <PolyStepSegment x="484.0" y="122.0" />
        <PolyStepSegment x="-25.0" y="122.0" />
        <PolyStepSegment x="-25.0" y="-10.0" />
      </Polygon>
    </Profile>
    <LayerFeature layerRef="OnlyLayer">
      <Set polarity="POSITIVE">
```

```
            <Color r="153" g="255" b="255" />
            <Features>
              <Arc startX="3" startY="3" endX="3" endY="3" centerX="0" centerY="0" clockwise="false">
                <LineDesc lineWidth="0.025" lineEnd="ROUND" />
              </Arc>
            </Features>
          </Set>
        </LayerFeature>
      </Step>
      <Step name="basicboard_3">
        <Datum x="0.0" y="0.0" />
        <Profile>
          <Polygon>
            <PolyBegin x="-25.0" y="-20.0" />
            <PolyStepSegment x="484.0" y="-20.0" />
            <PolyStepSegment x="484.0" y="245.0" />
            <PolyStepSegment x="-25.0" y="245.0" />
            <PolyStepSegment x="-25.0" y="-20.0" />
          </Polygon>
        </Profile>
        <LayerFeature layerRef="OnlyLayer">
          <Set polarity="POSITIVE">
            <Color r="153" g="255" b="255" />
            <Features>
              <Arc startX="3" startY="3" endX="3" endY="3" centerX="0" centerY="0" clockwise="false">
                <LineDesc lineWidth="0.025" lineEnd="ROUND" />
              </Arc>
            </Features>
          </Set>
        </LayerFeature>
      </Step>
    </CadData>
  </Ecad>
  <Avl name="AVL_Panel_1">
    <AvlHeader title="latest AVL" source="A PLM Tool" author="A User" datetime="2015-11-13T10:20:02" version="1"/>
    <AvlItem OEMDesignNumber=" orderNumber_brd1"/>
    <AvlItem OEMDesignNumber=" orderNumber_brd2"/>
    <AvlItem OEMDesignNumber=" orderNumber_brd3"/>
  </Avl>
</IPC-2581>
```

Appendix B

Potential Reference Designator Assignment for Non Electrical Items

Characteristic	Reference Designator	BomDes Types & Comments
CAPACITIVE	CA	MatDes
COATINGCOND	CC	MatDes
COATINGNONCOND	CN	MatDes
CONDFILM	CM	MatDes
CONDFOIL	CF	MatDes
CONDUCTOR	CD	MatDes
CONDUCTIVE_ADHESIVE.	CA	MatDes
DIELBASE	DB	MatDes
DIELCORE	DC	MatDes
DIELPREG	DP	MatDes
DIELADHV	DA	MatDes
DIELBONDPLY	DY	MatDes
DIELCOVERLAY	DV	MatDes
EDGE_PLATING	EP	MatDes
GLUE	GL	MatDes
HOLEFILL	HF	MatDes
LEGEND	LG	MatDes
MIXED	MX	MatDes
PASTEMASK	PM	MatDes
PLANE	PL	MatDes
RESISTIVE	RS	MatDes
SIGNAL	SG	MatDes
SILKSCREEN	SS	MatDes
STACKUP_COMPOSITE	SC	MatDes
SOLDERBUMP	SB	MatDes
SOLDERMASK	SM	MatDes
SOLDERPASTE	SP	MatDes
STIFFENER	ST	MatDes

Characteristic	Reference Designator	BomDes Types & Comments
DRILL	DT	ToolDes
EDGE_CHAMFER	EC	ToolDes
FIXTURE	FX	ToolDes
PROBE	PT	ToolDes
ROUTE	RT	ToolDes
SCORE	SC	ToolDes
V_CUT	VC	ToolDes
ASSEMBLY	AS	DocDes
BOARDFAB	PB	DocDes
BOARD_OUTLINE	BO	DocDes
COMPONENT	CO	DocDes
COMPONENT_TOP	CT	DocDes
COMPONENT_BOTTOM	CB	DocDes
COMPONENT_EMBEDDED	CE	DocDes
COMPONENT_FORMED	CF	MatDes
COURTYARD	CY	DocDes
DOCUMENT	DO	DocDes
GRAPHIC	GR	DocDes
LANDPATTERN	LP	DocDes
PIN	PN	DocDes
REWORK	RW	DocDes
THIEVING_KEEP_INOUT	TK	DocDes
OTHER	OR	(TBD)

Appendix C

Summary of Changes from Revision B Am1

Changes Per Section

Section 1 Scope

- Removed any company names and specific tool names, per IPC protocol

Section 3.1 Naming attributes within a IPC-2581 file

- Changed definition of of type `qualifiedNameType` to pattern ([^:]+)(:[^:]+)?"/>;]*)*
- Removed all references to type `shortName` in the this document, and replaced all instances of `shortName` with `xsd:string` in the schema file

Section 3.3 Transform Characteristics (Xform)

- Added optional attribute `faceUp` to element `Xform`, to be able to describe a component placement such that the component pins are the opposite side of the component to the mounting layer
- Added additional wording to the `mirror` attribute to describe how it can be used to describe a component
- placement on the underside of any layer, implying the component is embedded when not on the bottom layer.

Section 3.3.6 The faceUp Attribute

- Added details on how the `faceUp` attribute works in combination with the `mirror` attribute.

Section 3.4 Base Elements

- Added sub sections for `PinRef` & `Location`, since they are used by other parent elements.

Section 3.5.9 StandardPrimitive

- Removed the `Xform` child element from all `StandardPrimitive` types.
- Ensured all `LineDesc` and `FillDesc` child elements wer updated to `LineDescGroup` and `FillDescGroup` respectively, to match the schema change that occured in revision B1.

Section 4 CONTENT

- Order of the `Dictionary*` elements changed within the child element sequence of parent element `Content`, so that a dictionary item definition is created before a reference to that dictionary item is defined
- Reflected this order in the order which the `Dictionary*` elements are defined – sections 4.6 thru 4.12

Section 4.1 FunctionMode

- Added new `FunctionMode` type DFX to support only `Dfx` element data in the output file.
- Made the LogicalNet section optional for Fabrication & Assembly modes.

Section 6.1 HistoryRecord

- Added optional attribute `lifecyclePhase`

Section 7.2 BomItem

- Added child element `SpecRef` to reference any specification to the BomItem. Examples are a purchase specification or assembly instructions for `RefDes` or `FindDes` children, a material definition for `MatDes` children, or document details for `DocDes` children.

Section 8.1.1 Spec

- Added EdegPlating, Flex, Loss, SecondaryDrill, and SurfaceFinish to SpecificationType and updated others. See below for details.

Section 8.1.1.4 Conductor

- Updated the Conductor specification, adding attribute material with enumerated options COPPER | SILVER_INK | CARBON_INK | CONDUCTIVE_INK, and attribute foilType with enumerations per IPC-4652.
- Added enumeration values WEIGHT and PRODUCT_NAME to attribute type, and made type optional.

Section 8.1.1.5 Dielectric

- Added enumeration values Tg_DSC | Tg_DMA | Tg_TMA | Td | SLASH_ IPC4101 | SLASH _IPC4103 | PRODUCT_NAME to attribute type.
- Removed enumeration values PROCESSABLITY_TEMPERATURE and GLASS_TYPE from attribute type.

Section 8.1.1.7 Impedance

- Updated the Impedance specification so that all requirements a given impedance can be described within a single Impedance element. This includes adding new substitution child elements SingleEnded, EdgeCoupled, BroadsideCoupled, CoplanarWaveguide. The EdgeCoupled type includes a substitution group LineGap which can be either Spacing (edge to edge) or Pitch (center to center).

Section 8.1.1.7.6 Impedance SpecRef Instantiation

- Added an explanantion of the possible parent elements of an Impedance SpecRef, and their order of precedence when there is a conflict.

Section 8.1.1.11 Tool

- Amended the Tool Specifiaction Type to be able to described Torque requirements – added WRENCH | SCREWDRIVER | PLASMA | PUNCH to tooListType & TORQUE | HEX_NUT_SIZE | PHILLIPS_HEAD | FLAT_HEAD | TORX_HEAD | ALLEN_HEAD to toolPropertyListType

Section 8.1.1.14 SecondaryDrill

- Added new specification type SecondaryDrill, which can be refereneced by a SpecRef child element of Hole to define a countersink or counterbore feature

Section 8.1.1.15 EdgePlating

- Added new specification element EdgePlating, to define areas of the board edge that require copper plating

Section 8.1.1.16 SurfaceFinish

- Added new specification element SurfaceFinish, to define enumerated values of surface finish in accordance with IPC-6012

Section 8.1.1.17 Flex

- Added new specification element Flex, to define enumerated properties ADHESIVE_SQUEEZE_OUT, DIELECTRIC_SQUEEZE_OUT & STRESS_RELIEF_FILLET_WIDTH for a flex zone.

Section 8.1.1.18 Loss

- Added new specification element Loss, with a loss type of one of the following enumerated values ATTENUATION | IMPEDANCE | INSERTION | POWER | SIGNAL | VOLTAGE

Section 8.1.2 Property

- Added values IN-LB | IN-OZ | FT-LB | N-m | N-cm | MIN | MAX | OZ | OZ/SQ-FT | GRAMS | HENRYS | AMPS | WATTS | VOLTS | FARAD | dB | dB/INCH | dB/MM to propertyUnitType, which is used by attributes unit and refUnit of the Property element

Section 8.1.3 ChangeRec

- Added new attribute lifecyclePhase to ChangeRec, to indicate a lifecycle phase to the data

Section 8.2.1 Layer

- Added new child element `Profile` to `Layer`, to define board outlines as seen at the Layer, caused by variations in the stackup cross section across the stackup zones in a rigid-flex design.
- Changed EMBEDDED_COMPONENT to COMPONENT_EMBEDDED in `layerFunctionType`, to be consistent with COMPONENT_TOP and COMPONENT_BOTTOM
- Added new `layerFunctionType` COMPONENT_FORMED to define components created by a printing or etching process
- Added new `layerFunction` types to Table 5

Section 8.2.2.1 Stackup

- Changed child element `MatDes` to attribute `matDes`
- Added optional attribute `tolPercent`, to be able to specify thickness tolerance as a percentage
- Added required attribute `stackupStatus` with possible enumerations SPECIFIED | PROPOSED | APPROVED

Section 8.2.2.1 StackupGroup

- Changed child element `MatDes` to attribute `matDes`
- Added optional attribute `tolPercent`, to be able to specify thickness tolerance as a percentage

Section 8.2.2.1.1 StackupLayer

- Changed child element `MatDes` to attribute `matDes`
- Added optional attribute `tolPercent`, to be able to specify thickness tolerance as a percentage

(Old) Section 8.2.3.2 PadStack

- Completely removed the definition of PadStack, and its children LayerHole and LayerPad, and all references to them. These elements are no longer supported.
- Subseqent section numbers then moved up, e.g. PadStackDef changed form section 8.2.3.3 to 8.2.3.2, Datum changed from section 8.3.3.4 to 8.2.3.3, etc.

Section 8.2.3.2.1 PadStackHoleDef

- Added type VIA_CAPPED to the enumerated list of `platingStatusType`, for attribute `platingStatus`, to identify plated over vias, also known as VIPPO, CAPPED, POFV

(Old) Section 8.2.3.3 Route

- Removed the `Route` section, as it was unused and unnecessary. So the `Datum` section is now 8.2.3.3, and all other sub sections of 8.2.3 after that have moved up.

Section 8.2.3.6 Package

- Added child element `Topside` (described in section 8.2.3.6.5) to element `Package`, to define pins and other features on the top of the package.
- Added child element `OtherSideView` (described in section 8.2.3.6.6) to element `Package`, to define any outline, silkscreen, or assembly drawing on the other side of the board to the mounting layer of the package.

Section 8.2.3.6.4 Pin

- Added attribute `pinPolarity` having enumerations PLUS | MINUS |ANODE |CATHODE.
- Added enumeration WIRE_BOND to attribute `mountType`.

Section 8.2.3.7 Component

- Added child element `SlotCavityRef` (described in section 8.2.3.7.1) to element `Component`, to reference the slot or cavity that embeds or recesses the component in the PCB

- Added enumerated values EMBEDDED | FORMED | PRESSFIT | WIRE_BONDED | GLUED | CLAMPED | SOCKETED to attribute `mountType`.
- Added optional attributes `matDes, layerRefTopside, modelRef`. Made attributes `refDes, packageRef` optional
- Added optional child element `SpecRef`, to reference specifications pertaing to the component instance
- Added new section to define `SlotCavityRef`, used as a child element of `Component`.
- Added new sections to show examples of discrete emebedded, wire bonded, and formed components, and coins

Section 8.2.3.8 LogicalNet
- Added optional attribute `netPair,` to identify a net name that forms a differential pair with the net
- Added optional child element `PortRef`, that defines an interface to another `Step`, such as a daughter board or custom IC
- Added optional child element `SpecRef`, to reference specifications pertaing to the logical net

Section 8.2.3.9.1 PhyNet
- Added a new element `PortRef` as a child of `PhyNetPoint`, to reference a `Port` definition (see reference to Section 8.2.3.11 below).

Section 8.2.3.10.1 Set
- Added values TEXT, TEARDROP and GRAPHIC to enumerated list for attribute `geometryUsage`.
- Added optional attribute `netPair`, to identify a net name that forms a differential pair with the net associated with the set

Section 8.2.3.10.5 Hole
- Added optional attribute `type` to element `Hole`, with possible enumerated values CIRCLE or SQUARE.
- Added optional child element `Xform` to element `Hole,` to be able to assign an angle of rotation to a square drill hole
- Added type VIA_CAPPED to the enumerated list of `platingStatusType`, for attribute `platingStatus`, to identify plated over vias, also known as VIPPO, CAPPED, POFV

Section 8.2.3.10.6 SlotCavity
- Reworded definition of `SlotCavity`, to clarify, cut to depth, and cut to span options
- Added attributes `startCutLayer` and `direction` to child elements `MaterialCut` & `MaterialLeft` to enable definition for only one layer, or when direction is ambiguous.
- Added child element `Fill` to describe any full or partial fill of the `SlotCavity` by a given material

Section 8.2.3.10.7 Features
- Added the following clarifying description to the `Location` element in `Features`: “The location(s) of the Feature if it is a StandardShape. If the Feature has it’s own built in co-ordinates, such as Line or Polyline then a Location is not required, but then there can be only one instance”

Section 8.2.3.10.8 NetShort
- Added a new element `NetShort` as a child of `Set`, to define intentional physical and electrical shorts between two or more given net names

Section 8.2.3.11 BendArea
- Added a new element `BendArea` as a child of `Step`, to define a bend in a flex circuit

Section 8.2.3.12 StackupZone

- Added a new element `StackupZone` as a child of `Step`, to define stackup zones.

Section 8.2.3.13 Port

- Added a new element `Port` as a child of `Step`, to define a physical and electrical link of type `Wirebond`, `ConnectorMate`, or `ComponentPad`

Section 8.2.3.14 Model

- Added a new element `Model` as a child of `Step`, to define a 3D model. This currently has a single child element `Extrusion` to describe a simple 3D shape as a combination of 2D shapes extruded to a given height. But more child element types could be added in future releases.

Section 8.2.3.15 Dfx

- Renamed parent element `DfxMeasurementList` to simply `Dfx`
- Changed attribute `Criteria` to a child element of `Dfx` and made `DfxMeasurement` a child of `Criteria`
- Added a 2nd child element `DfxQuery` to `Dfx`, which has a child element `DfxResponse`
- Refer to section 8.2.3.15 for further details.

Appendix A

- Removed Appendix A, so that Appendix B became A, C became B, and so on.

Appendix B (was C)

- Added new and missing `layerFunction` types

Schema Only Changes

1. Removed `OEMDesignerKeyRef` entries, to remove requirement for `OEMDesignNumberRef` to point to an existing `OEMDesignNumber` in the AVL Section.
2. Added minOccurs="1" to `PolyStep` in `Polygon`, to match this document, section 3.4.4
3. Changed minOccurs="2" to minOccurs="1" for `PolyStep` in `PolyLine`, to match this document, section 3.4.5